QSC6240™/QSC6270™ QUALCOMM Single Chip

User Guide

80-VF846-3 Rev. A

September 25, 2007

Submit technical questions at:
https://support.cdmatech.com

QUALCOMM Confidential and Proprietary

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714
U.S.A.

Revision history

Revision	Date	Description
A	September 2007	Initial release

Contents

11 Internal Baseband Functions

12 Baseband-Analog/RF-PM Interfaces

13 Input Power Management

14 Output Voltage Regulation

15 General Housekeeping

Figures

Tables

1 Introduction

1.1 Documentation overview

Two QUALCOMM Single Chip™ (QSC™) devices are addressed in this user guide — the QSC6240™ and QSC6270™. These devices are jointly referred to as the QSC6240/QSC6270 or QSC62x0™ device whenever the material being presented applies equally to both devices. Technical information for both devices is primarily covered by five documents (Table 1-1). Each is a self-contained document, but a thorough understanding of the device and its applications requires familiarization with all five documents. All released QSC62x0 documents are posted on the CDMA tech support website (https://support.cdmatech.com) and are available for download.

Table 1-1 Primary QSC6240/QSC6270 documentation

Document number	Title/description
80-VF846-1	*QSC6240/QSC6270 QUALCOMM Single Chip0 Device Specification* The primary objective of this document is to convey all QSC62x0 electrical and mechanical specifications. Additional material includes pin assignment definitions, PCB mounting specifications, packing methods and materials, and part reliability. This document can be used by company purchasing departments to facilitate procurement.
80-VF846-2	*QSC6240/QSC6270 QUALCOMM Single Chip Software Interface Document* Provides detailed information about the QSC62x0 software interface and its clocks, security, user interface, and registers.
80-VF846-3 (this document)	*QSC6240/QSC6270 QUALCOMM Single Chip User Guide* Provides detailed descriptions of all QSC62x0 functions and interfaces, including its various operating modes.
80-VF846-4	*QUALCOMM Single Chip QSC6240/QSC6270 Revision Guide* Provides a history of QSC62x0 device revisions and changes to its device specification. It explains how to identify the various device revisions, discusses known issues (or bugs) for each revision and how to work around them, and lists performance specification changes between each revision of the Device Specification (80-VF846-1).
80-VF846-5	*QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* This document tries to anticipate and answer questions hardware engineers may have when incorporating the QSC62x0 device into their product designs. Example applications are presented, then specific design topics, such as layout guidelines, power distribution recommendations, external component recommendations, trouble shooting techniques, and more are addressed.

This QSC6240/QSC6270 user guide is organized as follows:

Chapter 1 An overview of QSC62x0 documentation, gives a high-level functional description of the device and a typical application, lists terms and acronyms used throughout this document, and defines the marking conventions used herein

Chapter 2 Device pin assignments

Chapter 3 RF signal paths and LO circuits — detailed functional descriptions and interface requirements

Chapter 4 Audio and housekeeping ADC — detailed functional descriptions and interface requirements

Chapter 5 Baseband processors — detailed functional descriptions

Chapter 6 Memory support (EBI1 and EBI2) — detailed functional descriptions and interface requirements

Chapter 7 Air interfaces — functional descriptions

Chapter 8 Multimedia (camera interface, video, etc.) — detailed functional descriptions and interface requirements

Chapter 9 Connectivity — detailed functional descriptions and interface requirements

Chapter 10 General purpose input/output — lists of configurable GPIOs, their pad structure, and their expected usage

Chapter 11 Internal baseband functions — detailed functional descriptions and interface requirements

Chapter 12 Baseband-analog/RF-PM interfaces — detailed functional descriptions and interface requirements

Chapter 13 Input power management — detailed functional descriptions and interface requirements

Chapter 14 Output voltage regulation — detailed functional descriptions and interface requirements

Chapter 15 General housekeeping — detailed functional descriptions and interface requirements

Chapter 16 PM interfaces and multipurpose pins — detailed functional descriptions and interface requirements

Chapter 17 Power and ground — lists of power and ground connections and associated recommendations

1.2 Concise device and application descriptions

QUALCOMM CDMA Technologies (QCT) strives to constantly improve the indispensable communication tools our customers use every day. QCT creates state-of-the-art chipsets, system software, development tools, and products such as the Launchpad™ suite of hardware cores and software.

The QSC62x0 family of devices represents the next generation of chipset architecture and enhancements for the QCT value and multimedia tiers of products. The QSC62x0 family includes:

- The QSC6240 device that replaces (for example) the MSM6245™ baseband modem, RTR6285™ RF transceiver, and PM6658™ power manager — for handset products.
- The QSC6270 device that replaces (for example) the MSM6255A™ or MSM6260™ baseband modem, RTR6285 RF transceiver, and PM6658 power manager — for handset products and data card/module configurations with high-speed downlink packet access (HSDPA).

These airlink and multimedia capabilities are supported by integrating Mobile Station Modem™ (MSM™) baseband, radioOne® RF, and power management functionality into a single 12 mm x 12 mm chip-scale package (CSP). Together these functions perform all the signal processing and power management tasks within a subscriber unit. This enables reduced handset complexity, cost, time-to-market, and board-space requirements while providing many features and functionalities.

The global expansion of 3G WCDMA and GSM networks has extended the availability of high-speed, wireless data access. With increased accessibility comes increased demand for wireless devices that function as cameras, camcorders, personal video players, MP3 audio players, gaming consoles, and phones. To efficiently support next-generation data speeds and functionality, wireless devices must integrate applications processors with high-performance modems. The QSC62x0 devices extend the level of integration to include radio frequency and power management functions.

3G products based upon the QSC62x0 devices may include:

- Voice and data phones
- Music player-enabled devices and applications
- Camera phones
- Multimedia phones, including gaming, streaming video, videoconferencing, and more
- Position location devices
- Wireless connectivity — Bluetooth®, WLAN, near-field communicator (NFC), broadcast
- Other applications and devices

QSC62x0 benefits are applied to each of these product types, including:

- Higher integration to reduce PCB surface area, power consumption, time-to-market, and bill of material (BOM) costs while adding capabilities and processing power
  - Baseband functions, including multiple hardware cores
  - radioOne RF and analog functions (Rx and Tx, both eliminating their intermediate frequency (IF) components)
  - Power management functions
- Integrated hardware cores eliminate multimedia co-processors, providing superior image quality and resolution for mobile devices while extending application times:
  - Longer run time for mobile devices over other industry solutions that use companion processors
- Location-based services and applications, including points of interest, personal navigation, and friend finder
- A single platform that provides dedicated support for all market leading codecs and other multimedia formats to support carrier deployments around the world
- DC power reduction using innovative techniques

To simplify discussion of this highly integrated device, topics are addressed within three major categories that generally align with previous generation IC partitioning:

- Analog/RF functions
  - RF transmitter
  - RF receiver
  - Housekeeping ADC
  - Audio codec
- Baseband (BB) functions
  - Processors
  - Memory support
  - Connectivity
  - Multimedia
  - Air interfaces
  - Interfaces with other functions
  - Internal BB functions
  - General purpose I/Os

- Power management (PM) functions
  - Input power management
  - Output voltage regulation
  - General housekeeping
  - User interfaces
  - PM-level interfaces

As mentioned earlier, there are two variants of this QSC device: the QSC6240 for handsets and the QSC6270 for handsets and data card/module configurations. Further summary of the differences between variants is presented in Section 1.3. Before discussing the subtle differences between variants, this chapter concisely describes a functional block diagram of the QSC62x0 device in a typical application and illustrates it in Figure 1-1. More detailed descriptions of each function and interface are included in this document's later chapters.

The high level of QSC62x0 integration not only incorporates all of the functions normally implemented within three separate ICs (RF transceiver IC, PMIC, and MSM device), it also moves the many interfaces between those functions into the 424 CSP package. The board area is reduced, and the layout effort is simplified.

The antenna is connected to the QSC device's transmitter and receiver paths using a switch module and simplified RF front-end components (simplified relative to previous generation implementations):

- UMTS paths require duplexers, power amplifiers, Tx power detection, and Tx bandpass filters.
  - QSC6240/QSC6270 advancements have eliminated the Rx inter-stage filters.
- GSM paths require power amplifiers and Rx input filter functions.
- In some applications, the UMTS and GSM Rx inputs can share the same RF front-end paths, including shared QSC RF inputs:
  - UMTS850 and GSM850 use the same front-end path.
  - UMTS1900 and GSM1900 use the same front-end path.

With the Rx interstage filters eliminated, once the RF signals are routed on-chip, they stay on-chip. The Rx and Tx baseband signals are fully contained within the device, as are all of the modulation and demodulation functions normally performed by an MSM device. The air interfaces (multiband UMTS and GSM) support voice and data capabilities.

All the usual digital signal processing and software functions are also supported: modem and applications processors, memory support, high connectivity, camera and video, audio, interfaces with other internal and external analog/RF and power management functions, other internal baseband functions (clock generation, security, etc.), and so on.

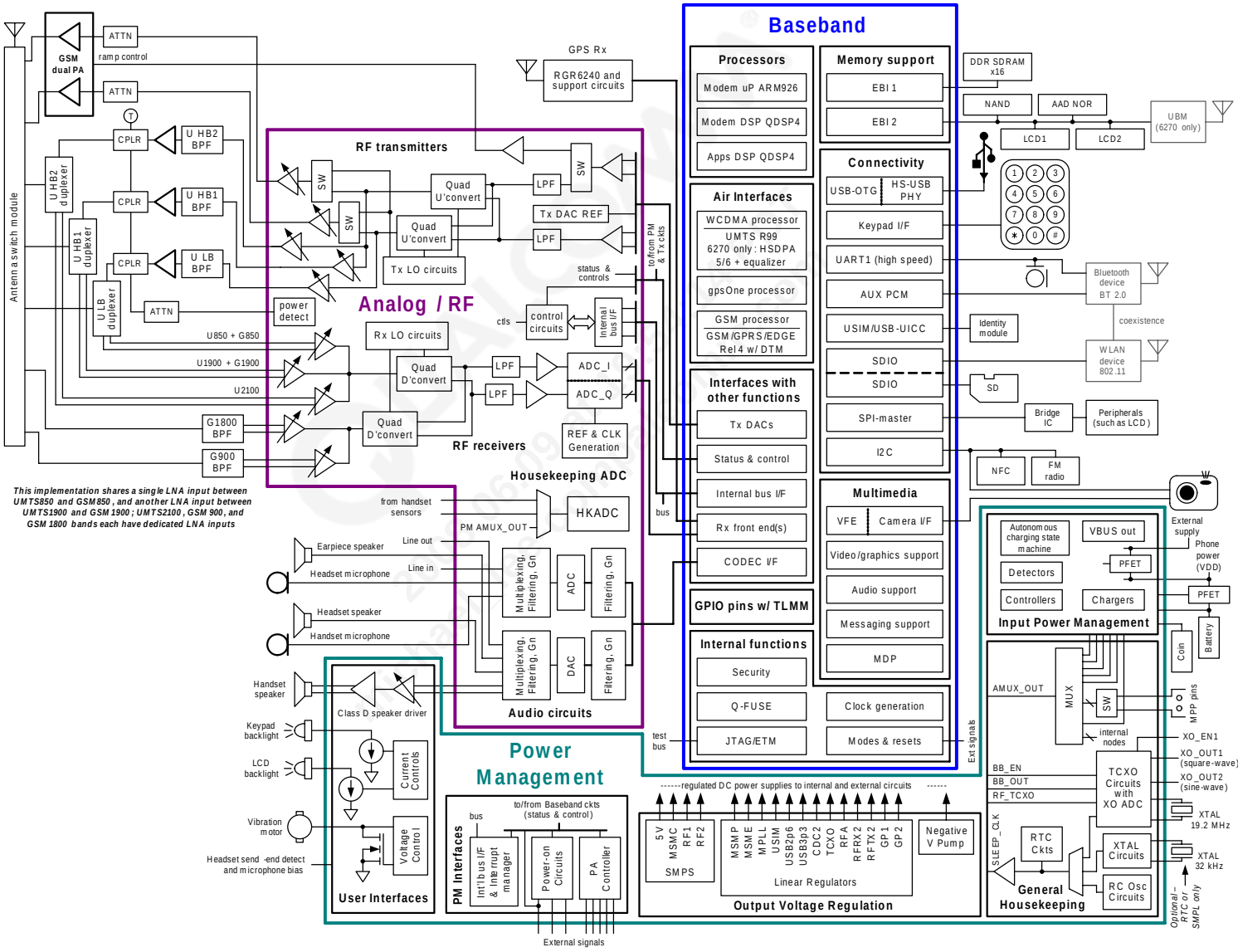


Figure 1-1 QSC62x0 functional block diagram and typical application

Supported user interfaces and connectivity options include:

- The keypad (and its backlighting)
- Universal serial bus on-the-go (USB-OTG)
- High-speed universal asynchronous receiver transmitter (UART)
- UMTS subscriber interface module (USIM)
- USB universal integrated circuit card (UICC)
- Dual secure digital input/output (SDIO)
- Serial peripheral interface (SPI), master only
- Inter-integrated chip (I2C)
- Bluetooth
- WLAN
- Universal Broadcast Modem™, UBM™ (QSC6270 only)
- NFC
- FM radio
- Camera
- Speakers
- Microphones
- Multiple LCDs (and their backlighting)
- A vibration motor

GPS position location is also supported by the QSC62x0 device with the addition of the RGR6240™ GPS RF Receiver IC.

With the integrated power management functions, the device generates its own regulated supply voltages from an external supply or the handset battery. Power sources are continually monitored, enabling the best source to be selected automatically and a low battery and/or coin cell to be recharged. System clocks are generated, verified, and distributed internally.

Brief descriptions of the three major QSC62x0 functional categories follow.

1.2.1 Analog/RF functional description

The analog/RF functions are highlighted in purple in Figure 1-1.

radioOne technology

The QSC62x0 device incorporates the QUALCOMM radioOne technology — the technology for RF transceivers that converts received signals directly from RF-to-baseband and transmits signals directly from baseband-to-RF (known as direct conversion or zero intermediate frequency (ZIF) processing). This technique eliminates the need for large IF surface acoustic wave (SAW) filters and supporting IF and LO circuits, thereby reducing the handset parts count and facilitating multiband, multimode handsets that can be produced in smaller form factors.

RF transmitter

The RF transmitters are leveraged from previous-generation RTR devices, including the latest innovations. It provides the ZIF transmit signal paths for multiband, multimode applications:

- Tri-band UMTS, with one low band and two high bands selected from:
  - Low band (select one)
    - BC5 (CELL, 824 to 849 MHz)
    - BC6 (JCELL, 830 to 840 MHz)
    - BC8 (EGSM, 880 to 915 MHz)
  - High band (select two)
    - BC1 (IMT, 1920 to 1980 MHz)
    - BC2 (PCS, 1850 to 1910 MHz)
    - BC3 (DCS, 1710 to 1785 MHz)
    - BC4 (AWS, 1710 to 1755 MHz)
    - BC9 (J1700, 1749.9 to 1784.9 MHz)
- Quad-band GSM:
  - Low band
    - GSM850 (824 to 849 MHz) and GSM900 (880 to 915 MHz)
  - High band
    - DCS1800 (1710 to 1785 MHz) and PCS1900 (1850 to 1910 MHz)

The transmit signal paths include a shared set of baseband amplifiers, a dedicated quadrature upconversion for each band type (low and high), gain control RF amplification, and multiple output driver amplifiers for each band type. Three UMTS output drivers support one low band and two high bands; two GSM output drivers support one low-band and one high-band type (but each GSM band type is dual-band). The GSM transmitters are also supported by a PA envelope control path, plus the ability to route LO signals to the transmit chains for test and calibration purposes.

Numerous secondary Tx functions are also integrated: a reference for the transmit DACs, the Tx phase-locked loop (PLL), the Tx local oscillator circuit, the Tx LO generation and distribution circuits, an RMS Tx power detector, and various interface, control, and status circuits.

The RF transmitter interfaces internally with the baseband circuits for its analog baseband input and status and control signaling. Power reduction features controlled by baseband circuits (such as selective circuit powerdown, gain control, and transmit puncturing) extend handset talk time. The driver amplifier outputs are routed externally to the final stages of the transmit chains, culminating with the antenna switch whose output drives the antenna.

Sophisticated Tx LO circuits implement the frequency plan and are completely integrated on-chip. All Tx LO signals are generated by the on-chip Tx local oscillator under the control of its PLL.

RF receivers

The RF receiver designs are leveraged from previous-generation RTR devices, including the latest innovations. The ZIF receive signal paths support multiband, multimode applications:

- Tri-band UMTS, with one low band and two high bands selected from:
  - Low band (select one)
    - BC5 (CELL, 869 to 894 MHz)
    - BC6 (JCELL, 875 to 885 MHz)
    - BC8 (EGSM, 925 to 960 MHz)
  - High band (select two)
    - BC1 (IMT, 2110 to 2170 MHz)
    - BC2 (PCS, 1930 to 1990 MHz)
    - BC3 (DCS, 1805 to 1880 MHz)
    - BC4 (AWS, 2110 to 2155 MHz)
    - BC9 (J1700, 1844.9 to 1879.9 MHz)
- Quad-band GSM:
  - Low band
    - GSM850 (869 to 894 MHz) and GSM900 (925 to 960 MHz)
  - High band
    - DCS1800 (1805 to 1880 MHz) and PCS1900 (1930 to 1990 MHz)

The on-chip receive signal paths are functionally identical for each mode (UMTS or GSM) and each band type (low or high). The external circuitry includes the antenna switch module and a filter function, either a duplexer or a bandpass filter.

All RF Rx inputs use a differential configuration to maximize common-mode rejection, Tx isolation, out-of-band suppression, and second-order intermodulation performance. The first of two quadrature downconverters accepts inputs from three LNAs (only one is active at a time). An example application could support the following bands using these three LNAs:

- UMTS2100
- UMTS1900 and GSM1900
- UMTS850 and GSM850

The second downconverter accepts inputs from another two LNAs (again, just one is active at time) and are dedicated to GSM900 and GSM1800 operation. See Table 2-1 for a complete listing of all Rx RF input pin assignments.

The two downconverter outputs drive analog baseband filters and buffer circuits that are programmed to support the active operating mode's waveforms (WCDMA or GSM). The analog baseband signals are then digitized by analog-to-digital converters (ADCs) whose outputs are routed to the digital baseband circuits for further processing.

Numerous secondary Rx functions are also integrated: Rx frequency synthesizers (each having their own PLL and local oscillator circuits), LO generation and distribution circuits, reference and clock circuits for the ADCs, and various interface, control, and status circuits. Power reduction features (such as selective circuit powerdown, gain control, and bias control) extend handset standby time.

Like the Tx LO, all Rx LO circuits are completely integrated. All received LO signals are generated by the on-chip Rx local oscillators under control of their PLLs.

GPS position location is supported by the QSC62x0 device when supplemented by the RGR6240 IC. GPIOs are configured to accept the RGR's output data streams, then the QSC62x0 device performs the necessary global positioning system (GPS) processing using gpsOne® technology as discussed below.

gpsOne technology

gpsOne technology merges GPS satellite and network information to provide a high-availability solution that offers industry-leading accuracy and performance. This solution performs well even in very challenging environmental conditions where conventional GPS receivers fail. It provides a platform to enable wireless operators to address both location-based services and emergency mandates, such as the United States FCC E911 mandate.

When a request for position location is issued in mobile-assisted mode, the available network information is provided to the location server (e.g., cell-ID), and assistance is requested from the location server. The location server sends the assistance information to the handset. The handset/mobile unit measures the GPS observables and provides the GPS measurements along with available network data (that is appropriate for the given air interface technology) to the location server. The location server then calculates the position location and returns the results to the requesting entity.

In mobile-based mode, the assistance data provided by the location server encompasses not only the information required to assist the handset in measuring the satellite signals, but also the information required to calculate the handset's position. Therefore, rather than provide the GPS measurements and available network data back to the location server, the mobile calculates the location on the handset and passes the results to the requesting entity.

In standalone (autonomous) mode, the handset demodulates the data directly from the GPS satellites. This mode has some reduced cold-start sensitivity and a longer time to first fix as compared to the assisted modes. However, it requires no server interaction and works in out-of-network coverage situations.

This combination of GPS measurements and available network information provides:

- A high-sensitivity solution that works in all terrains: indoor, outdoor, urban, and rural
- High availability that is enabled by using both satellite and network information

Therefore, while network solutions typically perform poorly in rural areas and areas of poor cell geometry/density, and while unassisted, GPS-only solutions typically perform poorly indoors, the QUALCOMM gpsOne solution provides optimal time-to-fix, accuracy, sensitivity, availability, and reduced network utilization in both of these environments, depending on the given condition.

The gpsOne solution in assisted modes provides a cold-start GPS sensitivity improvement of approximately 20 to 30 dB over unassisted, conventional GPS receivers.

Compared to network solutions that require equipment at each cell site, the gpsOne solution integrates a complete GPS receiver in the RGR6240 and QSC62x0 chipset combination. This means that each handset is capable of position location without requiring expensive cell site equipment. This solution not only can be used to help operators address the FCC E911 mandate in the United States (and mandates planned for other countries), but also provides a ubiquitous platform for location-based applications. The gpsOne solution enables consumer-priced, position-capable handsets for location-based services worldwide.

Housekeeping ADC

The housekeeping ADC (HKADC) circuit includes a multiplexer that selects between three inputs: one of two external signals or the PM analog multiplexer output. The ADC itself provides 12-bit resolution, 10-bit accuracy, with a sampling rate of 2.4 MHz (TCXO/8). Intended uses include monitoring of the battery voltage, various temperatures, and key power supply voltage nodes.

Audio codec

The final block within the analog/RF function is the audio codec.

The audio Tx path supports three analog inputs: a handset microphone, a headset microphone, and a differential or single-ended line input. The line input provides voice and stereo-to-mono conversion to enable FM radio mono record. The HSED_BIAS pin provides the microphone bias supply while supporting the one-touch headset control feature. The Tx path sample rate is 8, 16, or 32 kHz.

The audio Rx path includes stereo 16-bit audio DACs, with a typical dynamic range of 89 dB in the audio mode. Differential earpiece, differential or stereo headphone, handset speaker, and differential or stereo line outputs are supported. The stereo headphone interface can be implemented in either the new, ground-referenced capless configuration or in the legacy, capacitor-coupled configuration. Simultaneous output port operation is supported, and the DAC output(s) can be summed with one or both line inputs.

The codec also simultaneously supports different sampling rates in the Tx and Rx paths.

1.2.2 Baseband functional description

The baseband functions are highlighted in blue within Figure 1-1. The figure and the following description are primarily based upon the QSC6270 device. Differences between the QSC6240 and QSC6270 variants are discussed within this section and then summarized in Section 1.3.

Processors

The QSC62x0 device includes three integrated processors:

- Modem microprocessor — a low-power, high-performance RISC microprocessor core running at 184 MHz and featuring the ARM926EJ-S™ CPU and Jazelle™ accelerator circuit from ARM® Limited
- Modem digital signal processor (mDSP) — the low-power, high-performance QUALCOMM digital signal processor (QDSP®) targeting 115 MHz
- Application digital signal processor (aDSP) — the low-power, high-performance QDSP targeting 115 MHz

Hardware acceleration eliminates the need for the multimedia companion processors normally required for video and audio-based applications that support MP3 music files, a MIDI synthesizer, and still image capture and view. By removing the need for costly applications coprocessors and memory subsystems, the QSC62x0 solution reduces BOM costs and increases standby and talk times.

Memory support

In addition to the internal DSP memories, QSC62x0 baseband circuits support the following memory features:

- 92-MHz bus clock for 16-bit DDR SDRAM (on EBI1)
- Dual-memory buses separating the high-speed memory subsystem (EBI1) from low-speed peripherals (EBI2) such as LCD panels
- 1.8 V memory interface support on EBI1
- 1.8 V or 2.6 V memory interface support on EBI2
- NAND and AAD NOR support on EBI2

Air interfaces

The supported air interface standards and features include:

- *UMTS/WCDMA/GSM/GPRS/EDGE Specification Release '99* (3GPP R99)
- *GSM/GPRS/EDGE Specification Release 4* (3GPP R4)
- *UMTS/WCDMA Specification Release 5* (3GPP R5, QSC6270 only)
- Enhanced GPS position location using gpsOne

RF and PM interfaces

The QSC62x0 device supports multiple RF bands and various airlink interfaces (as identified earlier). Baseband and RF circuits — internal to the QSC device and external — are powered by the QSC device's power management circuits.

The baseband circuits' RF and PM interfaces enable communications with the other major functions within the QSC device, including:

- Real-time UMTS and GSM Rx and Tx data, plus GPS Rx data, via the Tx DACs and the Rx digital front-ends
- RF status and control signals via the internal bus interfaces and other one-to-one dedicated connections
- PM status and control signals via the internal bus interfaces and other one-to-one dedicated connections
- Sensor measurements from the HKADC via the internal bus interfaces
- Real-time codec data via dedicated connections

In addition to these internal QSC interfaces, a few connections are required for controlling external components — the antenna switch, for example. Some of the 78 GPIOs are used to control external RF components, but many are available for alternate uses as desired by the wireless product designers.

Camera and video functions

The QSC device provides image processing capabilities, including a direct interface for digital camera modules with video front-end (VFE) image processing. The high-quality digital camera processing supports CCD or CMOS image sensors up to 2.0 MP for the QSC6240 device and up to 3.0 MP for the QSC6270 device.

Connectivity

The QSC62x0 device supports a variety of consumer electronics with a wide range of connectivity options, including:

- Keypad
- Universal serial bus (USB-HS 2.0) and USB-OTG with integrated physical layer (PHY)
- High-speed UART serial port

- Bluetooth 2.0, supported through an external device using the high-speed UART interface for data transport and auxiliary PCM for audio
- USIM and USB-UICC controller that supports various identity modules
- Two 4-bit secure digital (SD) controllers for SD and mini SD cards
- WLAN (802.11), supported through an external device using one of the SDIOs
- SPI (master only) that communicates through an external bridge IC to control external peripherals such as LCDs
- I^2C bus that supports an NFC, FM radio, and camera controls
- Parallel LCD interface on EBI2
- Flexible, configurable general-purpose I/O pins
- UBM (broadcast) – QSC6270 only

Internal baseband functions

QSC62x0 baseband functions that do not easily fall into any of the other categories, that do not support the other functions directly, or that enable basic device operations are assigned to the internal functions category. These functions include PLLs and clock generation, JTAG/ETM, modes and resets, Qfuse, and other security functions.

The QSC62x0 device includes support for the SecureMSM™ v2 security solution, adding the following capabilities:

- Secure boot – protects against reverse engineering and reflashing attacks:
  - Trusted third-party certificate authority (GeoTrust) provides code-signing service to manufacturers and QUALCOMM.
  - Strong 2048-bit RSA keys are used and burned in ROM to ensure that the key is completely immutable.
  - Manufacturers and QUALCOMM can ensure that their code is running unchanged on the device.
- Secure software environment:
  - Capabilities-based architecture with APIs available only to applications components with the required, signed permissions.
  - Partitioned software provides isolation of compromised applications; memory is read-protected and write-protected.
  - Only known, trusted processes are permitted to access materials in the secure file system (SFS).
- SFS:
  - This provides a 128-bit hardware key for (AES) encryption of SFS.
  - Operations using hardware key are run in on-chip RAM, making observation extremely difficult.
  - An unchangeable hardware ID is unique to each QSC device.

These capabilities result in a very robust platform securing next-generation digital rights management (DRM) services, including Open Mobile Alliance (OMA) DRM v2 services and e-commerce transactions. An international mobile equipment identity (IMEI) freeze and SIM lock protection benefit from hardware security.

GPIOs

The QSC62x0 device includes 78 general-purpose I/O pins. Almost all of these versatile I/Os are configured by advanced mobile subscriber software (AMSS™) for intended functions with the option to easily be reconfigured for alternate functions. The GPIOs can be reconfigured to support any digital I/O function needed by a particular handset design.

Software support

In addition to all this hardware functionality, the QSC62x0 device runs software that extends handset capability to include:

- AMSS
- The Launchpad suite of applications technologies
- Support for the QUALCOMM Binary Run-time Environment for Wireless® (BREW®) solution

AMSS

QCT provides a complete software suite – AMSS – for building handsets based upon the QSC62x0 device. AMSS software is designed to run on reference phone platforms — optional development platforms optimized to assist in evaluating, testing, and debugging AMSS software.

Launchpad

The Launchpad suite of applications technologies offers wireless operators and manufacturers a cost-effective, scalable, and timely solution for providing advanced wireless data services. This seamlessly integrated solution enables advanced next-generation applications and services that incorporate multimedia, position location, connectivity, and customized user interface and storage capabilities. Launchpad features are available for each QUALCOMM chipset and QSC device, closely matching the specific functionality and cost-target objectives agreed upon in joint product planning with manufacturers and wireless service operators worldwide.

The QSC62x0 solution supports the advanced feature set of the QUALCOMM Launchpad suite of technologies, including streaming video and audio, still-image and video encoding and decoding, Java® acceleration, and a megapixel camera interface.

BREW

The QSC62x0 device includes support for the QUALCOMM BREW solution. BREW is a complete product and business system for the development and over-the-air deployment of data services on wireless devices. The BREW solution provides the necessary tools and value-added services to developers, device manufacturers, and wireless operators for application development and distribution, device configuration, and billing/payment.

1.2.3 Power management functional description

The power management (PM) functions are highlighted in green within Figure 1-1.

Input power management

The input power management portion of its block accepts power from common sources – the main battery or an external charger – and generates all the regulated voltages needed to power the appropriate handset electronics. It monitors and controls the power sources, detecting which sources are applied, verifying that they are within acceptable operational limits, and coordinating battery and coin cell recharging while maintaining the handset electronics supply voltages.

The QSC62x0 device increases input power management integration by pulling the charging pass transistor on-chip and eliminating the sense resistor.

Output voltage regulators

On-chip circuits generate the handset's supply voltages using four switched-mode power supplies (one boost and three buck converters), 13 low dropout voltage regulators, and one negative charge pump switching regulator. All are programmable and derived from a common trimmed voltage reference.

General housekeeping

The PM general housekeeping functions include an analog multiplexer that has several internal and external connections. The internal connections are used to monitor on-chip functions such as the temperature sensor and reference voltage. Six external connections are hardwired to access input power nodes such as VCHG, VBAT, etc. And finally, four multipurpose pins can be configured as analog inputs and routed to the multiplexer through a switch matrix. These are available to monitor system parameters such as temperatures, battery ID, or other analog voltages. The multiplexer output signal is buffered and routed to the analog/RF circuits for analog-to-digital conversion by the HKADC.

Various oscillator, clock, and counter circuits are provided to initialize and maintain valid pulse waveforms and measure time intervals for higher-level handset functions. Multiple controllers manage the TCXO warmup and signal buffering and generate two separate XO outputs. The XO_OUT_GP1 is a square wave intended for WLAN and/or Bluetooth, while XO_OUT_GP2 is a sine wave intended for GPS. In addition to WLAN, Bluetooth, and GPS, these outputs can be used to support other functions at the discretion of the handset designer. These general-purpose outputs are independent of the on-chip circuits'

TCXO signals (for baseband and RF functions), allowing external circuits to operate even while the phone's baseband circuits are asleep and its RF circuits are powered down.

A new and improved search-and-acquisition algorithm may eliminate the need for a VCTCXO module (typically ± 2.5 ppm), enabling a less expensive 19.2-MHz crystal (typically ± 12 ppm) to be used; this is referred to as the XO circuit. The XO is supplemented by an on-chip ADC for digitizing the XO thermistor network's analog output voltage and compensating frequency variations versus temperature.

NOTE See the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* (80-VF846-5) for details about the 19.2-MHz XO and VCTCXO options.

In applications that do not support the real-time clock (RTC) or sudden momentary power loss (SMPL) functions, the 32.768 kHz crystal can be eliminated by enabling on-chip circuits that derive the sleep clock from the 19.2 MHz TCXO signal.

And finally, housekeeping circuits monitor key parameters to detect detrimental conditions, allowing the QSC device to protect itself.

User interfaces

Handset-level user interfaces are also supported. Two multipurpose pins (MPPs) can be used as programmable high-current drivers (MPP3 and MPP4) and are intended for keypad and LCD backlighting. A vibration motor driver alerts handset users of incoming calls, and a class-D speaker driver with volume control can be used for audio alerts or speakerphone and melody-ringer applications. The VREG_5V regulated output voltage can be used as the speaker driver power source, increasing the drive capability to as much as 1 W. One-touch headset detection and microphone bias are also included.

PM interfaces

PM circuit interfaces include an internal bus used by the baseband circuits to control and determine the status of the PM functions. This bus is supplemented by an interrupt manager for time-critical information. Another dedicated interface circuit monitors multiple trigger events and controls the power-on/power-off sequences.

The QSC62x0 device extends the PM interface functions to include a PA controller. This new circuit accepts a single control signal from the baseband circuits and generates the five power amplifier control signals for multimode, multiband applications.

1.3 Device variants

Table 1-2 provides a list of key QSC6240 and QSC6270 features, highlighting the differences between the two devices (differences highlighted in bold blue text).

Table 1-2 Device variants - key features with differences highlighted

Features	QSC6240	QSC6270
Processors		
Modem	ARM926EJ-S at 184 MHz	ARM926EJ-S at 184 MHz
ADSP	QDSP4 at 115 MHz	QDSP4 at 115 MHz
MDSP	QDSP4 at 115 MHz	QDSP4 at 115 MHz
Process technologies	CMOS	CMOS
Airlinks		
WCMDA	Rel '99	Rel '99; **Rel 5 with HSDPA (3.6 Mbps)**
GSM/GPRS/EDGE w/ DTM	Rel 4	Rel 4
RF operating bands		
UMTS	Tri-band	Tri-band
GSM	Quad-band	Quad-band
Advanced receiver	—	**Equalizer**
Audio/video decoders	MP3, AAC, AAC+, ADPCM, MPEG4, H.263, H.264, RealAudio®, Windows Media®, AMR-WB/+	MP3, AAC, AAC+, ADPCM, MPEG4, H.263, H.264, RealAudio, Windows Media, AMR-WB/+
MIDI synthesizer	128-Poly - CMX	128-Poly - CMX
Acoustical audio improvements	Enhanced echo cancellation	Enhanced echo cancellation
Enhanced audio support	QConcert™/QAudioFX™	QConcert/QAudioFX
Qcamera™	Bayer and YUV sensors Up to 2.0 MP	**Bayer sensors up to 3.0 MP YUV sensors up to 2.0 MP**
Qcamcorder™ recording	15 fps at QCIF	**15 fps QVGA, 384 kbps (MP4) 10 fps QVGA, 384 kbps (H.264)**
Qvideophone™	Codecs: MPEG4, H.263 Performance: 15 fps at QCIF, 64 kbps	Codecs: MPEG4, H.263 Performance: 15 fps at QCIF, 64 kbps
Qtv™ video decode	15 fps QCIF streaming 15 fps QCIF playback	**15 fps QVGA streaming 30 fps QVGA playback**
gpsOne standalone & A-GPS	Through external RGR6240	Through external RGR6240
USB 2.0 high speed	Integrated PHY, OTG	Integrated PHY, OTG
Bluetooth 2.0 via UART/PCM	Through BTS402x™	Through BTS402x
WLAN (802.11) via SDIO	Through external SoC	Through external SoC
LCD via EBI2	18 bpp/up to QVGA panels	18 bpp/up to QVGA panels
Digital rights management	OMA DRM v2.0	OMA DRM v2.0

Table 1-2 Device variants - key features with differences highlighted (continued)

Features	QSC6240	QSC6270
SD/SDIO/MMC - 4 bit/52 MHz	2 (for media/audio and WLAN)	2 (for media/audio and WLAN)
High speed UART (4 Mbps)	1 (for BT, diagnostics)	1 (for BT, diagnostics)
USIM interface	USIM/USB-UICC	USIM/USB-UICC
EBI1	16-bit DDR SDRAM	16-bit DDR SDRAM
EBI2	NAND and AAD NOR	NAND and AAD NOR
I2C	1 (camera ctl, NFC, FM radio)	1 (camera ctl, NFC, FM radio)
NFC	Yes	Yes
MDP	Yes	Yes
SPI (master only)	Yes	Yes
UBM (broadcast)	No	**Yes**

1.4 Terms and acronyms

The following table defines abbreviations commonly used throughout this document.

Table 1-3 Terms and acronyms

Term	Definition
ACM	Abstract control model
ADC	Analog-to-digital converter
AEC	Acoustic echo cancellation
AGC	Automatic gain control
AMSS	Advanced Mobile Subscriber Station (software)
BER	Bit error rate
BPF	Bandpass filter
Bps	Bits per second
BT	Bluetooth
CAGC	CDMA AGC
CDC	Communication device class
CELP	Code excited linear prediction
CMX®	Compact Media Extension (registered by QUALCOMM)
Codec	Coder decoder
CSP	Chip-scale package
CRC	Cyclic redundancy code
DDR	Dual data rate
DRM	Digital rights management
DSP	Digital signal processor

Table 1-3 Terms and acronyms (continued)

Term	Definition
DTMF	Dual-tone multiple frequency
EBI	External bus interface
ECM	Ethernet networking control model or error control module
ETM	Embedded trace macrocell
EVRC	Enhanced variable rate codec
4GV™	Fourth-generation vocoder
GPIO	General-purpose input/output
HID	Human interface device
HPF	Highpass filter
HSDPA	High speed downlink packet access
HW	Hardware
IF	Intermediate frequency
JPEG	Joint Photographic Experts Group
JTAG	Joint Test Action Group (ANSI/ICEEE Std. 1149.1-1990)
kbps	Kilobits per second
LCD	Liquid crystal display
LNA	Low noise amplifier
LPF	Lowpass filter
LSBit or LSByte	Defines whether the LSB is the least significant bit or least significant byte. All instances of LSB used in this manual are assumed to be LSByte, unless otherwise specified.
MMC	Multimedia card
MPEG	Motion picture experts group
MSBit or MSByte	Defines whether the MSB is the most significant bit or most significant byte. All instances of MSB used in this manual are assumed to be MSByte, unless otherwise specified.
MSM™	Mobile Station Modem™ (trademarked by QUALCOMM)
MTP	Message transfer part (or point)
NMEA	National Marine Electronics Association
OBEX	Object exchange
OMA	Open Mobile Alliance
OTG	On-the-go
PA	Power amplifier
PCM	Pulse-coded modulation
PDM	Pulse-density modulation
PSRAM	Pseudo-static random access memory
QCELP	QUALCOMM Code Excited Linear Prediction

Table 1-3 Terms and acronyms (continued)

Term	Definition
QCIF	Quarter common intermediate format (176 pixels/line, 144 lines/frame)
QLIC	Quasi-linear interference cancellation
QSC	QUALCOMM Single Chip
RC	Resistance-capacitance
RF	Radio frequency
Rx	Receive
RTP	Real-time protocol
RTSP	Real-time streaming protocol
SDAC	Stereo digital-to-analog converter
SDIO	Secure digital input/output
SDRAM	Synchronous dynamic random access memory
SICD	Still-image capture device
SoC	System-on-Chip
SPI	Serial peripheral interface (master only)
Sps	Symbols per second (or samples per second)
SW	Software
TAP	Test access port
TCXO	Temperature-compensated crystal oscillator
Tx	Transmit
3GPP	Third-Generation Partnership Project
UART	Universal asynchronous receiver transmitter
UICC	Universal integrated circuit card
USB	Universal serial bus
USIM	UMTS subscriber interface module
VCTCXO	Voltage-controlled temperature-compensated crystal oscillator
WCDMA	Wideband Code Division Multiple Access
XO	Crystal oscillator
ZIF	Zero intermediate frequency

1.5 Special marks

The following table defines special marks used in this document.

Table 1-4 Special marks

Mark	Definition
[ ]	Brackets ([ ]) sometimes follow a pin, register, or bit name. These brackets enclose a range of numbers. For example, GPIO_INT [7:0] may indicate a range that is 8 bits in length, or DATA[7:0] may refer to all eight DATA pins.
_N	A suffix of _N indicates an active low signal. For example, EBI1_WE_N.
0x0000	Hexadecimal numbers are identified with an x in the number, for example, 0x0000. All numbers are decimal (base 10) unless otherwise specified. Non-obvious binary numbers have the binary term enclosed in parentheses at the end of the number, for example, 0011 (binary).

2 Pin Definitions

The highly integrated QSC6240/QSC6270 device is available in the 424-pin chip-scale package (424 CSP). Package details are available in the *BGA Package User Guide* (80-V2560-1). Pin assignments are illustrated and then listed in alphanumeric order.

Detailed performance specifications for all pins are included in the *QSC6240/QSC6270 QUALCOMM Single Chip Device Specification* (80-VF846-1) — voltage and current capabilities, pad attributes, (digital, analog, etc.), configurations (pull-up, pull-down, etc.), and so on.

2.1 Pin assignments

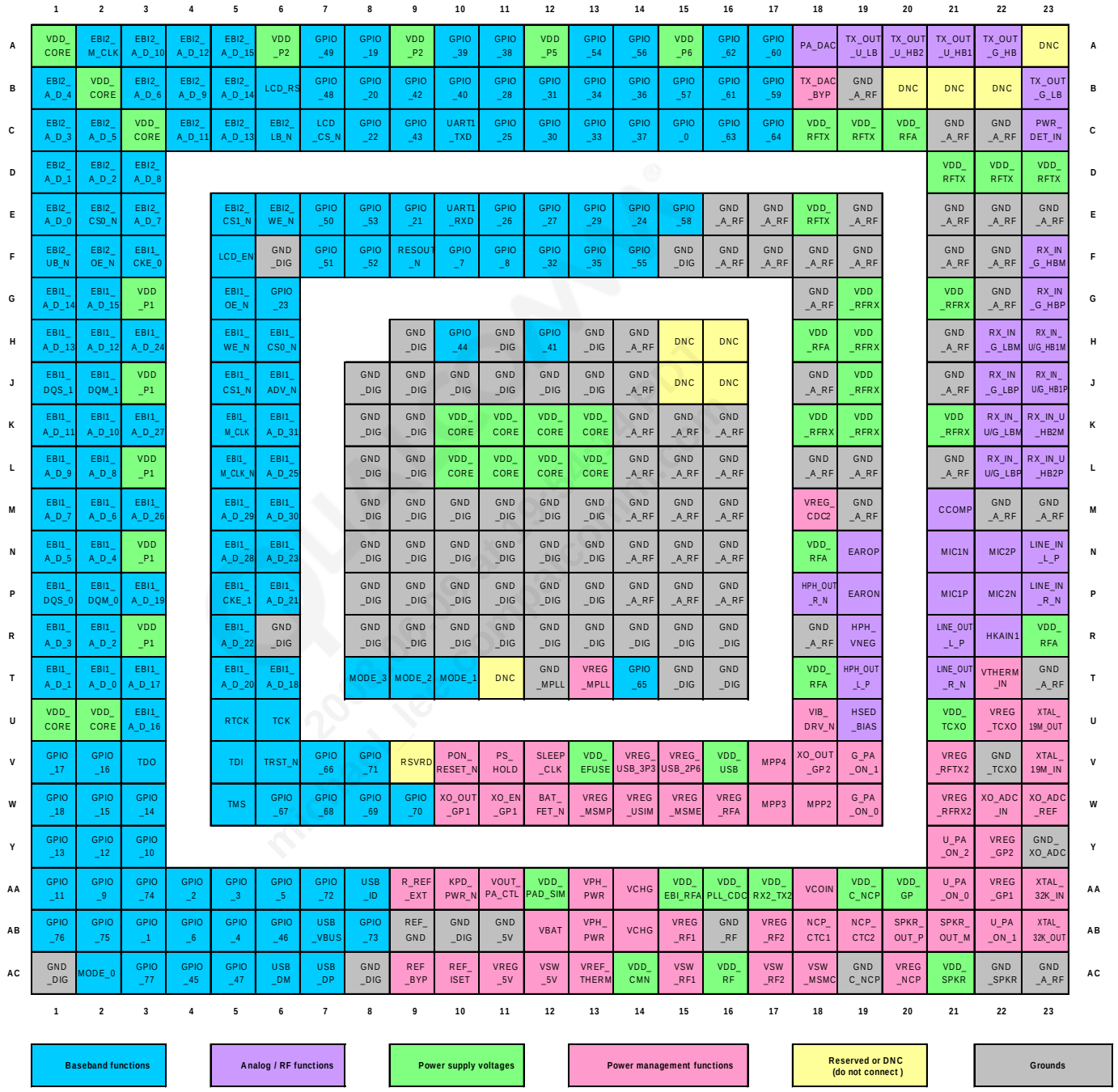


Figure 2-1 QSC6240/QSC6270 pin assignments (top view)

2.2 Pin descriptions

Pin assignments are listed alphanumerically and are assigned one or more electrical function (Table 2-1). Later chapters of this document contain greater detail; the mapping from electrical functions to later chapters is included in the table notes.

Table 2-1 QSC6240/QSC6270 pin assignments – listed in alphanumeric order

Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]	Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]
A1	VDD_CORE	VDD_CORE	PWR	B3	EBI2_A_D_6	EBI2_A_D_6	B-EBI
A2	EBI2_M_CLK	EBI2_M_CLK	B-EBI	B4	EBI2_A_D_9	EBI2_A_D_9	B-EBI
A3	EBI2_A_D_10	EBI2_A_D_10	B-EBI	B5	EBI2_A_D_14	EBI2_A_D_14	B-EBI
A4	EBI2_A_D_12	EBI2_A_D_12	B-EBI	B6	LCD_RS	LCD_RS	B-EBI
A5	EBI2_A_D_15	EBI2_A_D_15	B-EBI	B7	GPIO_48 EBI2_BUSY_CS3_N	Configurable I/O	B-GPIO B-EBI
A6	VDD_P2	VDD_P2	PWR	B8	GPIO_20 AUX_PCM_DIN	Configurable I/O	B-GPIO B-CON
A7	GPIO_49 EBI2_CS2_N	Configurable I/O	B-GPIO B-EBI	B9	GPIO_42 NFC_SHUTDOWN GP_CLK ETM_MODE_KYSNS_INT	Configurable I/O	B-GPIO B-CON B-INT B-ETM
A8	GPIO_19 AUX_PCM_SYNC	Configurable I/O	B-GPIO B-CON	B10	GPIO_40 I2C_SCL	Configurable I/O	B-GPIO B-CON
A9	VDD_P2	VDD_P2	PWR	B11	GPIO_28 CAMIF_DATA_0 ETM_PIPESTATB1	Configurable I/O	B-GPIO B-CAM B-ETM
A10	GPIO_39 I2C_SDA	Configurable I/O	B-GPIO B-CON	B12	GPIO_31 CAMIF_DATA_3 ETM_TRACE_PKTB1	Configurable I/O	B-GPIO B-CAM B-ETM
A11	GPIO_38 CAMIF_MCLK	Configurable I/O	B-GPIO B-CAM	B13	GPIO_34 CAMIF_DATA_6 ETM_TRACE_PKTB4	Configurable I/O	B-GPIO B-CAM B-ETM
A12	VDD_P5	VDD_P5	PWR	B14	GPIO_36 CAMIF_DATA_8 ETM_TRACE_PKTB6	Configurable I/O	B-GPIO B-CAM B-ETM
A13	GPIO_54 WLAN_PWR_EN	Configurable I/O	B-GPIO B-CON	B15	GPIO_57 GPS_ADCCLK	Configurable I/O	B-GPIO B-IOF
A14	GPIO_56 GPS_SSBI	Configurable I/O	B-GPIO B-IOF	B16	GPIO_61 ANT_SEL_1 WDOG_DISABLE	Configurable I/O	B-GPIO B-IOF B-INT
A15	VDD_P6	VDD_P6	PWR	B17	GPIO_59 ANT_SEL_3	Configurable I/O	B-GPIO B-IOF
A16	GPIO_62 ANT_SEL_0 BOOT_FROM_ROM	Configurable I/O	B-GPIO B-IOF B-INT	B18	TX_DAC_BYP	TX_DAC_BYP	P-OVR
A17	GPIO_60 ANT_SEL_2	Configurable I/O	B-GPIO B-IOF	B19	GND_A_RF	GND_A_RF	GND
A18	PA_DAC	PA_DAC	A-RTR	B20	DNC	DNC	NDR
A19	TX_OUT_U_LB	TX_OUT_U_LB	A-RTR	B21	DNC	DNC	NDR
A20	TX_OUT_U_HB2	TX_OUT_U_HB2	A-RTR	B22	DNC	DNC	NDR
A21	TX_OUT_U_HB1	TX_OUT_U_HB1	A-RTR	B23	TX_OUT_G_LB	TX_OUT_G_LB	A-RTR
A22	TX_OUT_G_HB	LNA_OUT_C_HB	A-RTR	C1	EBI2_A_D_3	EBI2_A_D_3	B-EBI
A23	DNC	DNC	NDR	C2	EBI2_A_D_5	EBI2_A_D_5	B-EBI
B1	EBI2_A_D_4	EBI2_A_D_4	B-EBI	C3	VDD_CORE	VDD_CORE	PWR
B2	VDD_CORE	VDD_CORE	PWR	C4	EBI2_A_D_11	EBI2_A_D_11	B-EBI

Table 2-1 QSC6240/QSC6270 pin assignments – listed in alphanumeric order (continued)

Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]	Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]
C5	EBI2_A_D_13	EBI2_A_D_13	B-EBI	E3	EBI2_A_D_7	EBI2_A_D_7	B-EBI
C6	EBI2_LB_N	EBI2_LB_N	B-EBI	E5	EBI2_CS1_N	EBI2_CS1_N	B-EBI
C7	LCD_CS_N	LCD_CS_N	B-EBI	E6	EBI2_WE_N	EBI2_WE_N	B-EBI
C8	GPIO_22 AUX_PCM_CLK	Configurable I/O	B-GPIO B-CON	E7	GPIO_50 SPI_MISO_DATA	Configurable I/O	B-GPIO B-CON
C9	GPIO_43 NFC_IRQ ETM_PIPESTATA2	Configurable I/O	B-GPIO B-CON B-ETM	E8	GPIO_53 SPI_CLK	Configurable I/O	B-GPIO B-CON
C10	UART1_TXD	UART1_TXD	B-CON	E9	GPIO_21 AUX_PCM_DOUT	Configurable I/O	B-GPIO B-CON
C11	GPIO_25 CAMIF_HSYNC ETM_PIPESTATB2	Configurable I/O	B-GPIO B-CAM B-ETM	E10	UART1_RXD	UART1_RXD	B-CON
C12	GPIO_30 CAMIF_DATA_2 ETM_TRACE_PKTB0	Configurable I/O	B-GPIO B-CAM B-ETM	E11	GPIO_26 CAMIF_VSYNC ETM_TRACESYNCB	Configurable I/O	B-GPIO B-CAM B-ETM
C13	GPIO_33 CAMIF_DATA_5 ETM_TRACE_PKTB3	Configurable I/O	B-GPIO B-CAM B-ETM	E12	GPIO_27 CAMIF_DISABLE ETM_MODE_INT	Configurable I/O	B-GPIO B-CAMIF B-ETM
C14	GPIO_37 CAMIF_DATA_9 ETM_TRACE_PKTB7	Configurable I/O	B-GPIO B-CAM B-ETM	E13	GPIO_29 CAMIF_DATA_1 ETM_PIPESTATB0	Configurable I/O	B-GPIO B-CAM B-ETM
C15	GPIO_0 GP_PDM_0 ETM_TRACECLK	Configurable I/O	B-GPIO B-IOF B-ETM	E14	GPIO_24 CAMIF_PCLK	Configurable I/O	B-GPIO B-CAM
C16	GPIO_63 PA_RANGE1	Configurable I/O	B-GPIO B-IOF	E15	GPIO_58 GPS_ADCQ	Configurable I/O	B-GPIO B-IOF
C17	GPIO_64 PA_RANGE0	Configurable I/O	B-GPIO B-IOF	E16	GND_A_RF	GND_A_RF	GND
C18	VDD_RFTX	VDD_RFTX	PWR	E17	GND_A_RF	GND_A_RF	GND
C19	VDD_RFTX	VDD_RFTX	PWR	E18	VDD_RFTX	VDD_RFTX	PWR
C20	VDD_RFA	VDD_RFA	PWR	E19	GND_A_RF	GND_A_RF	GND
C21	GND_A_RF	GND_A_RF	GND	E21	GND_A_RF	GND_A_RF	GND
C22	GND_A_RF	GND_A_RF	GND	E22	GND_A_RF	GND_A_RF	GND
C23	PWR_DET_IN	PWR_DET_IN	A-RTR	E23	GND_A_RF	GND_A_RF	GND
D1	EBI2_A_D_1	EBI2_A_D_1	B-EBI	F1	EBI2_UB_N	EBI2_UB_N	B-EBI
D2	EBI2_A_D_2	EBI2_A_D_2	B-EBI	F2	EBI2_OE_N	EBI2_OE_N	B-EBI
D3	EBI2_A_D_8	EBI2_A_D_8	B-EBI	F3	EBI1_CKE_0	EBI1_CKE_0	B-EBI
D21	VDD_RFTX	VDD_RFTX	PWR	F5	LCD_EN	LCD_EN	B-EBI
D22	VDD_RFTX	VDD_RFTX	PWR	F6	GND_DIG	GND_DIG	GND
D23	VDD_RFTX	VDD_RFTX	PWR	F7	GPIO_51 SPI_MOSI_DATA	Configurable I/O	B-GPIO B-CON
E1	EBI2_A_D_0	EBI2_A_D_0	B-EBI	F8	GPIO_52 SPI_CS_N	Configurable I/O	B-GPIO B-CON
E2	EBI2_CS0_N	EBI2_CS0_N	B-EBI	F9	RESOUT_N	RESOUT_N	B-IOF

Table 2-1 QSC6240/QSC6270 pin assignments – listed in alphanumeric order (continued)

Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]	Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]
F10	GPIO_7 UART1_CTS_N	Configurable I/O	B-GPIO B-CON	H12	GPIO_41 HEADSET_DET_N GP_CLK ETM_TRACESYNCA	Configurable I/O	B-GPIO B-CON B-IOF B-ETM
F11	GPIO_8 UART1_RFR_N	Configurable I/O	B-GPIO B-CON	H13	GND_DIG	GND_DIG	GND
F12	GPIO_32 CAMIF_DATA_4 ETM_TRACE_PKTB2	Configurable I/O	B-GPIO B-CAM B-ETM	H14	GND_A_RF	GND_A_RF	GND
F13	GPIO_35 CAMIF_DATA_7 ETM_TRACE_PKTB5	Configurable I/O	B-GPIO B-CAM B-ETM	H15	DNC	DNC	NDR
F14	GPIO_55 GPS_ADCI	Configurable I/O	B-GPIO B-IOF	H16	DNC	DNC	NDR
F15	GND_DIG	GND_DIG	GND	H18	VDD_RFA	VDD_RFA	PWR
F16	GND_A_RF	GND_A_RF	GND	H19	VDD_RFRX	VDD_RFRX	PWR
F17	GND_A_RF	GND_A_RF	GND	H21	GND_A_RF	GND_A_RF	GND
F18	GND_A_RF	GND_A_RF	GND	H22	RX_IN_G_LBM	RX_IN_G_LBM	A-RTR
F19	GND_A_RF	GND_A_RF	GND	H23	RX_IN_U/G_HB1M	RX_IN_U/G_HB1M	A-RTR
F21	GND_A_RF	GND_A_RF	GND	J1	EBI1_DQS_1	EBI1_DQS_1	B-EBI
F22	GND_A_RF	GND_A_RF	GND	J2	EBI1_DQM_1	EBI1_DQM_1	B-EBI
F23	RX_IN_G_HBM	RX_IN_G_HBM	A-RTR	J3	VDD_P1	VDD_P1	PWR
G1	EBI1_A_D_14	EBI1_A_D_14	B-EBI	J5	EBI1_CS1_N	EBI1_CS1_N	B-EBI
G2	EBI1_A_D_15	EBI1_A_D_15	B-EBI	J6	EBI1_ADV_N	EBI1_ADV_N	B-EBI
G3	VDD_P1	VDD_P1	PWR	J8	GND_DIG	GND_DIG	GND
G5	EBI1_OE_N	EBI1_OE_N	B-EBI	J9	GND_DIG	GND_DIG	GND
G6	GPIO_23 MDP_VSYNC_P FM_INT	Configurable I/O	B-GPIO B-CON B-CON	J10	GND_DIG	GND_DIG	GND
G18	GND_A_RF	GND_A_RF	GND	J11	GND_DIG	GND_DIG	GND
G19	VDD_RFRX	VDD_RFRX	PWR	J12	GND_DIG	GND_DIG	GND
G21	VDD_RFRX	VDD_RFRX	PWR	J13	GND_DIG	GND_DIG	GND
G22	GND_A_RF	GND_A_RF	GND	J14	GND_A_RF	GND_A_RF	GND
G23	RX_IN_G_HBP	RX_IN_G_HBP	A-RTR	J15	DNC	DNC	NDR
H1	EBI1_A_D_13	EBI1_A_D_13	B-EBI	J16	DNC	DNC	NDR
H2	EBI1_A_D_12	EBI1_A_D_12	B-EBI	J18	GND_A_RF	GND_A_RF	GND
H3	EBI1_A_D_24	EBI1_A_D_24	B-EBI	J19	VDD_RFRX	VDD_RFRX	PWR
H5	EBI1_WE_N	EBI1_WE_N	B-EBI	J21	GND_A_RF	GND_A_RF	GND
H6	EBI1_CS0_N	EBI1_CS0_N	B-EBI	J22	RX_IN_G_LBP	RX_IN_G_LBP	A-RTR
H9	GND_DIG	GND_DIG	GND	J23	RX_IN_U/G_HB1P	RX_IN_U/G_HB1P	A-RTR
H10	GPIO_44 ETM_MODE_CS_N	Configurable I/O	B-GPIO B-ETM	K1	EBI1_A_D_11	EBI1_A_D_11	B-EBI
H11	GND_DIG	GND_DIG	GND	K2	EBI1_A_D_10	EBI1_A_D_10	B-EBI

Table 2-1 QSC6240/QSC6270 pin assignments – listed in alphanumeric order (continued)

Pin #	Pin name	Pin function(s)	Functional group(s) [1 2]		Pin #	Pin name	Pin function(s)	Functional group(s) [1 2]
K3	EBI1_A_D_27	EBI1_A_D_27	B-EBI		M1	EBI1_A_D_7	EBI1_A_D_7	B-EBI
K5	EBI1_M_CLK	EBI1_M_CLK	B-EBI		M2	EBI1_A_D_6	EBI1_A_D_6	B-EBI
K6	EBI1_A_D_31	EBI1_A_D_31	B-EBI		M3	EBI1_A_D_26	EBI1_A_D_26	B-EBI
K8	GND_DIG	GND_DIG	GND		M5	EBI1_A_D_29	EBI1_A_D_29	B-EBI
K9	GND_DIG	GND_DIG	GND		M6	EBI1_A_D_30	EBI1_A_D_30	B-EBI
K10	VDD_CORE	VDD_CORE	PWR		M8	GND_DIG	GND_DIG	GND
K11	VDD_CORE	VDD_CORE	PWR		M9	GND_DIG	GND_DIG	GND
K12	VDD_CORE	VDD_CORE	PWR		M10	GND_DIG	GND_DIG	GND
K13	VDD_CORE	VDD_CORE	PWR		M11	GND_DIG	GND_DIG	GND
K14	GND_A_RF	GND_A_RF	GND		M12	GND_DIG	GND_DIG	GND
K15	GND_A_RF	GND_A_RF	GND		M13	GND_DIG	GND_DIG	GND
K16	GND_A_RF	GND_A_RF	GND		M14	GND_A_RF	GND_A_RF	GND
K18	VDD_RFRX	VDD_RFRX	PWR		M15	GND_A_RF	GND_A_RF	GND
K19	VDD_RFRX	VDD_RFRX	PWR		M16	GND_A_RF	GND_A_RF	GND
K21	VDD_RFRX	VDD_RFRX	PWR		M18	VREG_CDC2	VREG_CDC2	P-OVR
K22	RX_IN_U/G_LBM	RX_IN_U/G_LBM	A-RTR		M19	GND_A_RF	GND_A_RF	GND
K23	RX_IN_U_HB2M	RX_IN_U_HB2M	A-RTR		M21	CCOMP	CCOMP	A-HAC
L1	EBI1_A_D_9	EBI1_A_D_9	B-EBI		M22	GND_A_RF	GND_A_RF	GND
L2	EBI1_A_D_8	EBI1_A_D_8	B-EBI		M23	GND_A_RF	GND_A_RF	GND
L3	VDD_P1	VDD_P1	PWR		N1	EBI1_A_D_5	EBI1_A_D_5	B-EBI
L5	EBI1_M_CLK_N	EBI1_M_CLK_N	B-EBI		N2	EBI1_A_D_4	EBI1_A_D_4	B-EBI
L6	EBI1_A_D_25	EBI1_A_D_25	B-EBI		N3	VDD_P1	VDD_P1	PWR
L8	GND_DIG	GND_DIG	GND		N5	EBI1_A_D_28	EBI1_A_D_28	B-EBI
L9	GND_DIG	GND_DIG	GND		N6	EBI1_A_D_23	EBI1_A_D_23	B-EBI
L10	VDD_CORE	VDD_CORE	PWR		N8	GND_DIG	GND_DIG	GND
L11	VDD_CORE	VDD_CORE	PWR		N9	GND_DIG	GND_DIG	GND
L12	VDD_CORE	VDD_CORE	PWR		N10	GND_DIG	GND_DIG	GND
L13	VDD_CORE	VDD_CORE	PWR		N11	GND_DIG	GND_DIG	GND
L14	GND_A_RF	GND_A_RF	GND		N12	GND_DIG	GND_DIG	GND
L15	GND_A_RF	GND_A_RF	GND		N13	GND_DIG	GND_DIG	GND
L16	GND_A_RF	GND_A_RF	GND		N14	GND_A_RF	GND_A_RF	GND
L18	GND_A_RF	GND_A_RF	GND		N15	GND_A_RF	GND_A_RF	GND
L19	GND_A_RF	GND_A_RF	GND		N16	GND_A_RF	GND_A_RF	GND
L21	GND_A_RF	GND_A_RF	GND		N18	VDD_RFA	VDD_RFA	PWR
L22	RX_IN_U/G_LBP	RX_IN_U/G_LBP	A-RTR		N19	EAROP	EAROP	A-HAC
L23	RX_IN_U_HB2P	RX_IN_U_HB2P	A-RTR		N21	MIC1N	MIC1N	A-HAC

Table 2-1 QSC6240/QSC6270 pin assignments – listed in alphanumeric order (continued)

Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]		Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]
N22	MIC2P	MIC2P	A-HAC		R18	GND_A_RF	GND_A_RF	GND
N23	LINE_IN_L_P	LINE_IN_L_P	A-HAC		R19	HPH_VNEG	HPH_VNEG	A-HAC
P1	EBI1_DQS_0	EBI1_DQS_0	B-EBI		R21	LINE_OUT_L_P	LINE_OUT_L_P	A-HAC
P2	EBI1_DQM_0	EBI1_DQM_0	B-EBI		R22	HKAIN1	HKAIN1	A-HAC
P3	EBI1_A_D_19	EBI1_A_D_19	B-EBI		R23	VDD_RFA	VDD_RFA	PWR
P5	EBI1_CKE_1	EBI1_CKE_1	B-EBI		T1	EBI1_A_D_1	EBI1_A_D_1	B-EBI
P6	EBI1_A_D_21	EBI1_A_D_21	B-EBI		T2	EBI1_A_D_0	EBI1_A_D_0	B-EBI
P8	GND_DIG	GND_DIG	GND		T3	EBI1_A_D_17	EBI1_A_D_17	B-EBI
P9	GND_DIG	GND_DIG	GND		T5	EBI1_A_D_20	EBI1_A_D_20	B-EBI
P10	GND_DIG	GND_DIG	GND		T6	EBI1_A_D_18	EBI1_A_D_18	B-EBI
P11	GND_DIG	GND_DIG	GND		T8	MODE_3	MODE_3	B-INT
P12	GND_DIG	GND_DIG	GND		T9	MODE_2	MODE_2	B-INT
P13	GND_DIG	GND_DIG	GND		T10	MODE_1	MODE_1	B-INT
P14	GND_A_RF	GND_A_RF	GND		T11	DNC	DNC	NDR
P15	GND_A_RF	GND_A_RF	GND		T12	GND_MPLL	GND_MPLL	GND
P16	GND_A_RF	GND_A_RF	GND		T13	VREG_MPLL	VREG_MPLL	P-OVR
P18	HPH_OUT_R_N	HPH_OUT_R_N	A-HAC		T14	GPIO_65 TRK_LO_ADJ GP_CLK GP_PDM_1	Configurable I/O	B-GPIO B-IOF B-IOF B-IOF
P19	EARON	EARON	A-HAC		T15	GND_DIG	GND_DIG	GND
P21	MIC1P	MIC1P	A-HAC		T16	GND_DIG	GND_DIG	GND
P22	MIC2N	MIC2N	A-HAC		T18	VDD_RFA	VDD_RFA	PWR
P23	LINE_IN_R_N	LINE_IN_R_N	A-HAC		T19	HPH_OUT_L_P	HPH_OUT_L_P	A-HAC
R1	EBI1_A_D_3	EBI1_A_D_3	B-EBI		T21	LINE_OUT_R_N	LINE_OUT_R_N	A-HAC
R2	EBI1_A_D_2	EBI1_A_D_2	B-EBI		T22	VTHERM_IN	VTHERM_IN	A-HAC
R3	VDD_P1	VDD_P1	PWR		T23	GND_A_RF	GND_A_RF	GND
R5	EBI2_A_D_22	EBI2_A_D_22	B-EBI		U1	VDD_CORE	VDD_CORE	PWR
R6	GND_DIG	GND_DIG	GND		U2	VDD_CORE	VDD_CORE	PWR
R8	GND_DIG	GND_DIG	GND		U3	EBI1_A_D_16	EBI1_A_D_16	B-EBI
R9	GND_DIG	GND_DIG	GND		U5	RTCK	RTCK	B-INT
R10	GND_DIG	GND_DIG	GND		U6	TCK	TCK	B-INT
R11	GND_DIG	GND_DIG	GND		U18	VIB_DRV_N	VIB_DRV_N	P-IUI
R12	GND_DIG	GND_DIG	GND		U19	HSED_BIAS	HSED_BIAS	A-HAC
R13	GND_DIG	GND_DIG	GND		U21	VDD_TCXO	VDD_TCXO	PWR
R14	GND_DIG	GND_DIG	GND		U22	VREG_TCXO	VREG_TCXO	P-OVR
R15	GND_DIG	GND_DIG	GND		U23	XTAL_19M_OUT	XTAL_19M_OUT	P-GH
R16	GND_DIG	GND_DIG	GND		V1	GPIO_17 KEYSENSE1_N ETM_TRACE_PKTA6	Configurable I/O	B-GPIO B-CON B-ETM

Table 2-1 QSC6240/QSC6270 pin assignments – listed in alphanumeric order (continued)

Pin #	Pin name	Pin function(s)	Functional group(s) [1 2]	Pin #	Pin name	Pin function(s)	Functional group(s) [1 2]
V2	GPIO_16 KEYSENSE2_N ETM_TRACE_PKTA5	Configurable I/O	B-GPIO B-CON B-ETM	W7	GPIO_68 SDCC2_DATA1	Configurable I/O	B-GPIO B-CON
V3	TDO	TDO	B-INT	W8	GPIO_69 SDCC2_DATA2	Configurable I/O	B-GPIO B-CON
V5	TDI	TDI	B-INT	W9	GPIO_70 SDCC2_DATA3	Configurable I/O	B-GPIO B-CON
V6	TRST_N	TRST_N	B-INT	W10	XO_OUT_GP1	XO_OUT_GP1	P-GH
V7	GPIO_66 SDCC2_CMD	Configurable I/O	B-GPIO B-CON	W11	XO_EN_GP1	XO_EN_GP1	P-GH
V8	GPIO_71 SDCC2_CLK	Configurable I/O	B-GPIO B-CON	W12	BAT_FET_N	BAT_FET_N	P-IPM
V9	RSVRD	RSVRD	NDR	W13	VREG_MSMP	VREG_MSMP	P-OVR
V10	PON_RESET_N	PON_RESET_N	P-IUI	W14	VREG_USIM	VREG_USIM	P-OVR
V11	PS_HOLD	PS_HOLD	P-IUI	W15	VREG_MSME	VREG_MSME	P-OVR
V12	SLEEP_CLK	SLEEP_CLK	P-GH	W16	VREG_RFA	VREG_RFA	P-OVR
V13	VDD_EFUSE	VDD_EFUSE	PWR	W17	MPP3	multipurpose pin	P-MPP
V14	VREG_USB_3P3	VREG_USB_3P3	P-OVR	W18	MPP2	multipurpose pin	P-MPP
V15	VREG_USB_2P6	VREG_USB_2P6	P-OVR	W19	G_PA_ON_0	G_PA_ON_0	P-IUI
V16	VDD_USB	VDD_USB	PWR	W21	VREG_RFRX2	VREG_RFRX2	P-OVR
V17	MPP4	multipurpose pin	P-MPP	W22	XO_ADC_IN	XO_ADC_IN	P-GH
V18	XO_OUT_GP2	XO_OUT_GP2	P-GH	W23	XO_ADC_REF	XO_ADC_REF	P-GH
V19	G_PA_ON_1	G_PA_ON_1	P-IUI	Y1	GPIO_13 KEYPAD_0 ETM_TRACE_PKTA2	Configurable I/O	B-GPIO B-CON B-ETM
V21	VREG_RFTX2	VREG_RFTX2	P-OVR	Y2	GPIO_12 KEYPAD_1 ETM_TRACE_PKTA1	Configurable I/O	B-GPIO B-CON B-ETM
V22	GND_TCXO	GND_TCXO	GND	Y3	GPIO_10 KEYPAD_3 ETM_PIPESTATA0	Configurable I/O	B-GPIO B-CON B-ETM
V23	XTAL_19M_IN	XTAL_19M_IN	P-GH	Y21	U_PA_ON_2	U_PA_ON_2	P-IUI
W1	GPIO_18 KEYSENSE0_N ETM_TRACE_PKTA7	Configurable I/O	B-GPIO B-CON B-ETM	Y22	VREG_GP2	VREG_GP2	P-OVR
W2	GPIO_15 KEYSENSE3_N ETM_TRACE_PKTA4	Configurable I/O	B-GPIO B-CON B-ETM	Y23	GND_XO_ADC	GND_XO_ADC	GND
W3	GPIO_14 KEYSENSE4_N ETM_TRACE_PKTA3	Configurable I/O	B-GPIO B-CON B-ETM	AA1	GPIO_11 KEYPAD_2 ETM_TRACE_PKTA0	Configurable I/O	B-GPIO B-CON B-ETM
W5	TMS	TMS	B-INT	AA2	GPIO_9 KEYPAD_4 ETM_PIPESTATA1	Configurable I/O	B-GPIO B-CON B-ETM
W6	GPIO_67 SDCC2_DATA0	Configurable I/O	B-GPIO B-CON	AA3	GPIO_74 BT_WAKES_MSM	Configurable I/O	B-GPIO B-CON

Table 2-1 QSC6240/QSC6270 pin assignments – listed in alphanumeric order (continued)

Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]	Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]
AA4	GPIO_2 SDCC1_DATA0	Configurable I/O	B-GPIO B-CON	AB9	REF_GND	REF_GND	P-OVR
AA5	GPIO_3 SDCC1_DATA1	Configurable I/O	B-GPIO B-CON	AB10	GND_A_RF	GND_A_RF	GND
AA6	GPIO_5 SDCC1_DATA3	Configurable I/O	B-GPIO B-CON	AB11	GND_5V	GND_5V	GND
AA7	GPIO_72 MUSIM_DM	Configurable I/O	B-GPIO B-CON	AB12	VBAT	VBAT	P-IPM
AA8	USB_ID	USB_ID	B-CON	AB13	VPH_PWR	VPH_PWR	P-IPM
AA9	R_REF_EXT	R_REF_EXT	P-GH	AB14	VCHG	VCHG	P-IPM
AA10	KPD_PWR_N	KPD_PWR_N	P-IUI	AB15	VREG_RF1	VREG_RF1	P-OVR
AA11	VOUT_PA_CTL	VOUT_PA_CTL	P-IUI	AB16	GND_RF	GND_RF	GND
AA12	VDD_PAD_SIM	VDD_PAD_SIM	PWR	AB17	VREG_RF2	VREG_RF2	P-OVR
AA13	VPH_PWR	VPH_PWR	P-IPM	AB18	NCP_CTC1	NCP_CTC1	P-OVR
AA14	VCHG	VCHG	P-IPM	AB19	NCP_CTC2	NCP_CTC2	P-OVR
AA15	VDD_EBI_RFA	VDD_EBI_RFA	PWR	AB20	SPKR_OUT_P	SPKR_OUT_P	P-IUI
AA16	VDD_PLL_CDC	VDD_PLL_CDC	PWR	AB21	SPKR_OUT_M	SPKR_OUT_M	P-IUI
AA17	VDD_RX2_TX2	VDD_RX2_TX2	PWR	AB22	U_PA_ON_1	U_PA_ON_1	P-IUI
AA18	VCOIN	VCOIN	P-IPM	AB23	XTAL_32K_OUT	XTAL_32K_OUT	P-GH
AA19	VDD_C_NCP	VDD_C_NCP	PWR	AC1	GND_DIG	GND_DIG	GND
AA20	VDD_GP	VDD_GP	PWR	AC2	MODE_0	MODE_0	B-INT
AA21	U_PA_ON_0	U_PA_ON_0	P-IUI	AC3	GPIO_77 WLAN_PWR_DN	Configurable I/O	B-GPIO B-CON
AA22	VREG_GP1	VREG_GP1	P-OVR	AC4	GPIO_45 USIM_RESET	Configurable I/O	B-GPIO B-CON
AA23	XTAL_32K_IN	XTAL_32K_IN	P-GH	AC5	GPIO_47 USIM_DATA	Configurable I/O	B-GPIO B-CON
AB1	GPIO_76 BT_PWR_ON	Configurable I/O	B-GPIO B-CON	AC6	USB_DM	USB_DM	B-CON
AB2	GPIO_75 MSM_WAKES_BT	Configurable I/O	B-GPIO B-CON	AC7	USB_DP	USB_DP	B-CON
AB3	GPIO_1 SDCC1_CMD	Configurable I/O	B-GPIO B-CON	AC8	GND_DIG	GND_DIG	GND
AB4	GPIO_6 SDCC1_CLK	Configurable I/O	B-GPIO B-CON	AC9	REF_BYP	REF_BYP	P-OVR
AB5	GPIO_4 SDCC1_DATA2	Configurable I/O	B-GPIO B-CON	AC10	REF_ISET	REF_ISET	P-OVR
AB6	GPIO_46 USIM_CLK	Configurable I/O	B-GPIO B-CON	AC11	VREG_5V	VREG_5V	P-OVR
AB7	USB_VBUS	USB_VBUS	P-IPM	AC12	VSW_5V	VSW_5V	P-OVR
AB8	GPIO_73 MUSIM_DP	Configurable I/O	B-GPIO B-CON	AC13	VREF_THERM	VREF_THERM	P-GH

Table 2-1 QSC6240/QSC6270 pin assignments – listed in alphanumeric order (continued)

Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]		Pin #	Pin name	Pin function(s)	Functional group(s) [1] [2]
AC14	VDD_CMN	VDD_CMN	PWR		AC19	GND_C_NCP	GND_C_NCP	GND
AC15	VSW_RF1	VSW_RF1	P-OVR		AC20	VREG_NCP	VREG_NCP	P-OVR
AC16	VDD_RF	VDD_RF	PWR		AC21	VDD_SPKR	VDD_SPKR	PWR
AC17	VSW_RF2	VSW_RF2	P-OVR		AC22	GND_SPKR	GND_SPKR	GND
AC18	VSW_MSMC	VSW_MSMC	P-OVR		AC23	GND_A_RF	GND_A_RF	GND

[1] The functional groups are:
A-RTR = Analog/RF – RF transceiver functions (Chapter 3)
A-HAC = Analog/RF – housekeeping ADC and audio codec (Chapter 4)
B-EBI = Baseband – external bus interface (EBI1 and EBI2 - Chapter 6)
B-CAM = Baseband – camera interface (Chapter 8)
B-CON = Baseband – connectivity (Chapter 9)
B-ETM = Baseband – embedded trace macrocell (Chapter 11)
B-GPIO = Baseband – general-purpose input/output (configurable - Chapter 10)
B-INT = Baseband – internal function (Chapter 11)
B-IOF = Baseband – interfaces with other functions (Chapter 12)
P-GH = Power management – general housekeeping (Chapter 15)
P-IUI = Power management – interfaces (IC-level and user-level interfaces - Chapter 16)
P-IPM = Power management – input power management (Chapter 13)
P-MPP = Power management – multipurpose pins (configurable - Chapter 16)
P-OVR = Power management – output voltage regulation (Chapter 14)
PWR = DC input power supply voltage (Chapter 17)
GND = Ground (Chapter 17)
NDR = No internal connection (NC), do not connect (DNC), or reserved (RSRVD)
Other functional groups with no external pins: baseband processors (Chapter 5), air interfaces (Chapter 7)

[2] More detailed descriptions of all I/Os appear within the chapters identified in note 1.

3 RF Signal Paths and LO Circuits

The QSC62x0 device includes most of the active RF and LO functions for low-cost, multiband, multimode wireless products. This chapter describes these on-chip functions in detail and defines their interface requirements. It is organized according to function: RF receive signal paths, Rx LO circuits, RF transmit signal paths, and Tx LO circuits.

3.1 RF receive signal paths

The QSC62x0 device provides five low noise amplifier (LNA) circuits capable of supporting tri-band UMTS (WCDMA) and quad-band GSM applications. The five LNAs are supported by two RF-to-baseband receiver paths: three are multiplexed to drive one quadrature downconverter; the other two are multiplexed to drive a second downconverter. The UMTS bands always use the set of three LNAs, while the GSM bands might use both sets of LNAs depending upon the application. The option to share LNAs between the UMTS and GSM bands is discussed in Section 3.1.2.

All QSC receiver paths are discussed in this section, with separate sections for UMTS and GSM receivers.

3.1.1 UMTS receivers

The antenna collects the base station forward-link signal and radiates the phone's reverse-link signal. In the example of a multiband, multimode phone (Figure 3-1), a switch routes the antenna signals to one of the three UMTS Rx/Tx paths, each beginning with its own band-specific duplexer that separates that band's receive and transmit paths.

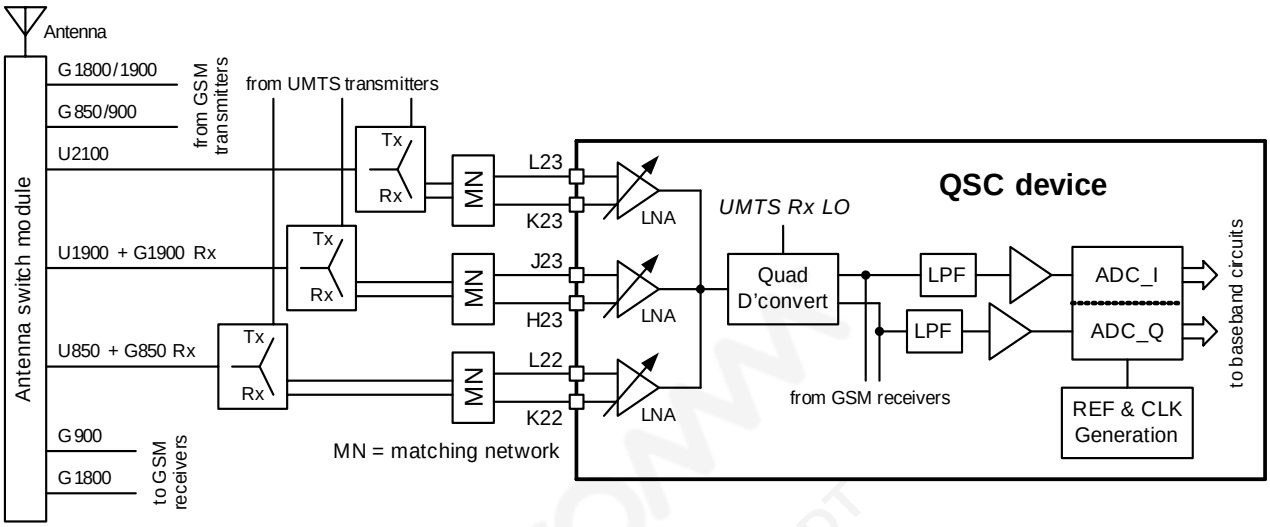


Figure 3-1 QSC UMTS receiver signal paths functional diagram

Each UMTS duplexer provides a differential output signal that is compatible with its QSC LNA input. The duplexer-to-LNA interface requires a differential matching network (MN) that optimizes the power transfer into the LNA. Although there are three UMTS LNAs, only one is active at a time. The active gain-stepped LNA output drives a shared quadrature downconverter directly — an off-chip inter-stage filter is not required. The elimination of this filter is achieved by a combination of factors:

- New on-chip QSC processing
- Higher performance achieved by the differential duplexer-to-LNA interface
- Greater duplexer suppression of Tx leakage

The downconverter's RF circuitry includes another gain-stepped amplifier that supplements the LNA gain steps to further extend the receiver dynamic range. The downconverter translates the active LNA's RF signal directly to baseband, producing two analog outputs: in-phase (I) and quadrature (Q). The UMTS baseband signals are routed to lowpass filters whose passband and stopband characteristics are optimized for the active WCDMA waveform. Both filter outputs are buffered to drive their analog-to-digital converters for digitization. The digital baseband outputs are routed to QSC baseband circuits for further processing.

The Rx LO signal is delivered to the downconverter circuits from the LO generation and distribution circuits as described in Section 3.2.

3.1.2 GSM receivers

There are two recommended GSM receiver path configurations; both are shown in Figure 3-2. The configuration shown on top allows the GSM850 and GSM1900 bands to share the UMTS850 and UMTS1900 paths for US applications and uses four LNA inputs to support quad-band GSM operation. The lower example uses two LNAs for quad-band GSM operation (one low band and one high band), with two-way SAW filters between the antenna switch and the QSC inputs for each band type.

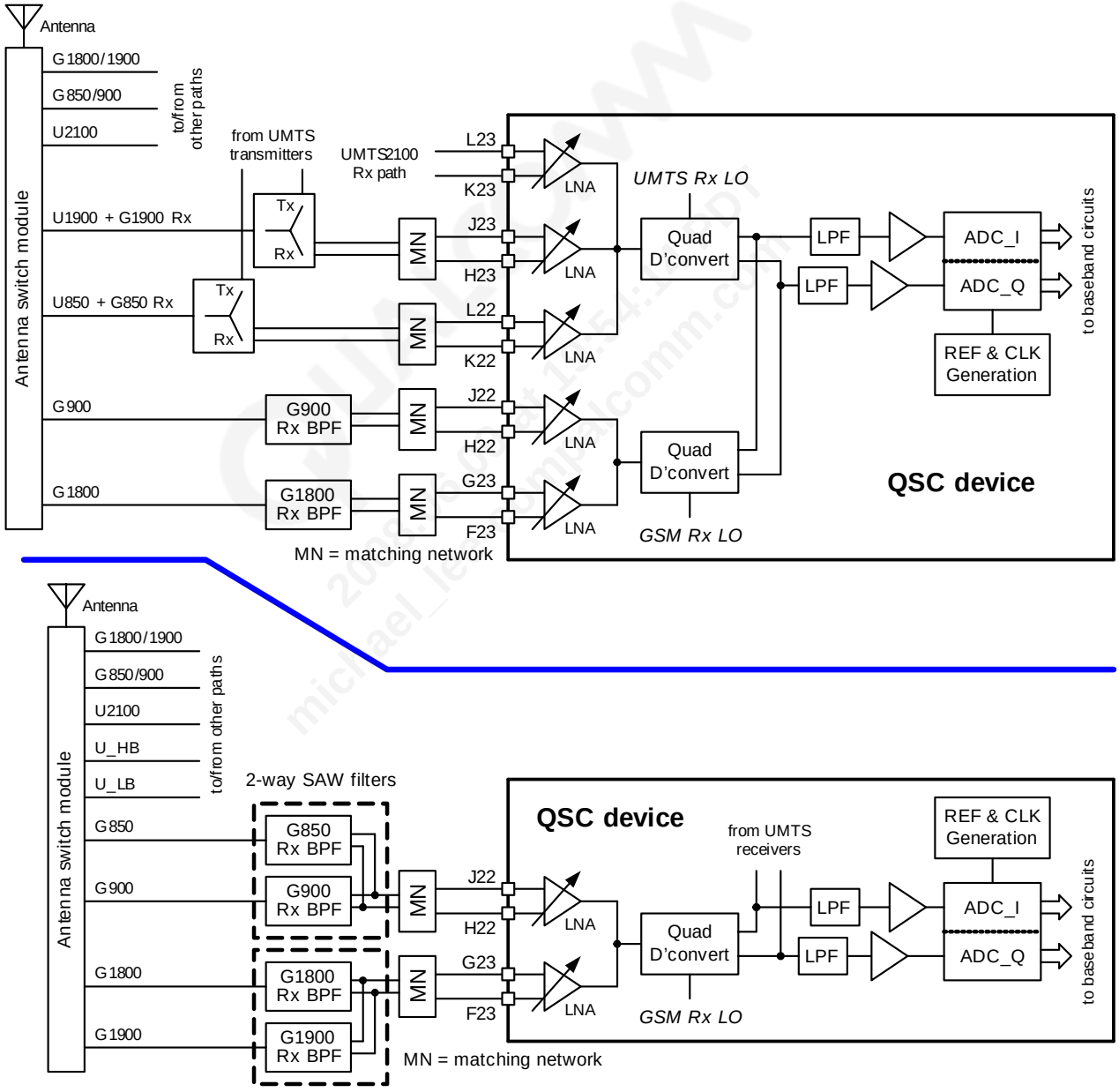


Figure 3-2 QSC GSM receiver signal paths functional diagram

Shared UMTS/GSM configuration (four GSM LNAs)

In this configuration, the GSM850 receive path shares the UMTS850 receiver front-end path (including LNA). Beginning at the antenna switch output, the GSM signal is routed through the UMTS850 duplexer to the shared LNA input at pins L22 and K22. Likewise, the GSM1900 receive path shares the UMTS1900 front-end, including pins J23 and H23.

The GSM900 and GSM1800 bands have dedicated receive paths from the antenna switch outputs to the QSC LNA inputs. Each band has its own band-select filter that drives its LNA input.

All four GSM bands include input filtering: the 850 and 1900 bands share the UMTS duplexer filtering, while the 900 and 1800 bands have dedicated bandpass filters. The filter functions suppress out-of-band received signals and the handset's GSM transmitter leakage. Transmit power suppression must be adequate to avoid overdriving the GSM Rx chain. Like the UMTS paths, the GSM paths use a differential configuration into their LNAs, and thus require differential matching networks.

The internal GSM receivers are functionally identical to the UMTS receivers: although there are multiple GSM LNAs, only one is active at a time. The active gain-stepped LNA output drives a shared quadrature downconverter directly — an off-chip inter-stage filter is not required. The elimination of this filter is achieved by a combination of factors:

- New on-chip QSC processing
- Higher performance achieved by the differential duplexer-to-LNA interface
- Greater duplexer suppression of Tx leakage

The downconverter's RF circuitry includes another gain-stepped amplifier that supplements the LNA gain steps to further extend the receiver dynamic range. The downconverter translates the active LNA's RF signal directly to baseband, producing two analog outputs: in-phase (I) and quadrature (Q). The GSM baseband signals drive lowpass filters whose passband and stopband characteristics are optimized for the active GSM waveform. Both filter outputs are buffered to drive their analog-to-digital converters for digitization. The digital baseband outputs are routed to QSC baseband circuits for further processing.

The Rx LO signal is delivered to the downconverter circuits from the LO generation and distribution circuits as described in Section 3.2.

Dedicated GSM configuration (two GSM LNAs)

In this configuration, the GSM850 and GSM1900 bands do not pass through the UMTS duplexers. Instead, the two GSM LNA inputs are shared: the GSM850 and GSM900 bands share the low-band GSM LNA, and the GSM1800 and GSM1900 bands share the high-band LNA. Four switch module outputs are required, each driving its own GSM Rx path. A two-way SAW filter takes the two low-band (or high-band) single-ended inputs from the antenna switch and provides one filtered, differential output that drives the appropriate QSC LNA input.

Beyond the LNA inputs, this GSM receiver configuration is identical to the paths described earlier for the shared UMTS/GSM configuration.

3.1.3 RF receiver features

- Optimization for tri-band UMTS plus quad-band GSM operation
  - Tri-band UMTS, with one low band and two high bands selected from:
    - Low band (select one)
      - BC5 (CELL, 869 to 894 MHz)
      - BC6 (JCELL, 875 to 885 MHz)
      - BC8 (EGSM, 925 to 960 MHz)
    - High band (select two)
      - BC1 (IMT, 2110 to 2170 MHz)
      - BC2 (PCS, 1930 to 1990 MHz)
      - BC3 (DCS, 1805 to 1880 MHz)
      - BC4 (AWS, 2110 to 2155 MHz)
      - BC9 (J1700, 1844.9 to 1879.9 MHz)
  - Quad-band GSM:
    - Low band
      - GSM850 (869 to 894 MHz) and GSM900 (925 to 960 MHz)
    - High band
      - DCS1800 (1805 to 1880 MHz) and PCS1900 (1930 to 1990 MHz)
- Support for UMTS and GSM voice and data operation
- Full downconversions from RF-to-analog baseband
- Receive signal path circuits (UMTS and GSM; low and high bands):
  - Five gain-stepped LNA circuits

    Off-chip inter-stage filters are **not** required.
  - Two RF-to-baseband quadrature downconverters
  - Baseband lowpass filters and buffers
  - Analog-to-digital converters; digital outputs to baseband circuits
- Complete Rx LO sources: local oscillator and PLL circuits
- All LO distribution circuits (UMTS and GSM) needed to drive the on-chip downconverters
- ADC reference and clock circuits

3.1.4 RF receiver connections

RF receiver connections are summarized in Table 3-1 and illustrated in Figure 3-3. Some components shown may not be required but all should be included in the initial layouts to allow matching flexibility and optimization.

Table 3-1 Analog/RF-RF transceiver functions [1]

Pin name/function	Pin #	Pin type [2]	Functional description
RX_IN_U/G_LBM	K22	AI	Low-band RF input; differential configuration with plus (P) and minus (M) pins; requires external match. Supported bands depend upon the application: ■ UMTS850 and GSM850 ■ UMTS850 only ■ UMTS900 only
RX_IN_U/G_LBP	L22	AI	
RX_IN_U/G_HB1M	H23	AI	High-band RF input; differential configuration with plus (P) and minus (M) pins; requires external match. Supported bands depend upon the application: ■ UMTS1900 and GSM1900 ■ UMTS1900 only ■ UMTS1800 only ■ UMTS J1700 only
RX_IN_U/G_HB1P	J23	AI	
RX_IN_U_HB2M	K23	AI	High-band RF input dedicated to UMTS2100 band; differential configuration with plus (P) and minus (M) pins; requires external match.
RX_IN_U_HB2P	L23	AI	
RX_IN_G_LBM	H22	AI	Low-band RF input; differential configuration with plus (P) and minus (M) pins; requires external match. Supported bands depend upon the application: ■ GSM900 only ■ GSM850 and GSM900
RX_IN_G_LBP	J22	AI	
RX_IN_G_HBM	F23	AI	High-band RF input; differential configuration with plus (P) and minus (M) pins; requires external match. Supported bands depend upon the application: ■ GSM1800 only ■ GSM1800 and GSM1900
RX_IN_G_HBP	G23	AI	

[1] The expected RF matching networks are shown in Figure 3-3. Some components may not be required but all should be included in the initial layouts to allow matching flexibility and optimization.

[2] Pin type is AI = analog input.

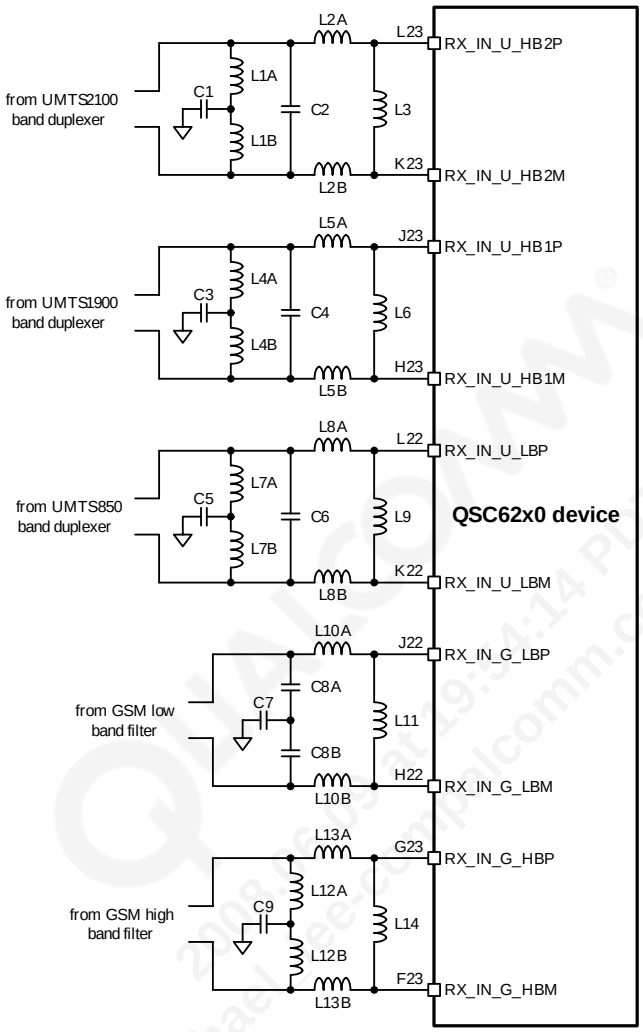


Figure 3-3 Expected UMTS and GSM RF Rx matching network topologies

UMTS (WCDMA) RF inputs

The three UMTS inputs are all connected to their duplexers through a differential matching network. Beginning at the duplexer output, the matching networks include:

- Shunt inductors with a center-tap capacitor that provides AC-ground
- A shunt capacitor across the differential pair
- A series inductor (the same value in each leg)
- A shunt inductor across the differential QSC input pins

The two legs of each differential match must be kept as equal as possible in amplitude and phase.

Some matching components shown may not be required but all should be included in the initial layouts to allow matching flexibility and optimization.

GSM RF inputs

The two dedicated GSM inputs are both connected to their band-select filters through a differential matching network. The low-band and high-band matching networks are slightly different: the low band begins with shunt capacitors with a center-tap capacitor for AC-ground, while the high band begins with shunt inductors and a center-tap capacitor. From this point forward, the two networks have the same configuration: a series inductor (the same value in each leg) and a shunt inductor across the differential QSC input pins. The two legs of each differential match must be kept as equal as possible in amplitude and phase.

Like the UMTS circuits, the GSM circuits include some matching components that may not be required but all should be included in the initial layouts to allow matching flexibility and optimization.

As mentioned earlier, the GSM850 and GSM1900 bands could share the UMTS850 and UMTS1900 paths. If so, the UMTS matching networks discussed earlier support those GSM bands as well.

3.2 Rx LO circuits

The QSC62x0 device integrates all of the frequency synthesizer functions that generate the UMTS and GSM receive LO signals (UHF local oscillator, PLL circuits, and loop filter), plus the distribution circuits that deliver the quadrature LO signals to the two downconverters.

The buffered 19.2 MHz TCXO or XO signal provides the synthesizer input (REF), the frequency reference to which the PLL is phase and frequency locked. The reference is divided to create a fixed frequency input to the phase detector, F_R. The other phase detector input (F_V) varies as the loop acquires a lock and is generated by dividing the local oscillator output frequency using the feedback path's counter. The closed-loop will force F_V to equal F_R when locked. If the loop is not locked, the error between F_V and F_R will create an error signal. This error signal is filtered by the loop filter and applied to the local oscillator, tuning the output frequency so that the error is decreased. Ultimately the loop forces the error to approach zero and the PLL is phase and frequency locked.

All loop variables — filtering, divider values, etc. — are internal and/or controlled by QSC software. The synthesizer performance is guaranteed without any adjustment by the handset designer.

The reference signal is generated by the handset 19.2 MHz oscillator as described in Section 15.3.1. This reference signal is shared between the Tx and Rx PLL circuits.

Status information from both the Rx and Tx PLLs is reported to the baseband circuits for software monitoring and verification of locked or unlocked conditions.

The UHF output of each receive LO synthesizer feeds a distribution network that drives its downconverter LO ports with the appropriate frequency and amplitude. The LO signals applied to the downconverter are differential with a quadrature phase relationship.

3.2.1 Rx LO features

The QSC62x0 Rx LO features include:

- Optimization for tri-band UMTS plus quad-band GSM Rx operation
- Fully integrated frequency synthesizer
  - UHF Rx LO
  - Feedback divider
  - Reference divider
  - Phase/frequency detector
  - Loop filter
- Synthesizer output that is processed to drive the quadrature downconverters
  - Frequency plans ensure that f_{LO} is not equal to f_{RF} (an important consideration for ZIF reception).
  - Applied downconverter inputs are differential with quadrature phase relationships.

3.2.2 Rx LO connections

All Rx LO interfaces are internal to the QSC device.

3.3 RF transmit signal paths

The QSC62x0 device supports UMTS and GSM transmissions with three UMTS outputs and two GSM outputs. The five transmitter paths share a common interface from the baseband circuits but have their own, dedicated output drivers. A single RF Tx power-detector circuit is integrated on-chip and supports all UMTS bands (only one is active at a time).

The QSC transmitter paths and Tx power-detector input are discussed in this section.

3.3.1 UMTS transmitters

The QSC62x0 device supports tri-band UMTS transmissions with three separate driver amplifier outputs; in fact, most Tx active circuits are contained within the device. All three UMTS transmit paths (Figure 3-4) begin with a single, shared analog baseband signal from the device's baseband circuits. The baseband signal is composed of two differential lines, one in-phase component and one quadrature-phase component. Each component is lowpass filtered and amplified to levels sufficient for driving the quadrature upconverters. There are two upconverters — one for low-band signals and one for high band — but only one upconverter is active at a time. The transmitter LO signals are generated by circuits described in Section 3.4 and delivered to the upconverter circuits at the correct frequency, with the proper phase relationship, and with adequate drive level.

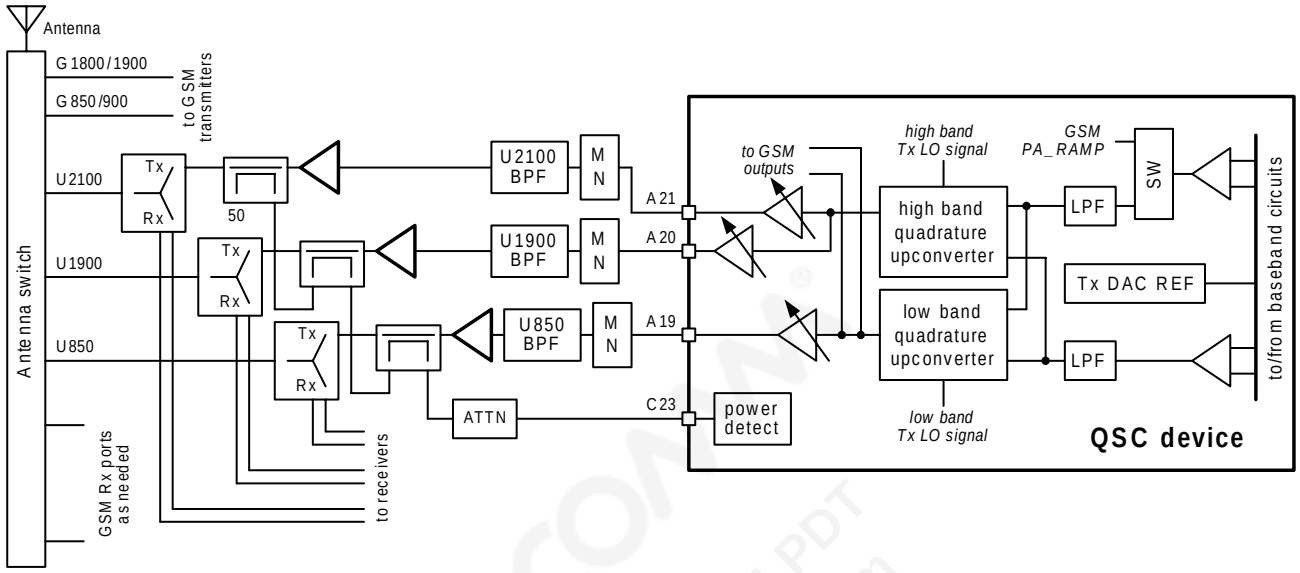


Figure 3-4 QSC UMTS transmitter signal paths functional diagram

The active upconverter's output is at the desired RF channel frequency and drives the QSC output stages. These RF circuits include multiple variable-gain stages that provide transmit AGC control. A greater than 81 dB gain control range is realized using information from the on-chip Tx power detector combined with a control signal from the baseband circuits. The wide range of driver amplifier output levels is achieved while supporting the WCDMA standard's requirements for ACLR, spurious emissions, Rx-band noise, etc.

The high-band driver amplifier output is followed by a SPDT switch that provides two high-band outputs, thereby fulfilling the tri-band UMTS requirement. The low-band output port is driven directly by its output amplifier. All three output ports are single-ended with 50 Ω nominal impedance. Each requires a matching network to interface with its bandpass filter.

Each of the three UMTS Tx output chains are functionally identical: the QSC Tx output drives the PA through the bandpass filter and a matching network; a directional coupler provides a sample of the PA output signal; the through path of the coupler is routed to the Tx port of the duplexer; the duplexer antenna port is connected to the antenna switch; and the switch is connected to the antenna.

The coupler outputs provide a low-level sample of the active transmitter's Tx power. An on-chip power-detector circuit provides a Tx power estimate that assists in setting the transmit gains and helps ensure that the maximum allowed output power is not exceeded. The three UMTS couplers (one for each band) use a daisy-chain configuration that allows them to share the single, on-chip power detector circuit.

3.3.2 GSM transmitters

The QSC62x0 device supports quad-band GSM transmissions with two separate dual-band driver amplifier outputs; in fact, most Tx active circuits are contained within the device. Both GSM transmit paths (Figure 3-5) begin with a single, shared analog baseband signal from the device's baseband circuits — the same interface and baseband circuits used by the UMTS transmitters. The GSM transmitters use the same quadrature upconverters as well — one for low band signals and one for high band — with just one active at a time. The transmitter LO signals are generated by circuits described in Section 3.4 and delivered to the upconverter circuits at the correct frequency, with the proper phase relationship, and with an adequate drive level.

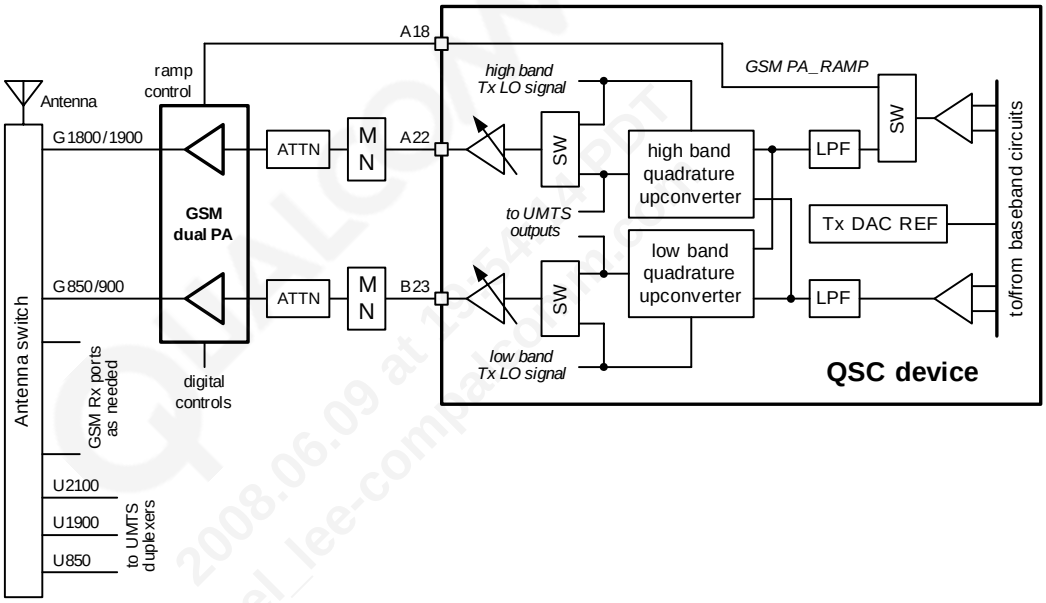


Figure 3-5 QSC GSM transmitter signal paths functional diagram

The SPDT switches at each driver amplifier input allow selection of the output signal: either the actual GSM signal from the upconverter or a test signal generated by the Tx LO synthesizer. The Tx output chain is functionally identical for both the low band and the high band: the power amplifier is driven by the QSC device through a matching network and a resistive attenuator; the PA output is routed to the antenna switch module whose output is connected to the antenna.

In addition to the through signal path, the QSC device also provides the PA ramp control signal that ensures smooth transitions while the transmitter is turned on and off for GSM's burst transmissions. The ramp signal is generated by one of the baseband circuit's Tx DACs. A switch after the baseband amplifier selects whether the DAC output signal is routed to GSM PA module for ramping the PA up or down, or to the transmitter signal path for data transmission.

3.3.3 WCDMA Tx power detector

The QSC62x0 device includes an integrated RF transmit power detector circuit that assists with UMTS Tx gain control and PA-mode selection. This function is included in Figure 3-4 and was discussed in Section 3.3.1.

3.3.4 RF transmitter features

The QSC62x0 RF transmitter features include:

- Optimization for tri-band WCDMA (UMTS) and quad-band GSM operation
  - Tri-band UMTS, with one low band and two high bands selected from:
    - Low band (select one)
      - BC5 (CELL, 824 to 849 MHz)
      - BC6 (JCELL, 830 to 840 MHz)
      - BC8 (EGSM, 880 to 915 MHz)
    - High band (select two)
      - BC1 (IMT, 1920 to 1980 MHz)
      - BC2 (PCS, 1850 to 1910 MHz)
      - BC3 (DCS, 1710 to 1785 MHz)
      - BC4 (AWS, 1710 to 1755 MHz)
      - BC9 (J1700, 1749.9 to 1784.9 MHz)
  - Quad-band GSM:
    - Low band
      - GSM850 (824 to 849 MHz) and GSM900 (880 to 915 MHz)
    - High band
      - DCS1800 (1710 to 1785 MHz) and PCS1900 (1850 to 1910 MHz)
- Support for UMTS and GSM voice and data operation
- Full upconversion from analog baseband to RF
- Transmit signal path circuits
  - Baseband interface with amplifiers and lowpass filters
  - Low-band and high-band zero-IF baseband-to-RF quadrature upconverters
  - Low-band and high-band RF AGC amplifiers and filters
  - Five driver amplifier outputs
    - Three UMTS (one low-band, two high-band)
    - Two GSM (one low-band, one high-band)
- UMTS transmit RMS power detector and supporting circuits

- GSM test and calibration circuits
- GSM PA ramp generation
- Complete Tx LO source: local oscillator and PLL circuits
- All LO distribution circuits needed to drive the on-chip upconverters
- Greater than 81 dB UMTS transmit power control range
- Power reduction features via baseband control that extends handset talk time:
  - Optimization for low DC power consumption versus RF output power level
  - Transmit puncturing, selective circuit powerdown, and gain control

3.3.5 RF transmitter connections

RF transmitter connections are summarized in Table 3-2 and illustrated in Figure 3-6.

Table 3-2 Analog/RF-RF transceiver functions

Pin name/function	Pin #	Pin type [1]	Functional description [2]
TX_OUT_U_LB	A19	AO	UMTS low-band driver amplifier RF output; requires external match.
TX_OUT_U_HB1	A21	AO	First of two UMTS high-band driver amplifier RF outputs; requires external match.
TX_OUT_U_HB2	A20	AO	Second of two UMTS high-band driver amplifier RF outputs; requires external match.
TX_OUT_G_LB	B23	AO	GSM low-band driver amplifier RF output; requires external match.
TX_OUT_G_HB	A22	AO	GSM high-band driver amplifier RF output; requires external match.
PWR_DET_IN	C23	AI	UMTS Tx power detector RF input; 50 Ω nominal.
PA_DAC	A18	AO	DAC output for GSM power amplifier envelope (PA_RAMP).

1 Pin type is AI = analog input or AO = analog output.

2 The expected RF matching network is shown in Figure 3-6.

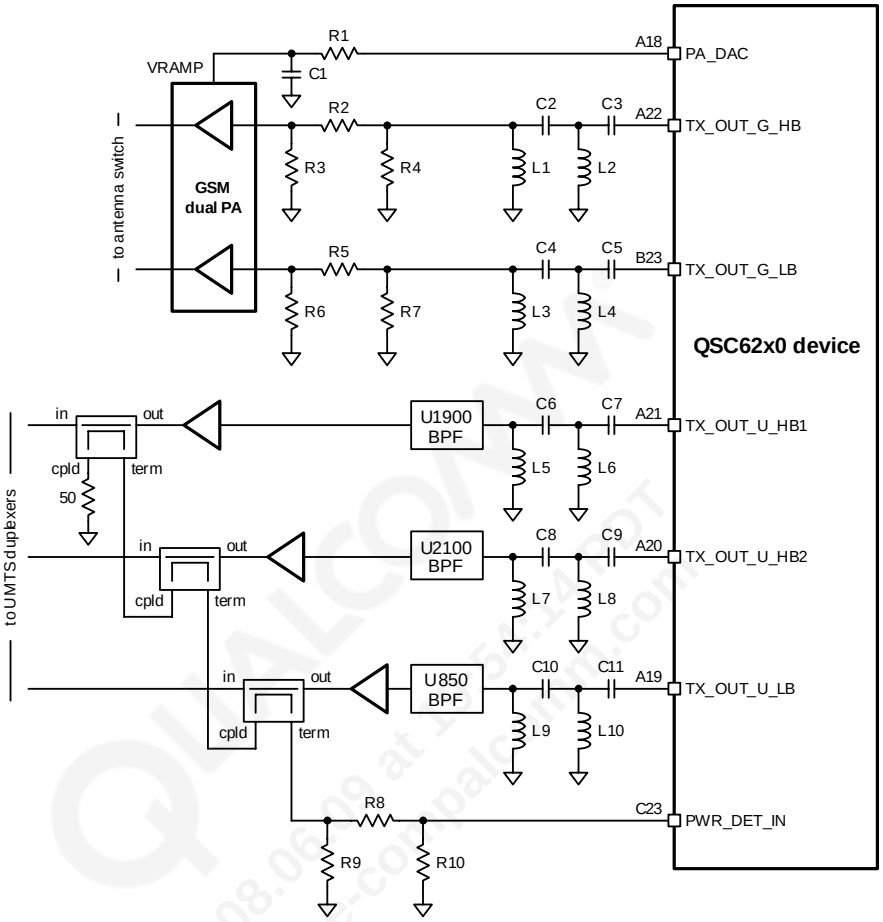


Figure 3-6 Expected UMTS and GSM RF Tx matching network topologies

Each of the five QSC Tx outputs are single-ended and nominally 50 Ω. They require matching networks for compatibility with the bandpass filters they are driving. The matching networks, beginning at the QSC pin, include: series capacitor, shunt inductor, series capacitor, and shunt inductor. Although some of the matching components ultimately might not be necessary, they should all be included in the initial layouts for matching flexibility and optimization.

In addition to the matching networks, the two GSM paths also require resistive attenuators to set the drive levels into the dual-PA module.

Each UMTS Tx output chain includes the bandpass filter, power amplifier, coupler, duplexer, and antenna switch. The GSM output chains include the dual-PA and the switch.

In addition to the main Tx signal paths, each mode is supported by a secondary QSC function:

- UMTS transmissions are supported by an integrated power detector. This QSC input (pin C23, PWR_DET_IN) operates over all of the UMTS transmit bands, presenting a wideband 50 Ω load. The three couplers are connected in a daisy-chain fashion, with a resistive attenuator setting the drive level into pin C23.
- The GSM transmit envelope is controlled by the PA_RAMP signal generated by the QSC device. This analog output is routed from pin A18 (PA_DAC) through an RC lowpass filter near the PA module.

3.4 Tx LO circuits

The QSC62x0 device integrates all of the frequency synthesizer functions that generate the UMTS and GSM transmit LO signals (UHF local oscillator, PLL circuits, and loop filter), plus the distribution circuits that deliver the quadrature LO signals to the two downconverters. The Tx LO signal can also be switched into the active GSM driver amplifiers in support of test and calibration.

The Tx LO synthesizer works in the same way as the Rx LO described in Section 3.2; see that section for a functional description.

3.4.1 Tx LO features

The QSC62x0 Tx LO features include:

- Optimization for tri-band UMTS plus quad-band GSM Tx operation
- Fully integrated frequency synthesizer
  - UHF Tx local oscillator
  - Feedback divider
  - Reference divider
  - Phase/frequency detector
  - Loop filter

- Synthesizer output is processed to drive the quadrature upconverters
  - Frequency plans ensure that f_{LO} is not equal to f_{RF} (an important consideration for ZIF reception)
  - Applied upconverter inputs are differential with quadrature phase relationships
- Synthesized LO signals are also routed to GSM driver amplifiers in support of test and calibration procedures

3.4.2 Tx LO connections

All Tx LO interfaces are internal to the QSC device.

4 Audio and Housekeeping ADC

Two QSC62x0 analog functions are discussed in this chapter: audio circuits and HKADC.

4.1 Audio

The QSC device supports forward-link (Rx) music/ringer melody applications and 8 kHz voice applications. Rx-path software selections include 13-bit or 16-bit linear conversion with sampling at 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, or 48 kHz. As shown in the audio functional block diagram (Figure 4-1), the QSC device supports one differential earpiece output, one differential or two single-ended line outputs, one differential speaker output (connected internally to the class-D speaker driver circuits), and one differential or two single-ended headset outputs.

In the audio transmit path, a 13-bit linear converter is provided with a software-selectable sampling rate of 8, 16, or 32 kHz. Two differential microphone inputs and one differential or two single-ended line inputs are supported, along with one-touch headset detection and microphone bias.

The integrated codec contains all required conversion and amplification stages needed by the audio front end. The receive and transmit filters meet ITU-T G.712 requirements. A programmable side-tone path is included for summing a portion of the Tx audio into the Rx path.

The codec is configured through QDSP4u8 commands and is not directly controlled by the microprocessor. The codec configuration command is sent to the QDSP4u8, which then executes the command to configure the codec. Data is exchanged between the codec interface and the QDSP4u8 DMA interface. The QDSP4u8 uses the Ex_DMA_4 channel for reading data from the codec and uses the Ex_DMA_5 channel for writing data to the codec.

The PCM interface allows an external codec to be used instead of the internal codec; this supports I2S modes that enable an external stereo DAC. See Section 9.7 for details.

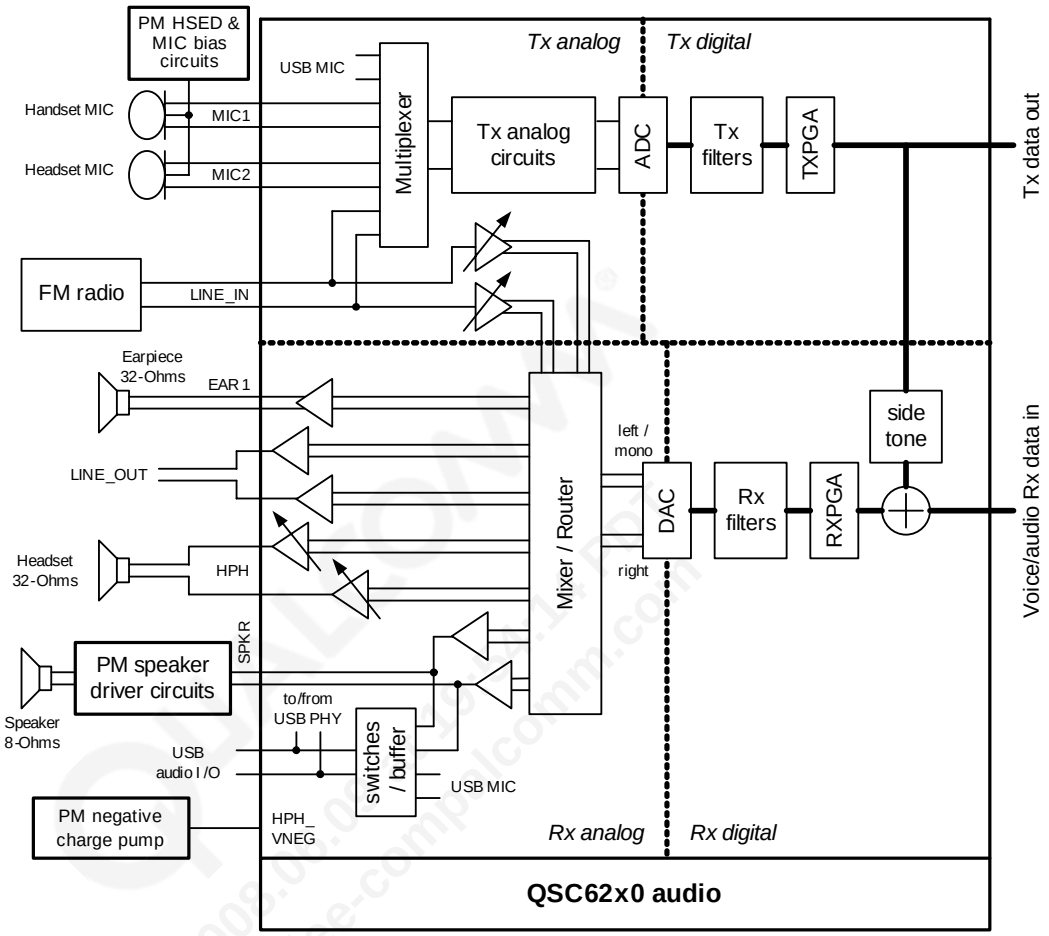


Figure 4-1 Audio functional block diagram

4.1.1 Audio connections

High-level QSC62x0 audio connections are illustrated in Figure 4-2.

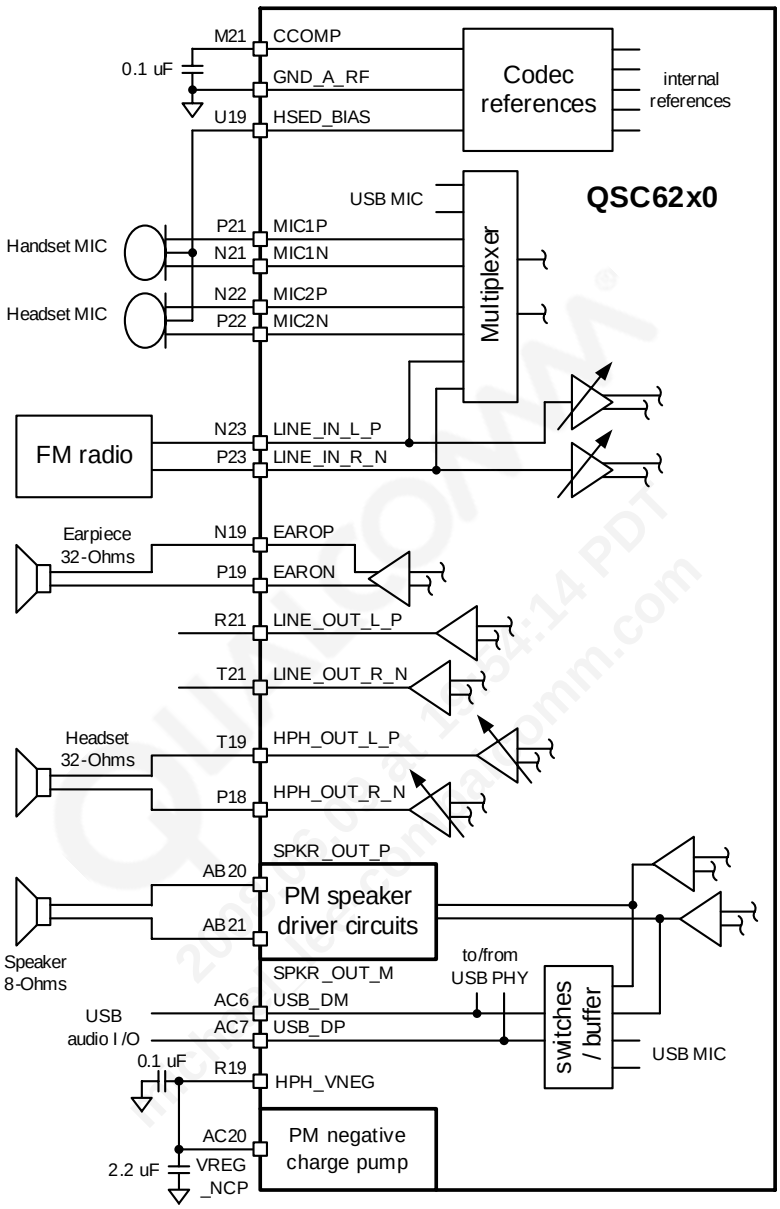


Figure 4-2 High-level audio connection diagram

All QSC62x0 dedicated audio connections are listed in Table 4-1. In addition:

- Some connectivity I/Os (such as USB) may be used to access audio functions; see Chapter 9 for details.
- The class-D speaker driver details are given in Section 16.2.3.
- The one-touch headset detection and associated microphone bias features are discussed in Section 16.2.4.
- A negative charge-pump (NCP) switched-mode power supply (SMPS) creates a -1.8 V supply that could be used as the negative supply to the headphone circuits; see Section 14.4.3.

Table 4-1 Audio connections

Pin #	Pin name	Pin type [1]	Functional description
Audio codec input connections (microphones)			
N21	MIC1N	AI	Microphone #1 input (-)
P21	MIC1P	AI	Microphone #1 input (+)
P22	MIC2N	AI	Microphone #2 input (-)
N22	MIC2P	AI	Microphone #2 input (+)
P23	LINE_IN_R_N	AI	Line input #1 (right side or negative)
N23	LINE_IN_L_P	AI	Line input #2 (left side or positive)
U19	HSED_BIAS	AI/AO	Headset send-end detect and microphone bias output
Audio codec output connections (speakers)			
P18	HPH_OUT_R_N	AO	Headphone output #1 (right side or negative)
T19	HPH_OUT_L_P	AO	Headphone output #2 (left side or positive)
R19	HPH_VNEG	AI	Negative bias voltage for headphone circuits; connect to ground or VREG_NCP (negative supply)
N19	EAROP	AO	Earphone amplifier output (+)
P19	EARON	AO	Earphone amplifier output (-)
R21	LINE_OUT_L_P	AO	Line output #1 (left side or positive)
T21	LINE_OUT_R_N	AO	Line output #2 (right side or negative)
M21	CCOMP	AO	External decoupling capacitor connection for the codec voltage reference. Connect a 0.1 µF capacitor between pin M21 and GND_A_RF.

[1] Pin type is AI = analog input or AO = analog output.

An on-chip voltage/current reference generates the precise voltages and currents required by audio functions. This reference circuit requires a single 0.1 µF bypass capacitor connected from the CCOMP pin (M21) to the analog/RF ground (GND_A_RF), then on to PCB GND.

The power management circuits provide a microphone bias voltage (via HSED_BIAS, pin U19) required for electric condenser microphones typically used in handset applications. This 1.8 V output provides as much as 1 mA of current. The same pin is used to implement one-touch headset detection as explained in Section 16.2.4.

The PM circuits also generate a -1.8 V supply for the headphone circuits. If the headphone circuits require a negative reference (rather than ground), connect VREG_NCP (pin AC20) to HPH_VNEG (pin R19). Negative charge-pump details are included in Section 14.4.3.

Additional audio functional and interface details are given in the following subsections.

4.1.2 Tx path inputs and gains

The Tx paths begin with one of two off-chip microphones or the line input; all three inputs support differential and single-ended configurations. The active input is selected by the codec configuration command (MIC_SEL). Only one of the three inputs is active at a time; the other (inactive) inputs are powered down. Interconnect circuit details are provided in Section 4.1.8 and its subsections; on-chip Tx input processing is shown in detail within Figure 4-1 (the blocks with Tx in their labels).

The desired audio Tx input is selected and routed to the gain stages; programmable for either 0 or +24 dB gain. The gain is controlled by software.

4.1.3 Rx path outputs

The Rx path terminates by driving one of three transducers or the line output. Transducer options include an earpiece, a loudspeaker (using the PM function's speaker driver), or a headset; differential configurations are supported in all cases, and the line output can also be configured as two single-ended signals. The active output is selected by the QDSP4u8 codec configuration command (AMP_SEL). Any or all of the four outputs can be active at any time; an inactive output is disabled and placed in a high-impedance state. Interconnect circuit details are provided in Section 4.1.8 and its subsections; on-chip Rx output processing is shown in detail within Figure 4-1 (the blocks with Rx in their labels).

4.1.4 Side-tone path

A side-tone path is available to add a portion of the Tx audio into the receive path; the path of the injected Tx signal includes a programmable gain stage with a range of 0 to -96 dB (controlled by the QDSP4u8 DMA parameter CODEC_ST_GAIN contained within the AFE_CMD_INT_CODEC_GAIN_CONFIG register). A programmed value of 0x4000 results in -12 dB of gain while 0x0000 mutes the side-tone. The gain calculation for CODEC_ST_GAIN is:

Gain = 20xLOG(CODEC_ST_GAIN/16384) - 12 dB

QUALCOMM recommends this side-tone technique to discourage users from talking too loudly into their phones.

4.1.5 Audio DSP

The audio DSP is discussed below, with separate sections for Tx and Rx processing.

4.1.5.1 Tx path audio DSP

Transmit data from the microphone is digitally filtered with an ITU G.712-compliant filter that attenuates input signals outside the 3400 Hz baseband and decimates the data rate to 8 kHz. There are two optional digital filters on the Tx path prior to the vocoder: a slope filter and a highpass filter. The slope filter (Figure 4-3) is designed to provide pre-emphasis for the high frequency audio prior to the vocoder.

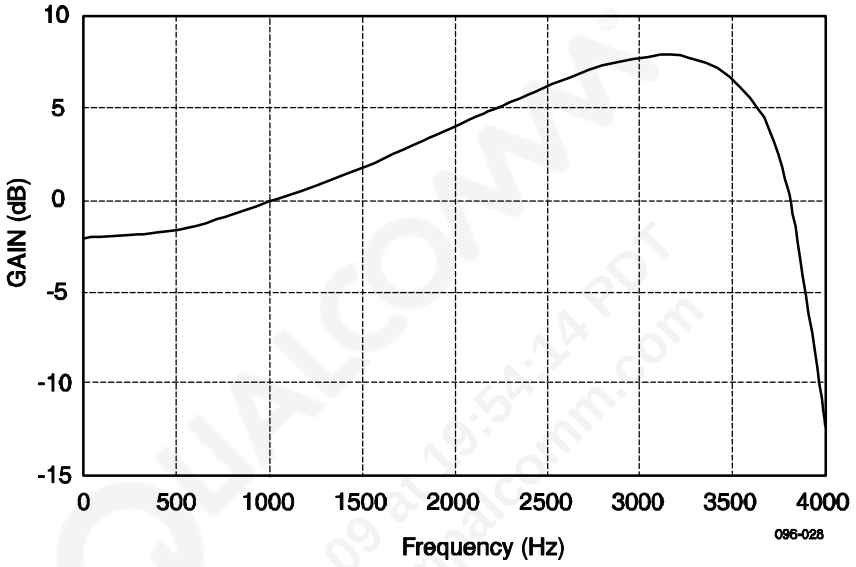


Figure 4-3 Tx slope filter response

The highpass filter (Figure 4-4) provides at least 30 dB of attenuation below 120 Hz. This QSC feature has more significance than in previous-generation products that included a highpass filter within an external gain circuit. It is recommended that the internal highpass filter be enabled for all applications where a microphone is the input source, thereby acting as a wind-noise rejection filter.

The two filters are individually enabled via codec configuration commands (TX_HPF_DIS_N and TX_SLOPE_FILT_DIS_N).

To supplement the other gain stages within the transmit path, the Tx DSP provides two more gain stages:

- The first stage has a range of -84 dB to +12 dB and is programmed using CODEC_TX_GAIN within the AFE_CMD_INT_CODEC_GAIN_CONFIG register.
- The second stage has a range of -84 dB to +12 dB and is programmed using TX_VOLUME within the VOLUME_CTRL_CMD register.

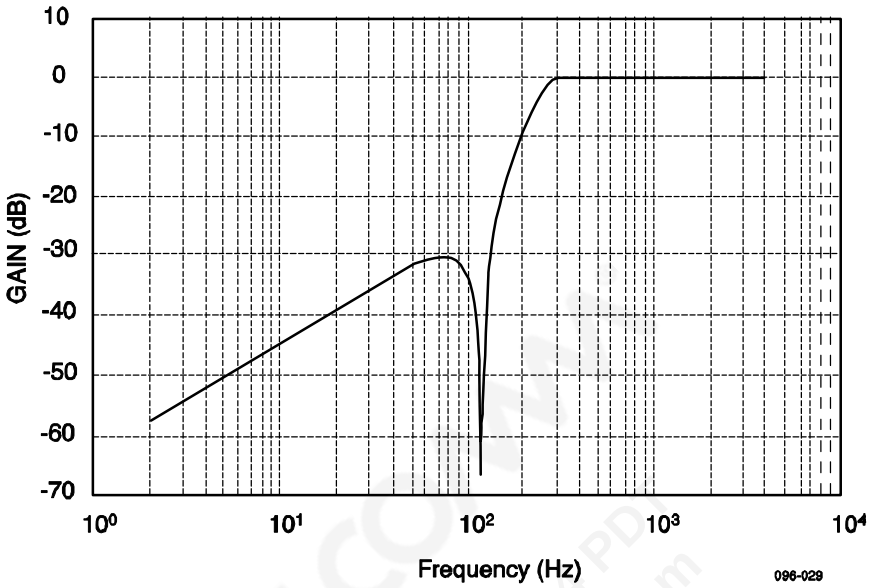


Figure 4-4 Highpass filter response

4.1.5.2 Rx path audio DSP

The receive DSP input is digitally filtered with an ITU G.712-compliant filter. The filter response is flat out to 3400 Hz but still provides at least 14 dB attenuation at 3.98 kHz to ensure adequate image rejection.

A user-selectable highpass filter is available for rejection of low-frequency noise. This filter provides at least 30 dB of attenuation below 120 Hz and is identical to the Tx highpass filter having the frequency response shown in Figure 4-4. The codec configuration command RX_HPF_DIS_N selects this highpass filter.

The Rx audio path includes two gain stages. The audio DSP gain is fixed for a particular application, though it is controllable via by the QDSP4u8 DMA parameter CODEC_RX_GAIN within the AFE_CMD_INT_CODEC_GAIN_CONFIG register. The CODEC_RX_GAIN is set to compensate for the sensitivity of the speaker driver.

4.1.6 Recommended gain settings

Tx and Rx path gain setting recommendations are defined in this section.

Tx path gain settings

The following four Tx path audio gains are recommended when using a microphone input for a voice call:

1. Set the MIC_AMP1 gain (0 or 24 dB) so that the output is 23 dB to 25 dB below the saturation point (2.828 Vpp) of the analog-to-digital converter. A typical phone design will set the MIC_AMP1 for 24 dB gain.
2. Set CODEC_TX_GAIN to approximately +4 dB.

3. Set the TX_VOLUME setting based on whether the TX AGC is turned on or not:
   a. If TX AGC is off, set TX_VOLUME to approximately 0 dB.
   b. If TX AGC is turned on with a -21 dB compression threshold, then set TX_VOLUME to approximately +3 dB. In this case there should also be 0 dB AGC static gain applied (assuming the previous recommendations are followed).
4. Implement gain adjustments using the CODEC_TX_GAIN gain. This should be set so that the mobile meets the send loudness rating (SLR) and distortion requirements of the 3GPP/3GPP2 standards and the carriers.

Rx path gain settings

The following two Rx path audio gains are recommended for a voice call in handset mode:

1. Set the RX_VOLUME setting based on whether the RX AGC is turned on or not:
   a. If RX AGC is off, set the maximum RX_VOLUME level to approximately 0 dB.
   b. If RX AGC is turned on with a -21 dB compression threshold, then set the RX_VOLUME maximum level to approximately +6 dB.
2. Generally, set the CODEC_RX_GAIN to any value within its usable range so that the receive loudness rating (RLR) and noise requirements of the 3GPP/3GPP2 standards and the carriers are met. **However, if values outside of the range of ±12 dB are needed, handset designers should contact QUALCOMM for assistance.**
3. When setting the headset gain, note that it is programmable; its range depends upon the external circuit's configuration:
   a. Legacy configurations (consistent with previous generation products that did not support the capless configuration) can be set for a nominal gain of -58.5 to -4.5 dB in 1.5 dB steps.
   b. Capless configurations can be set for a nominal gain of -58.5 to 0 dB in 1.5 dB steps.

4.1.7 Auxiliary PCM interface

The QSC62x0 auxiliary PCM interface enables communication with an external codec to support hands-free applications; linear, μ-law, and a-law codecs are supported. This PCM interface can also be used to drive an external stereo DAC (SDAC) to play stereo sound or music (MP3 or MIDI, for example). The I2S bus in output mode provides a serial link specifically for digital audio; it handles the transfer of audio data and transports it to the interface.

See Section 9.7 for further discussion.

4.1.8 External analog interface details

Three example audio interface configurations are shown in this section:

- Handset interface
- Headset interfaces (plus one-touch detection)
- Far-field speaker

4.1.8.1 Handset interfaces

Typical handset interfaces are shown in Figure 4-5. The earpiece output pins are connected directly to the handset earpiece, each with its own bypass capacitor. The capacitor value is selected to optimize performance in each design, but a value of 100 pF or less is expected (100 pF is used in the example). The output power for the differential ear output is typically 35 mW for a full-scale +3 dBm sine wave into a 32 Ω speaker.

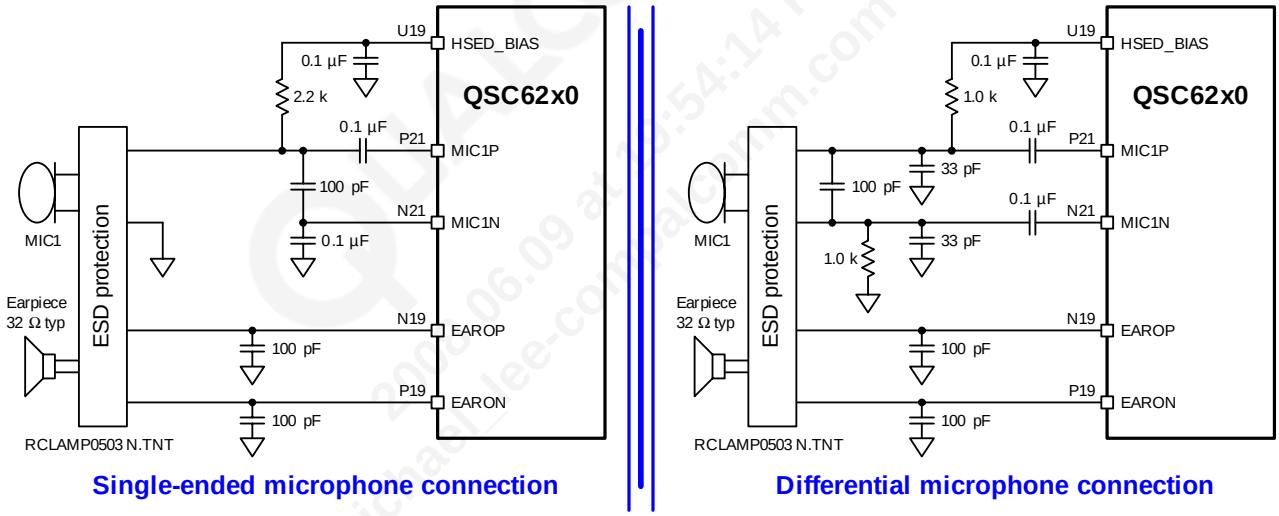


Figure 4-5 Typical handset microphone and earpiece interfaces

Two microphone configurations are shown in Figure 4-5:

- Single-ended (left side of figure) – The inactive negative microphone terminal is bypassed to ground by a 0.1 µF capacitor; the active positive terminal is AC-coupled to the microphone using a 0.1 µF capacitor. A differential 100 pF capacitor is attached across the two 0.1 µF MIC capacitors. Microphone biasing is applied using the HSED_BIAS source (pin U19, 1.8 V at 1 mA) and a 2.2 k resistor. A 0.1 µF bypass capacitor should be located near pin U19.
- Differential (right side of figure) – Each microphone pin includes a 1.0 k bias resistor and a 0.1 µF AC-coupling capacitor, with the capacitor located near the QSC input pin. The positive microphone terminal is connected to the HSED_BIAS pin (U19) through one of the 1.0 k resistors to bias the microphone. Again, a 0.1 µF bypass capacitor should be located near pin U19. In addition, each MIC connection includes a bypass capacitor (33 pF is used in the example) and a 100 pF capacitor is connected across the differential pair near the earpiece.

4.1.8.2 Headset interfaces

The most basic headset configuration is shown in Figure 4-6. This configuration uses a DC-coupled headphone interface and a standard single-ended microphone interface. Each headphone output pin (HPH_OUT_L_P, pin T19, and HPH_OUT_R_N, pin P18) includes a 33 pF bypass capacitor near the earpiece.

NOTE For FM radio applications, refer to the *QSC6240/QSC6270 QUALCOMM Single Chip Reference Schematic* (80-VF846-41) of that radio to ensure that the interface is correct.

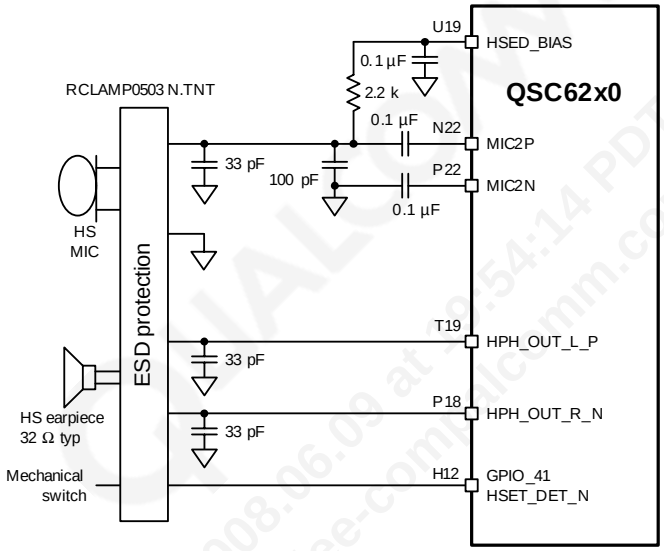


Figure 4-6 Typical headset interfaces (cap-coupled)

Although the microphone uses a single-ended configuration, both MIC2 inputs include a 0.1 µF AC-coupling capacitor. The positive input accepts the active signal and is AC-coupled directly to the microphone; the negative input is inactive and AC-coupled to ground. A 100 pF capacitor is connected across the two AC-coupling capacitors on the microphone side, followed by a 33 pF capacitor that bypasses the microphone output.

The positive microphone terminal is connected to the HSED_BIAS output pin through a 2.2 k resistor; again, this 1.8 V output provides 1 mA of bias current for the microphone. The HSED_BIAS pin is bypassed by a 0.1 µF capacitor (the same component mentioned earlier).

The output power for the single-ended HPH output is typically 10.8 mW for a full-scale +3 dBm sine wave into a 32 Ω speaker.

QSC62x0 also supports a capless headset interface as shown in Figure 4-7. While implementing this capless interface, the VREG_NCP output must be connected to the HPH_VNEG input with appropriate bypassing (as shown). In addition, as implied by its name, the 0.1 µF AC-coupling capacitors and 100 pF differential capacitor at the MIC inputs are omitted.

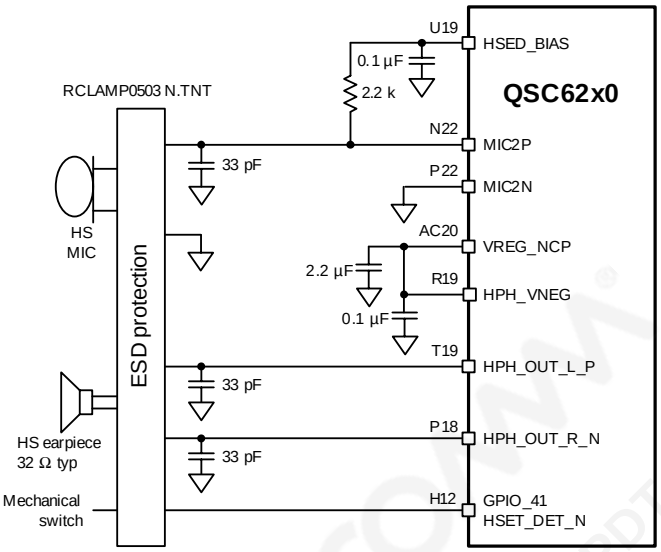


Figure 4-7 Typical headset interfaces (capless)

4.1.8.3 One-touch headset detection (headset send-end detect)

In addition to providing microphone bias voltage, HSED_BIAS can also be used for one-touch headset detection (or headset send/end detect). This detection technique is interrupt-driven and no polling by software is required. The baseband circuits can remain in their sleep mode, eliminating wake-up and GPIO scanning and saving DC power. Details are included in Section 16.2.4.

4.1.8.4 Far-field speaker amplifier (PM function)

The audio circuits described in this chapter can be supplemented by QSC speaker driver circuits to drive an external speaker. See Section 16.2.3 for details.

4.1.9 USB audio

The QSC62x0 device supports USB audio as described in Section 9.2.8.

4.2 HKADC

The HKADC (Figure 4-8) includes an analog multiplexer that selects an input for the sample and hold circuit. One of three inputs can be selected:

- HKAIN1, pin R22 – an external connection that is available as a general-purpose input, though it is often used to monitor the power amplifier(s) temperature.
- An on-chip connection to the power management circuit's analog multiplexer output. This allows monitoring of:
  - Key power supply nodes such as VBAT, VCHG, etc.
  - Multipurpose pins (when configured as analog inputs)
  - A few on-chip parameters such as the die temperature or V_{REF}

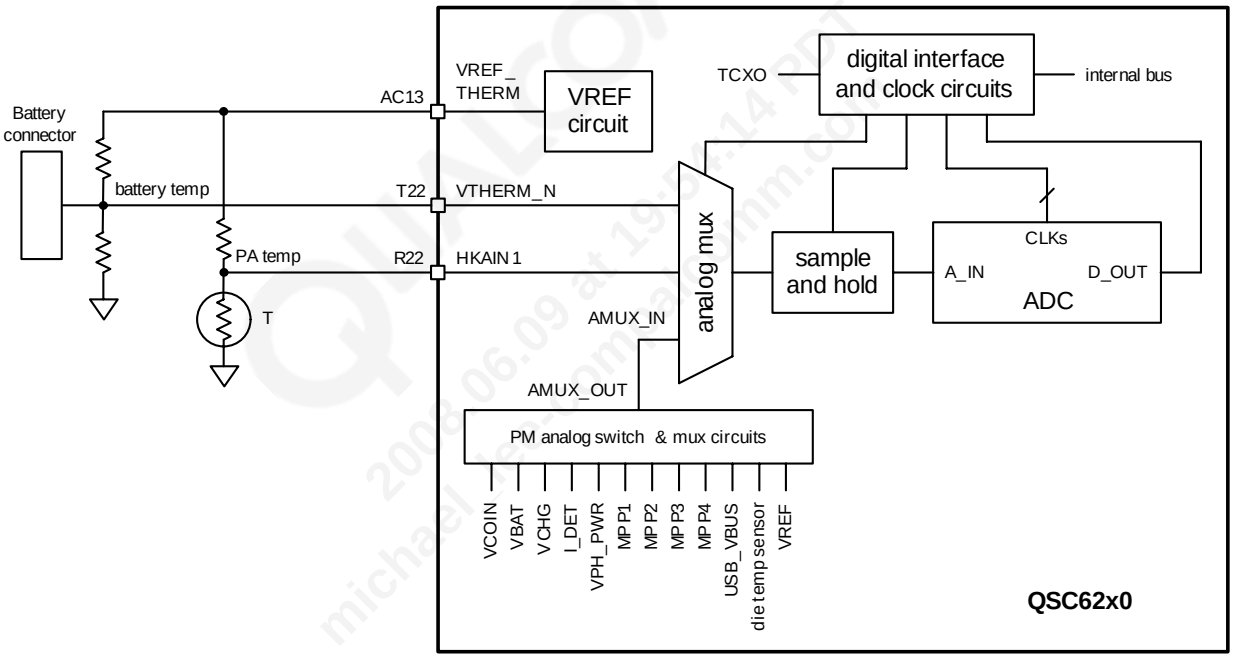


Figure 4-8 HKADC functional block diagram

The output of the 3:1 analog multiplexer is routed to the sample-and-hold circuit. The sampled analog voltage is then routed to the configurable successive-approximation ADC. The digitized output is routed to digital interface circuits that use the internal bus to communicate the requested values to software.

All the necessary timing and control signals are generated within the digital interface and clock circuits: analog multiplexer selection, sample-and-hold clock, ADC clocks, and interfacing with the internal bus. The ADC is sampled at 2.4 MHz.

4.2.1 HKADC connections

Two pins are assigned as analog inputs to the HKADC multiplexer circuit:

- T22, VTHERM_IN
- R22, HKAIN1

In addition to these two pins that are connected directly to the HKADC 3:1 multiplexer, the third input is internally connected to the power management circuit's analog multiplexer output that expands the number of analog signals available for HKADC conversion considerably. See Section 15.2 for further PM multiplexer details.

4.2.2 Analog input voltage range

The analog inputs applied to pins T22 and R22 must be limited to the range of 0 V to about 0.1 V below the programmed value of VREG_RFA (the regulated output voltage; its default value is 2.2 V).

4.2.3 HKADC operation (including conversion time)

Several HKADC parameters are configurable, including:

- The sampling interval $T_{sampling}$ is programmed using TSHK_DIG_CONFIG[7:6]. The programmed value results in a multiple of TCXO/8 clocks, either 3, 24, 36, or 48, with a default value of 00 (which corresponds to 3x[8/TCXO]). Because large resistances are typically connected to the HKADC pins, the recommended default value is 11 (which corresponds to 48x[8/TCXO]).
- The resolution N is programmed using TSHK_DIG_CONFIG[3:2] to a value of 8, 10, or 12, with a default value of 12 bits.
- Another parameter — the ADC clock rate — is programmable, but should not be changed from its default value of 2.4 MHz.

These variables set the conversion time (and therefore the throughput rate) according to the following equation:

$$T_{total} = T_{sampling} + T_{acquisition} + N \times T_{ADC\_clock}$$

For example, using the default values:

$$T_{total} = 24 \times (8/19.2 \times 10^6) + 1/2.4 \times 10^6 + 12 \times (1/2.4 \times 10^6) = 15.4\ \mu sec$$

This same example — using the default values — is used to create an example timing diagram (Figure 4-9) that illustrates the conversion process.

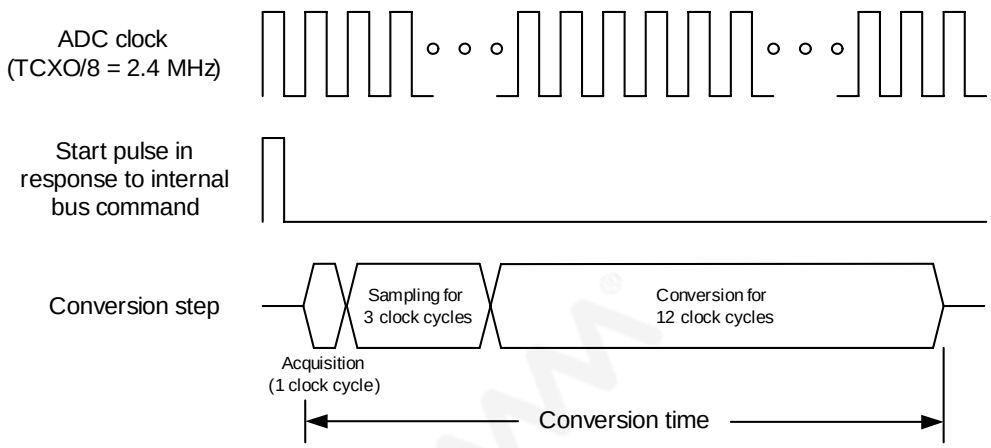


Figure 4-9 HKADC conversion timing diagram

4.2.4 HKADC analog interface considerations

An equivalent model for the interface from an external circuit to the HKADC input is shown in Figure 4-10. The input multiplexer selects an input pin by closing the corresponding analog switch. When a conversion is initiated, the analog multiplexer and sample (SAMPL) switches both close until the end of the sample interval (three HKADC clock cycles). After sampling, both switches open and the voltage to be converted is held on the sampling capacitor, C_s.

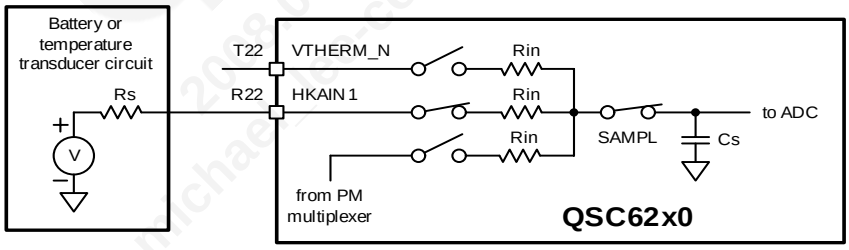


Figure 4-10 Equivalent circuits for HKADC inputs and external voltage sources

During the sampling interval, the external circuit connected to the selected HKAIN pin must supply enough current to charge the C_s through the on-resistance (R_{in}) of the input multiplexer (5 k maximum). For accurate conversion results, the voltage on the C_s must settle to within 0.01% of its final value (~1/2 LSB) within the sampling interval.

This discussion leads to a circuit and timing relationship that must be met:

$$7 \times (R_s + R_{in}) \times Cs < (48/CLK\_IN)$$

Given that CLK_IN = TCXO/8 = 2.4 MHz, this equation simplifies to:

$$7 \times (R_s + R_{in}) \times Cs < 20\ \mu sec$$

Knowing R_{in} (5 k maximum) and C_s (approximately 60 to 72 pF), the maximum source resistance of the external circuit is calculated to be 42.6 k ($20x10^{-6}/(60x10^{-12} \times 7)$ - 5 k).

The relationships given above allow the maximum source resistance to be calculated for various circuit conditions.

Multiplexer input pins that are not selected present very high impedance to their source circuits, essentially providing no load to the external circuits at the HKAIN pins.

4.2.5 Example HKADC application: temperature sensor

The HKADC can be used to monitor a temperature sensor by following this procedure:

1. Set the sampling interval to 48 x (8/TCXO) using TSHK_DIG_CONFIG[7:6] = 11.
2. Calibrate the temperature sensor in software:
   a. Route V_{bg} from the VTHERM_IN input to the HKADC by setting GPR_B1[7] = 1 and GPR_B2[0] = 1.
   b. Convert V_{bg} to its digital value $D_{bg\_measured}$ using the HKADC.
3. Correct the measured temperature:
   a. $D_{temp\_correct} = D_{temp\_measured} + D_{delta}$
   b. $D_{delta} = D_{bg\_ideal} - D_{bg\_measured}$
   c. $D_{gb\_ideal} = (1.189/2.6) \times 1024 = 468$

5 Baseband Processors

The QSC62x0 device integrates multiple processors on-chip: one ARM microprocessor and two DSP processors. Each processor is part of a functional subsystem:

- The **micro subsystem** includes the ARM926EJ-S microprocessor.
- The **modem subsystem** includes the QDSP4u8 digital signal processor (mDSP).
- The **application subsystem** includes the QDSP4u8 application digital signal processor (aDSP).

All of these subsystems, along with the advanced high-performance bus (AHB), are discussed in this chapter.

Figure 5-1 shows a high-level functional block diagram of the processors and bus architecture. Master capability (initiating reads or writes) is shown as an arrow from a block to a bus; slaves are shown by an arrow from a bus to a block.

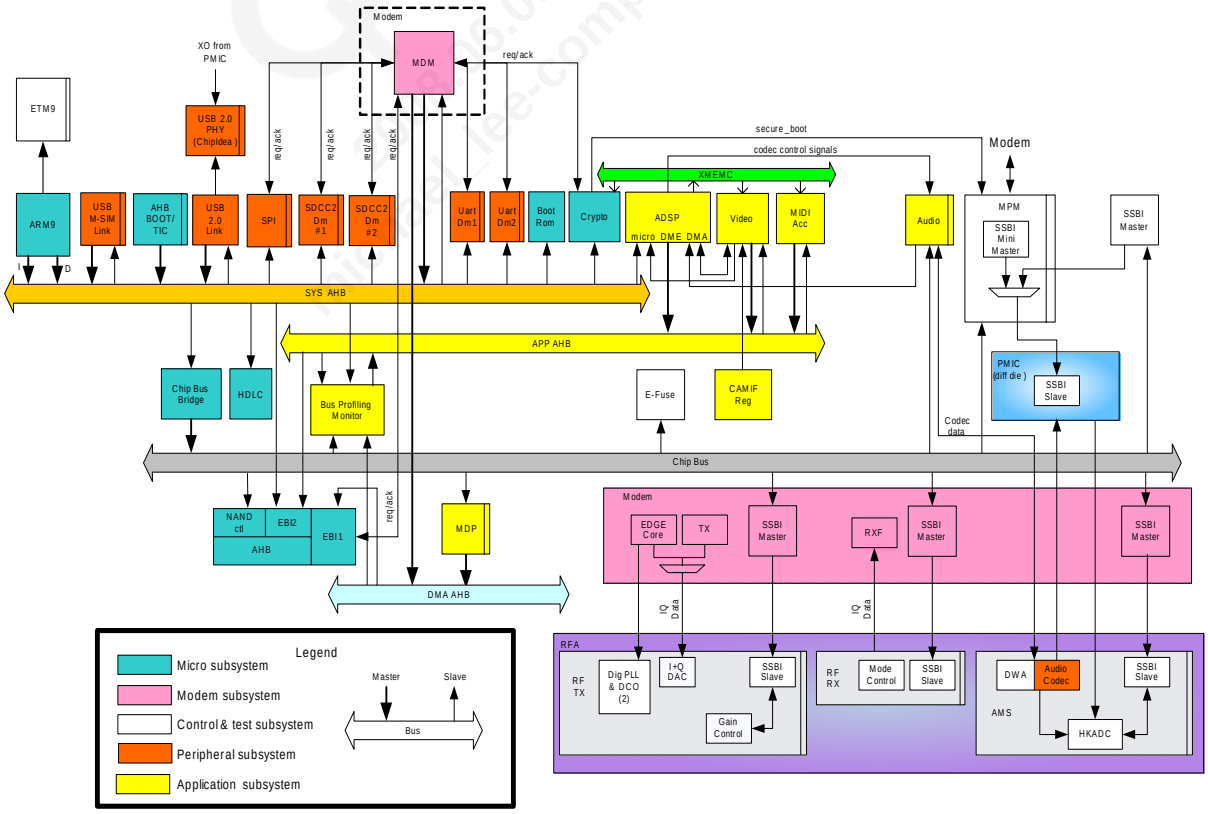


Figure 5-1 Processors and bus architecture

5.1 Micro subsystem

The QSC62x0 device supports several types of voice, video, and data features, requiring significant processing power and memory-access bandwidth. The embedded ARM926EJ-S microprocessor, advanced high-performance bus, and supporting peripherals meet these needs with a targeted microprocessor speed of 184 MHz.

This architecture provides key improvements over previous-generation products:

- EGPS accelerators and GACC and 1024x searchers are connected to the SYS_AHB and DMA_AHB using the MDM™ for external memory access.
- A high-speed UART (UART_DM) is connected to the SYS_AHB, with handshake sideband signals taking advantage of DMA transfers mastered by MDM.
- UART2 is connected to SYS_AHB for USIM only.
- An MDM master interface to the DMA_AHB bus, facilitated by two MDM client interfaces, supports two bus connections. This allows the low-latency data required by the EGPS accelerators to gain high priority during external memory access arbitration.
- All buses include grant-table programmability-priority arbiters to allow more flexibility in priority and bandwidth assignments.

5.1.1 Micro subsystem features

- ARM926EJ-S integration
- ARM926EJ-S operating up to a target frequency of 184 MHz
- Multilayer AHB system operating up to a maximum frequency of 92 MHz
- Two external memory interfaces with arbitration for the multilayer AHB system and memory controllers
- Support for external memory and peripherals access for the QDSP4u8 processors and the graphics processor
- Support for SDRAM
- Boot-up from NAND flash device
- Debugging capabilities at target speeds

5.2 Advanced high-performance bus system

As shown in Figure 5-1, the ARM and peripherals architecture contains three AHB buses, a chip bus, and a chip bus bridge. The three AHB buses are:

- System AHB (SYS_AHB): dedicated to the ARM9™ for functional operation. Hardware register configuration is only accessible from this bus.
- Application AHB (APP_AHB): handles multimedia data traffic to and from EBI1 and is unaffected by EBI2 accesses.

- Direct memory access AHB (DMA_AHB): handles DMA data transfers controlled by the ARM. This is primarily used to transfer screen updates from EBI1 main memory to an LCD. It is also used for low-bandwidth I/O devices (SD card, USB, and MDDI).

Further discussion of the three buses is presented in the following paragraphs.

SYS_AHB

The SYS_AHB uses the grant-table programmable-priority arbiter to set the following recommended priorities:

1. TIC (used in test mode)
2. D AHB
3. I AHB
4. MDM

Every master is programmed with a unique priority index. Priority assignments are unique, and only one master is assigned to each priority level.

Other SYS_AHB details:

- I AHB is the default master.
- SPLIT and RETRY responses are not supported.
- Early bus termination is not supported.
- A programmable arbiter is used rather than a hardwired arbiter (with defaults shown above). The priorities of different masters can be reconfigured using a four-word STM to the AHB_SYS_BUS_PRIORITY register.

APP_AHB and DMA_AHB

An encoded grant-table programmable-priority scheme arbitrates the APP_AHB and DMA_AHB bus masters; the highest priority master requesting the bus is granted access. Further details include:

- Priority levels depend upon the number of masters:
  - There are n + 1 priority levels for n masters.
  - The 0-priority gets no access.
- Every master is programmed with a unique priority index; priority assignments are unique and only one master is assigned to each priority level.
- Every master also has a unique master ID, used to indicate the default for the master bus as follows:
  - APP_AHB
    - 0b000: SW master
    - 0b001: Video
    - 0b010: ADSP DME

- DMA_AHB
  - 0b000: SW master
  - 0b001: MDM
  - 0b010: USB_TDI
  - 0b011: RLP_DMA
  - 0b101: MDP
  - 0b110: SDCC
- Control registers
  - APP_AHB uses the AHB_APP_BUS_PRIORITY register
  - DMA_AHB uses the AHB_DMA_BUS_PRIORITY register

- Arbitration control includes an **SW master** that allows software to stop activity on the DMA_AHB and APP_AHB buses. When granted, the SW master places the bus in its idle state and remains in control as long as it keeps requesting the bus. The SW control register is AHB_BUS_REQUEST.
- The current granted master is monitored through the AHB_BUS_MASTERS register.
- SPLIT and RETRY responses are not supported.
- Early bus termination is not supported.

Arbitration

There are two levels of arbitration when accessing external devices. A master must first win arbitration to access its AHB bus and then win arbitration in the multiported memory controller for EBI1 or EBI2.

5.3 ARM926EJ-S microprocessor

The ARM926EJ-S is a single-clock, high-performance, Java-enabled, synthesizable core. It includes a five-stage pipelined RISC architecture, both 32-bit ARM and 16-bit Thumb instruction sets, a 32-bit address bus, and a 32-bit internal data bus. For more information on using the ARM926EJ-S, refer to the Advanced RISC Machines publication *ARM926EJ-S Data Sheet* (document number ARM DDI 0029E).

The ARM926EJ-S provides increased processing power by using a Harvard cache architecture; its processor subsystem includes an ARM9EJ-S integer core, separate instruction and data SYS_AHB bus interfaces, and separate instructions and data caches. Not all the features of the ARM926EJ-S are used in the QSC62x0 device.

5.3.1 Microprocessor features

- Harvard caches
- 32 kB instruction and 32 kB data caches, each four-way associative

- Memory management unit (MMU)
- Harvard AHB interface
- ARM and Thumb modes supported
- Jazelle ARM9EJ-S core (Java support)
- Embedded trace macrocell (ETM) support
- Low-power features

5.3.2 Burst access

The ARM926EJ-S supports burst transfers to its slaves, specifically for cache operations. Burst transfers coupled with the use of page mode, burst mode, and SDRAM memories help increase system performance. Table 5-1 shows the burst-transfer types supported.

Table 5-1 ARM burst-transfer support

HBURST[2:0]	Description	Operation
SINGLE	Single transfer of a word, half-word, or byte	Single transfer of word NC instruction fetches Page-table walk read Continuation of a burst that either lost a grant or received a split/retry response
INCR4	Four-word incrementing burst	Half-line cache write-back Instruction pre-fetch (if enabled) Four-word burst NCNB, NCB, WT, WB write
INCR8	Eight-word incrementing burst	Full-line cache write-back Eight-word burst NCNB, NCB, WT, WB write
WRAP8	Eight-word wrapping burst	Cache line-fill

5.4 Memory map and memory map decoder

The memory map (Table 5-2) indicates how the different slaves are assigned memory address ranges on the system. The memory map decoder performs a centralized address decoding function to generate select lines to each of the system bus slaves, indicating that a read or write access to the slave is required.

The QSC62x0 device supports **only** trusted boot. The QSC device boots from its internal boot ROM located at 0xFFFF0000. More boot details are presented in Section 11.3.

Table 5-2 QSC62x0 memory map (NAND BOOT; MODE = 1)

Region	Base address	Bytes
EBI1_CS0	0x00000000	256 M
EBI1_CS1	0x10000000	256 M
UART2_DM	0x28000000	128 M

Table 5-2 QSC62x0 memory map (NAND BOOT; MODE = 1) (continued)

Region	Base address	Bytes
EBI2_LCD	0x30000000	128 M
EBI2_CS0	0x38000000	128 M
EBI2_CS1	0x40000000	128 M
EBI2_CS2	0x48000000	128 M
EBI2_CS3	0x50000000	128 M
NAND	0x60000000	128 M
ADSP	0x70000000	128 M
IMEM	0x78000000	128 M
QSC	0x80000000	
GPIO2	0x84000000	64 M
UART	0x88000000	128 M
USB	0x90000000	128 M
USB_2ND	0x68000000	128 M
HDLC	0x98000000	128 M
CRYPTO2	0xA0000000	128 M
MIDI	0xA8000000	128 M
MODEM	0xB0000000	128 M
SDCC	0x20000000	128 M
SDCC_2ND	0xB8000000	128 M
SPI	0xC0000000	256 M
RESERVED_3	0xD0000000	
BOOT_ROM	0xFFFF0000	64 k

5.5 Trusted boot (required mode)

Nontrusted boot is not supported by the QSC62x0 device; only trusted boot is supported. Boot topics are addressed within the security section of this guide (Section 11.3).

5.6 Reset and pause

The reset/pause block is responsible for:

- Generating a synchronous reset (Figure 5-2) for the ARM and the other advanced microcontroller bus architecture (AMBA) peripherals
- Providing status information about the source of the last reset (pin or watchdog generated)
- Providing a software-controllable mechanism for stalling the ARM for a specified amount of time

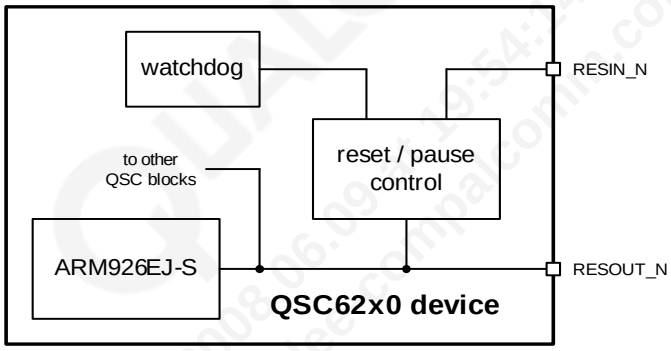


Figure 5-2 Reset generation

This block generates reset signals for the ARM and the system. It also controls the clock to the ARM microprocessor, thus providing the ability to reset or pause the ARM. Software can use the pause mechanism to halt the microprocessor for a specified amount of time.

Two reset signals are combined: power-on reset (RESIN_N) and watchdog reset, and then a synchronously deasserting reset pulse is sent to the rest of the chip. HRESET_N is guaranteed to be asserted for a minimum of two CLK pulses, regardless of the duration of the RESIN_N pulse. The source of the last reset is stored in bit 0 of register RESET_STATUS.

The ARM can be paused by deasserting HREADY, which stalls the microprocessor until HREADY asserts.

Software pauses the processor by writing to a 16-bit counter through the register PAUSE_TIMER. This action immediately deasserts HREADY for four cycles. ARM is paused for 4 + PAUSE_TIMER periods. For example, if TCXO is a 19.2 MHz clock (52 ns period), then writing the value 10 to PAUSE_TIMER pauses the microprocessor for 52 ns x 4 + 10 x 52 ns = 728 µs.

This timer is meant to introduce small delays in the system software. If the microprocessor clock frequency is sufficiently low, delays introduced by this timer can exceed the watchdog timer duration, resulting in a watchdog reset of the system.

5.7 Watchdog timer

The watchdog timer (Figure 5-3) is a 21-bit counter running on the sleep controller clock regime that enables the mobile station to recover from unexpected hardware or software anomalies. Unless the microprocessor periodically resets the watchdog timer, the watchdog timer expiration resets the mobile station.

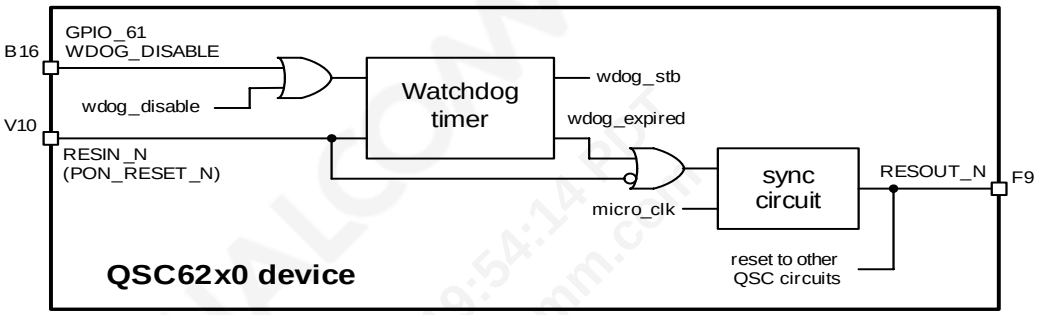


Figure 5-3 Watchdog timer configuration

The watchdog timer is disabled and reset by asserting the WDOG_DISABLE pin (B16). Asserting the RESIN_N pin also resets the watchdog timer. The watchdog timer is enabled by leaving the WDOG_DISABLE pin unconnected (it has an internal pull-down) or by connecting it to ground. The watchdog timer is disabled by hardware as soon as the ARM processor enters debug mode (indicated by the DBGACK pin of the processor transitioning from low to high). To re-enable the watchdog timer, apply a system reset to clear the state of the internal signal WDOG_DISABLE.

The WDOG_DISABLE pin (B16) is sampled coming out of reset and held until the next reset. When enabled, the watchdog circuit pulses the signal WDOG_EXPIRED when the watchdog timer has expired. In native mode, this signal is combined with the RESIN_N pin to generate RESOUT_N and the internal reset for the QSC device.

5.7.1 Sleep mode

In sleep mode, the microprocessor cannot reset the watchdog timer. The sleep controller contains an auto-kicker that resets the watchdog every 32 sleep clock cycles. The auto-kicker is enabled by the microprocessor by writing a particular bit sequence to the SLEEP_CTL: AUTO_KICK_ARM bit. When the sleep counter reaches its terminal count, the auto-kicker circuit is disabled,

When the sleep crystal is not being used (SLEEP_XTAL_EN = 0), the watchdog timer is disabled.

5.7.2 Non-sleep mode

In non-sleep mode, the microprocessor must reset the watchdog timer at least once every 213 ms (assuming a 32.768 kHz sleep clock). If not, the watchdog timer expires and asserts an internal RESIN_N signal to the system. The WDOG_EXPIRED signal lasts about 100 ms (assuming a 32.768 kHz sleep clock). To reset the watchdog timer, the microprocessor must write a particular sequence to the WDOG_RESET:WATCH_DOG bit.

5.7.3 General-purpose timer operation

When the microprocessor is powered down after programming a general-purpose timer delay that exceeds the watchdog timer interval, the general-purpose timer auto-kicker must be enabled to prevent the watchdog from resetting the ARM microprocessor. The auto-kicker is disabled when the general-purpose timer reaches its terminal count.

5.8 mDSP

The QSC62x0 devices have an integrated QDSP4u8 core that is dedicated for modem functions and WCDMA/GSM processing.

5.9 aDSP

QSC62x0 devices have a second integrated QDSP4u8 core that is dedicated for application subsystem functions such as all vocoder functions, audio applications, CMX, and Qcamera.

6 Memory Support - EBI1 and EBI2

The QSC62x0 device has two external bus interface (EBI) ports: EBI1 and EBI2.

EBI1 supports high-speed synchronous dynamic devices. Its memory controller supports the new mobile DDR SDRAM memories with its higher bandwidth and ability to run at high clock frequencies. This interface supports the high-bandwidth, high-density, and low-latency requirements of the QSC's advanced on-chip capabilities such as the ARM9 processor, high-performance graphics, and video applications.

EBI2 is the slower speed interface intended to support memory devices such as NAND flash and asynchronous SRAM, peripheral devices such as LCDs, and the UBM receiver for multicast or broadcast reception (QSC6270 only). In addition, EBI2 is required to support a synchronous-burst AAD NOR flash to enable a NOR/DDR SDRAM memory configuration because the simultaneous mode (NOR, SDRAM) is not supported on the EBI1 bus.

The ARM926EJ-S microprocessor is a cached processor and all its accesses to external memory use burst techniques of four or eight 32-bit words when the memory region is declared to be cacheable/bufferable. To take advantage of this QSC higher performance feature, data from memories must satisfy the requirements for these burst accesses.

The QSC62x0 supports the memory configurations listed in Table 6-1.

Table 6-1 Memory configurations

CFG	EBI1 memory	EBI2 memory	Comments
1	64 MB DDR SDRAM	64 MB NAND	
2	32 MB DDR SDRAM	64 MB AAD NOR	
3	32 MB DDR SDRAM	32 MB NAND	
4	16 MB DDR SDRAM	32 MB AAD NOR	Low-tier configuration

Sections dedicated to the details of each EBI type are presented below.

6.1 EBI1

EBI1 is a high-performance external memory interface for the QSC62x0 digital block that supports DDR SDRAM devices.

NOTE Slower-speed devices such as NAND and LCD must **not** be connected to EBI1 but instead to EBI2.

Specifically, the following memory devices are supported on EBI1:

- SDRAM
  - 16-bit low-power DDR SDRAM
  - Minimum size per chip-select: 16 MB (128 Mbit)
  - Maximum size per chip-select: 128 MB (1 Gbit, 1 k columns only)

Table 6-2 SDRAM bank/row/column configuration for monolithic devices

Density (Mbits)	Arrangement	Banks	Rows	Columns	Comments
128	8Mx16	4	4 K	512	JEDEC standard
256	16Mx16	4	8 K	512	JEDEC standard
512	32Mx16	4	8 K	1 K	JEDEC standard
512	32Mx16	4	16 K	512	Reduced page size results in lower power consumption Requires address pin A[13]
1024	64Mx16	4	16 K	1 K	JEDEC standard Requires address pin A[13]

Characteristics of the EBI1 clock are listed below:

- Maximum clock rate is 92 MHz, defined by the AMSS software.
- The EBI1 memory controller clock is synchronous to the bus clock (HCLK).

6.1.1 EBI1 features

6.1.1.1 General EBI1 memory controller features

The memory controller offers the following features:

- Low-power, 1.8 V I/O memories only
- Two chip-selects that can support 16-bit DDR SDRAM memories
- A buffering scheme for all write accesses
- An SDRAM optimizer to optimize accesses to the DDR SDRAM
- Clock rates from 20 MHz to 92 MHz (max)
- One clock-enable pin for each chip-select to allow one DDR SDRAM to operate in its self-refresh mode while the second is active

6.1.1.2 EBI1 DDR memory-controller features

This block forms the core controller for 16-bit DDR memory devices. The following features and memory devices are supported on this interface:

- 16-bit DDR (JEDEC LPDDR compatible) memory devices only
- Four-bank DDR SDRAM devices only
- 14 row address pins
  - Supported row pin selection: 12, 13, 14
- 10 column address pins
  - Supported column pin selection: 9, 10
- Maximum device density: 1 Gbit per chip-select
- Two chip-selects
- Supported burst lengths: 4, 8
- Precharge modes:
  - Manual precharge and auto precharge features are supported.
  - Concurrent precharge (software configurable) is supported.
- Auto-refresh and self-refresh commands supported
- Powerdown and deep powerdown features supported
- Linear and interleave (bank address and row address bits are interleaved) memory modes supported
- DDR SDRAM optimization

  The SDRAM optimizer regroups/reorders SDRAM commands to maximize the efficiency and bandwidth. This enables the SDRAM controller to pipeline many of the operations (for example, bank activate and precharge) and reduces the average system access latency and improves use of the external memory. See Section 6.3.2 for a description of SDRAM optimization.

6.1.2 EBI1 connections

All EBI1 connections are listed in Table 6-3.

Table 6-3 EBI1 connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
EBI1_A_D_31	K6	P1 (1.8 V)	BS-K	(5.6)	SDRAM_BA[1] (bank address)
EBI1_A_D_30	M6	P1 (1.8 V)	BS-K	(5.6)	SDRAM_BA[0] (bank address)
EBI1_A_D_29	M5	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[13]
EBI1_A_D_28	N5	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[12]
EBI1_A_D_27	K3	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[11]
EBI1_A_D_26	M3	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[10]
EBI1_A_D_25	L6	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[9]
EBI1_A_D_24	H3	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[8]
EBI1_A_D_23	N6	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[7]
EBI1_A_D_22	R5	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[6]
EBI1_A_D_21	P6	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[5]
EBI1_A_D_20	T5	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[4]
EBI1_A_D_19	P3	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[3]
EBI1_A_D_18	T6	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[2]
EBI1_A_D_17	T3	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[1]
EBI1_A_D_16	U3	P1 (1.8 V)	BS-K	(5.6)	SDRAM_A[0]
EBI1_A_D_15	G2	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[15]
EBI1_A_D_14	G1	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[14]
EBI1_A_D_13	H1	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[13]
EBI1_A_D_12	H2	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[12]
EBI1_A_D_11	K1	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[11]
EBI1_A_D_10	K2	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[10]
EBI1_A_D_9	L1	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[9]
EBI1_A_D_8	L2	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[8]
EBI1_A_D_7	M1	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[7]
EBI1_A_D_6	M2	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[6]
EBI1_A_D_5	N1	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[5]
EBI1_A_D_4	N2	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[4]
EBI1_A_D_3	R1	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[3]
EBI1_A_D_2	R2	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[2]
EBI1_A_D_1	T1	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[1]
EBI1_A_D_0	T2	P1 (1.8 V)	BS-K	(5.6)	SDRAM_D[0]

Table 6-3 EBI1 connections (continued)

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
EBI1_ADV_N	J6	P1 (1.8 V)	DO	(5.6)	SDRAM_RAS_N
EBI1_M_CLK	K5	P1 (1.8 V)	DO	(5.6)	SDRAM DDR clock; differential.
EBI1_M_CLK_N	L5	P1 (1.8 V)	DO	(5.6)	
EBI1_CKE_0	F3	P1 (1.8 V)	DO	(5.6)	SDRAM_CLK_EN[0]
EBI1_CKE_1	P5	P1 (1.8 V)	DO	(5.6)	SDRAM_CLK_EN[1]
EBI1_WE_N	H5	P1 (1.8 V)	DO	(5.6)	SDRAM_WE_N
EBI1_OE_N	G5	P1 (1.8 V)	DO	(5.6)	SDRAM_CAS_N
EBI1_CS1_N	J5	P1 (1.8 V)	DO	(5.6)	SDRAM_CS_N[1]
EBI1_CS0_N	H6	P1 (1.8 V)	DO	(5.6)	SDRAM_CS_N[0]
EBI1_DQM_0	P2	P1 (1.8 V)	DO	(5.6)	SDRAM_DQM[0]
EBI1_DQM_1	J2	P1 (1.8 V)	DO	(5.6)	SDRAM_DQM[1]
EBI1_DQS_0	P1	P1 (1.8 V)	BS-PU	(5.6)	SDRAM_DQS[0]
EBI1_DQS_1	J1	P1 (1.8 V)	BS-PU	(5.6)	SDRAM_DQS[1]

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs to allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

6.1.3 EBI1 memory configurations

EBI1 memory connectivity issues for various memory types are presented. In addition, note that the burst interface is always a MUXed interface from the QSC62x0 standpoint.

6.1.3.1 DDR SDRAM memory connectivity

EBI1 is used to support DDR SDRAM memories. Figure 6-1 shows the connections required to support a single 16-bit DDR SDRAM device.

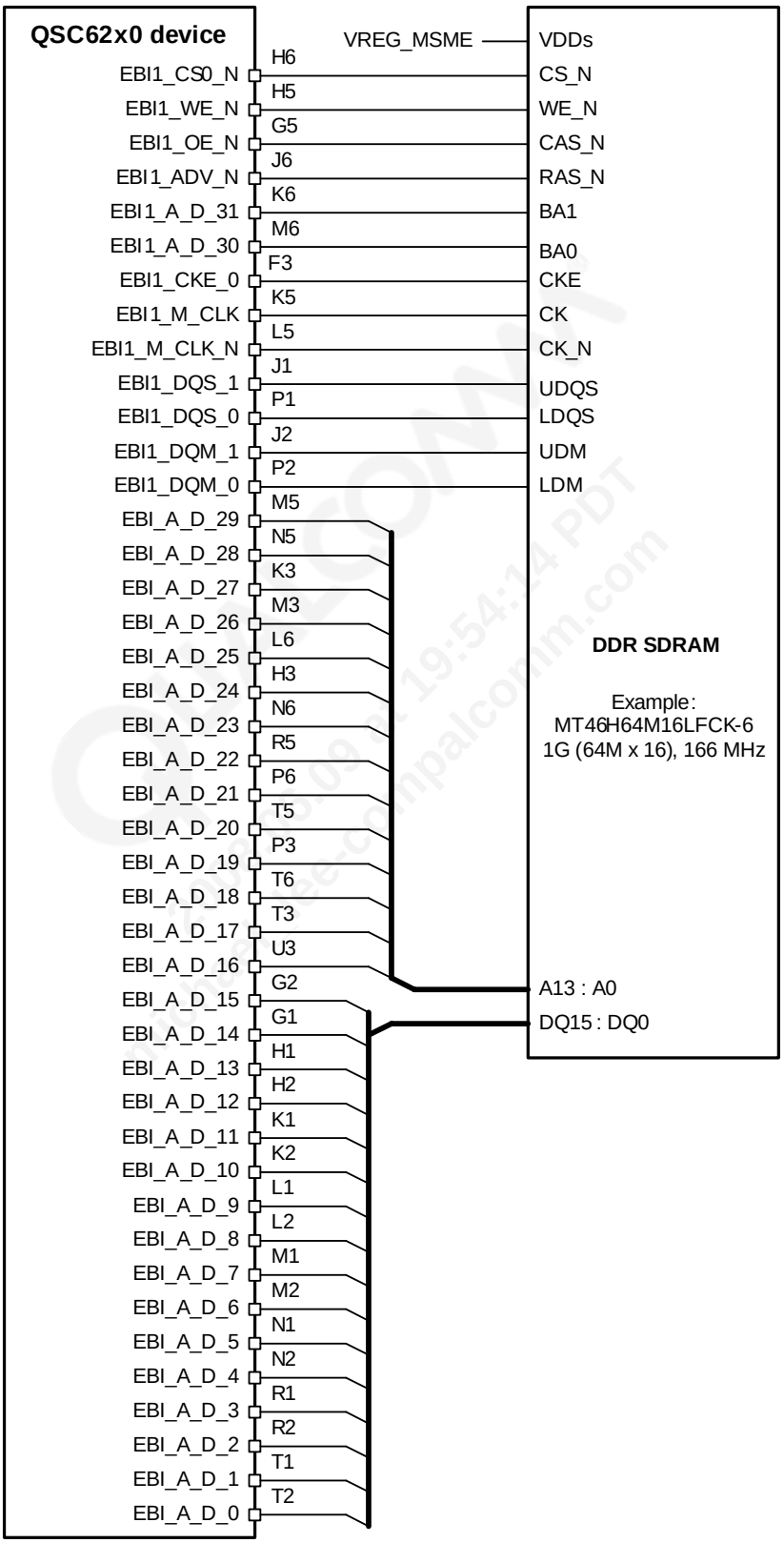


Figure 6-1 Connectivity to one 16-bit DDR SDRAM memory device

6.1.3.2 Connecting two DDR SDRAM chips on EBI1

The QSC62x0 device contains software-programmable registers for its DDR SDRAM controller and external DDR SDRAM devices; register functions include:

- Configuring the internal DDR SDRAM controller with parameters of the devices connected to the EBI1 chip-selects
- Configuring the DDR SDRAM devices
- Configuring the DDR SDRAM devices' refresh times

The QSC device includes two EBI1 chip-selects (EBI1_CS0_N and EBI1_CS1_N) that share the same set of registers used to configure and access DDR SDRAM devices connected to them. This creates a limitation when using both chip-selects: since only one set of configuration registers is available, the DDR SDRAM controller assumes that both chip-selects are connected to DDR SDRAM devices of the same density. Even if DDR SDRAM devices of different densities are connected, only one set of configuration information can be programmed into the QSC registers. In this case, the DDR SDRAM controller is not correctly configured to control and access the second DDR SDRAM device. This may lead to control and/or access issues, especially related to refresh timing.

To avoid potential control or access issues, two sets of recommendations are presented below: one for DDR SDRAM devices of equal density and one for DDR SDRAM devices of different densities.

Two DDR SDRAM devices of equal density

When using two separate blocks of DDR SDRAM on EBI1, each with its own chip-select, JEDEC standard-compliant DDR SDRAM devices of the same density (same row and column sizes) must be used. The DDR SDRAM devices can operate in either linear or interleaved mode (software selectable). This configuration is shown in Figure 6-2.

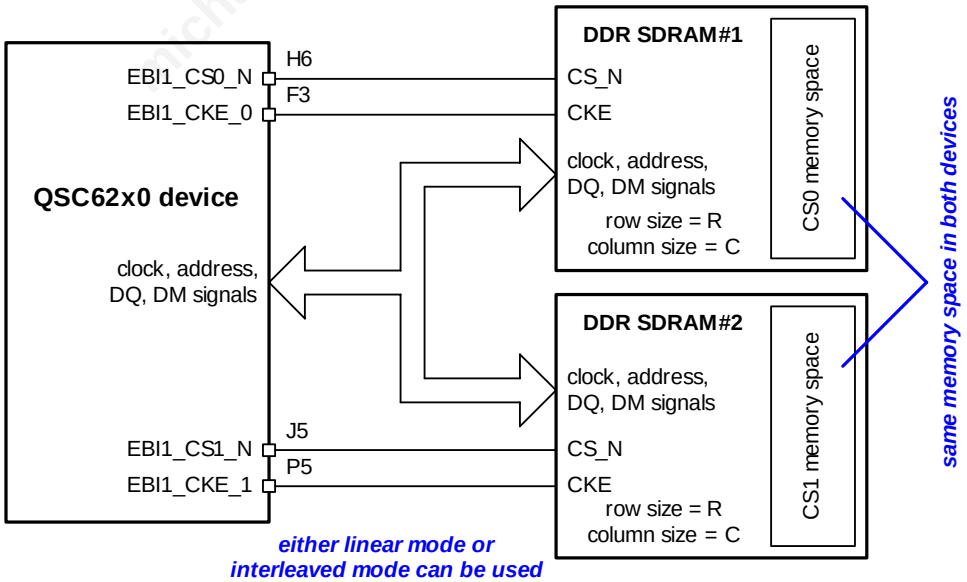


Figure 6-2 EBI1 connections to two DDR SDRAM chips (same density)

Two DDR SDRAM devices of different densities

When two DDR SDRAM devices of different densities are connected to EBI1, the following conditions **must** be met:

- The larger size memory device is connected to EBI1_CS0_N.
- The column size of both DDR SDRAM devices must be the same. DDR SDRAM devices with different column sizes are not supported.
- The two DDR SDRAM devices must operate in the linear mode — this requires proper software configuration.

This configuration is shown in Figure 6-3.

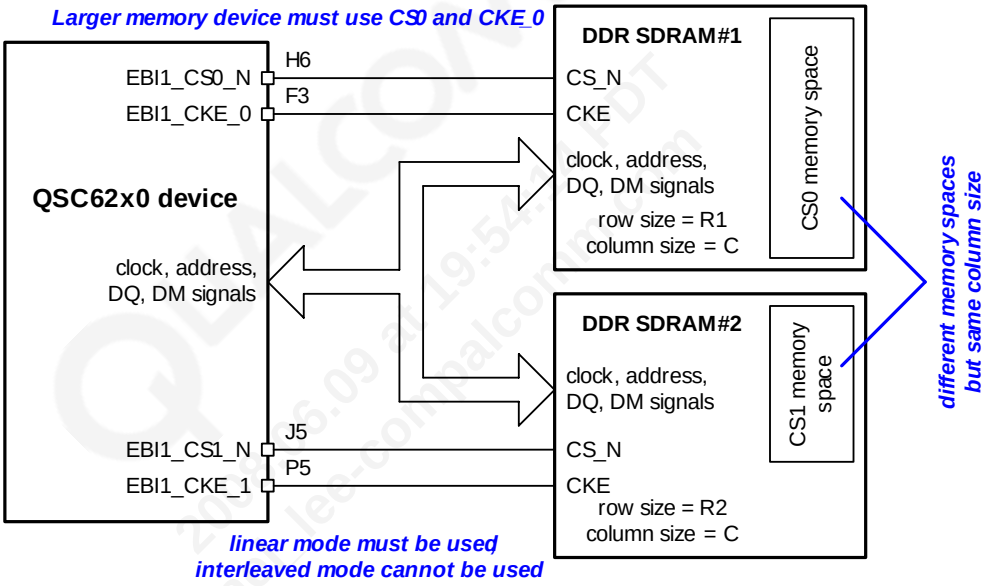


Figure 6-3 EBI1 connections to two DDR SDRAM chips (different densities)

6.1.4 DDR SDRAM controller

A DDR SDRAM uses a double-data-rate architecture to achieve high-speed operation. The double-data-rate architecture has an interface that is designed to transfer two data words per clock cycle and one word per half-clock cycle at the I/O pins.

DDR SDRAM is a source-synchronous interface that requires that the device supplying data should also supply the clock. For this purpose, DDR SDRAM has a bidirectional signal DQS (DQ strobe) used as the clock during the data transfer. During a read operation, SDRAM drives data on the DQ and DQS signals. In a write to the SDRAM, the QSC device drives data on DQ and DQS signals. The DQS signal is used to capture the data at the receiver end.

Topics specific to the use of an external DDR SDRAM are discussed in this section.

6.1.4.1 DDR SDRAM system requirements

The QSC's EBI1 SDRAM controller supports most SDRAM commands. These commands are issued through configuration registers within the SDRAM controller.

More information regarding the registers discussed in this section can be found in the *QSC6240/QSC6270 QUALCOMM Single Chip Software Interface* (80-VF846-2) document.

Initialization

After QSC powerup, the DDR SDRAM device is configured as part of the EBI1 configuration sequence — two DDR SDRAM registers (mode register and extended mode register) must be initialized. There are two different ways to initialize these registers:

1. Upon software request to write to the register, the controller FSM (finite state machine) generates the command sequence and initializes the register (the command sequence for configuring DDR SDRAM registers takes one clock).
2. The controller provides a mechanism for the software to generate the command sequence to the DDR SDRAM through a register write.

This describes the initialization flow of the DDR SDRAM device. The initialization sequence is device-specific and the flow given here is for a reference flow. Refer to the device datasheet for a detailed description of the initialization flow sequence.

1. The system is initialized.
2. The system is idle for 200 µs.
3. Issue at least one NOP command (CS[1:0] = H, RAS = 1, CAS = 1, WE = 1).
4. Issue a precharge command to all of the banks.
5. Initialize the SDRAM device mode register (per the device datasheet).
6. Initialize the SDRAM device extended mode register (per the device datasheet).
7. Issue an auto-refresh command to all of the banks.

The SDRAM_DEV_INIT_CFG register contains all the signals required to initialize the external SDRAM device. When software writes a value into this register, the controller FSM goes into its configuration mode as early as possible and maps the bits of this register to the SDRAM interface pins. Using this mechanism, the software can update the mode and extended mode registers within any SDRAM device.

The following procedure outlines the sequence for a mode register access:

1. The software writes data to this register.
2. The software writes to the configuration access register.
3. The state machine completes its pending transaction and moves to the configuration state.
4. The state machine controls the interface module to take the MRS values from the register storage.
5. The interface module outputs the contents of this register to the I/O pads.
6. The state machine clears the contents of this register.
7. The state machine completes the MRS access.

Refresh logic

The DDR SDRAM memory devices must be refreshed at regular intervals to retain the stored data. The QSC62x0 device supports both auto-refresh and self-refresh commands. The purpose of the self-refresh command is to place the DDR SDRAM into a self-refresh mode where it retains its data without external clocking. This is most desired when the QSC device powers itself down. Refer to the software interface document (80-VF846-2) for a detailed description of registers.

6.1.5 General-purpose buffers

This block performs write buffering for all incoming accesses. It enables coalescing of single-write accesses within an eight-word page. It also buffers bursts to allow for SDRAM optimizations and facilitates fixed-length bursts to the burst memory controller. There are eight such buffers. This block also comes with flushing logic to appropriately flush the buffers in various situations.

More information about the buffering architecture will be added in a future revision of this document.

6.1.6 EBI1 layout guidelines

The extremely high data and clock rates supported by the QSC62x0 EBI1 require careful PCB layout and routing techniques. Guidance is provided in *Application Note: MSM7xxx DDR SDRAM Design Considerations*, 80-V9038-54.

NOTE The PCB layout guidelines presented in *Application Note: MSM7xxx DDR SDRAM Design Considerations* (80-V9038-54) are highly recommended for QSC62x0 EBI1.

Some of the most important layout guidelines are listed below.

General layout guidelines

The PCB layout requires extra care and attention to preserve the high-speed EBI1 performance — in particular, the loads seen by the bus must be minimized and the jitter on all signals must be limited. The following signals must have similar PCB characteristics (such as trace length, capacitive loading, and logic delay) in order to achieve high rates:

- The differential clock signals (EBI1_M_CLK, EBI1_M_CLK_N) should be routed together and have the same length between the QSC device and the memory device(s).
- Every eight bits of DQ data (D[15:8], D[7:0]) have corresponding DQS and DQM bits on the SDRAM interface. Each data byte and its corresponding DQS and DQM bits are considered a group.
- All of the other signals (RAS, CAS, WE, CS, CKE, and A[n:0]) should have similar delays.
- All signal outputs are source-terminated. Additional source termination is not required.

Package and PCB impedances

Keep the QSC62x0 package and PCB impedance characteristics in mind when designing the high-speed interface to external memory devices. QCT provides electrical models (IBIS files) for the drivers and the package. It is recommended that designers run system-level simulations using these models to ensure overall high-integrity system performance.

While maximum performance is limited by the ICs, handset designers must focus on the PCB parameters under their control to ensure that the specified performance is maintained. Use controlled-impedance traces (microstrip or stripline) between the QSC and memory devices. The line impedance depends upon several variables — the trace width and thickness, the height of dielectric material between the trace and ground plane(s), and the dielectric constant of the PCB material. Given the PCB material selected, the geometry of the microstrip or stripline elements must be designed properly to provide the desired controlled impedance value. The design of microstrip and stripline elements is well documented in design literature and supported in many microwave software applications.

Additional high-speed trace-design guidelines include:

- When routing on an external layer, use microstrip techniques where possible.
- When routing on internal layers, use stripline techniques where possible.
- Maintain continuous ground below microstrip traces and above and below stripline traces.
- Keep traces short and direct to minimize loss and undesired coupling.
- When finished routing the signal on the board, fill any available areas with ground, providing adequate clearance to minimize co-planar capacitance and leakage.
- Use several ground vias to connect outer ground fill areas to the internal ground planes.
- Avoid crossing gaps in ground planes for referencing high-speed traces.
- Clear the internal layer (or layers) of metal immediately below (and above) signal traces and component pads to reduce parasitic capacitances (for example, clear metal layer 2 below layer 1, as shown in Figure 6-4). This also improves microstrip (or stripline) geometries, allowing wider traces. A cross-sectional view of this important technique is shown in Figure 6-4 for microstrip implementations.

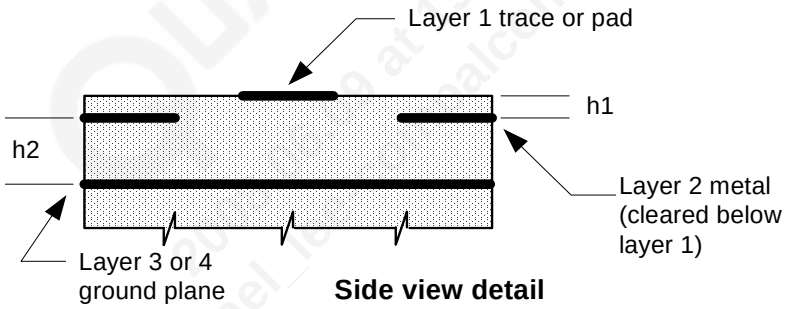


Figure 6-4 Clearance below component pads and signal traces

Differential clock trace layout

A special case of high-speed trace routing is the differential clock signal (EBI1_M_CLK and EBI_M_CLK_N). Guidelines (in addition to those listed above) include:

- Use controlled impedance lines for each of the two differential traces.
- Route the two traces in parallel and close to each other. If the trace width is W, the spacing between traces should be on the order of 2 W to 3 W.
- The two traces must have nearly equal length; otherwise, the complementary phase relationship between the two is degraded (impacting the timing, duty cycle, and possibly the detection margins).

DQ/DQS skew

The DQS signals are like clocks to the QSC device and DDR SDRAM circuits. The QSC62x0 device uses the DQS signals to capture data and the DDR memory device uses DQS signals during writes. The timing difference (delay) between the DQ and DQS signals needs to be minimized — the timing must be well matched.

There are two DQS lines on the EBI1 bus, with each DQS signal associated with its own group of data (DQ) lines:

- SDRAM_DQS[0] ↔ SDRAM_D[7:0]
- SDRAM_DQS[1] ↔ SDRAM_D[15:8]

6.2 EBI2

EBI2 is used to interface with slower memory and peripheral devices (NAND flash, burst NOR, LCDs, etc.). The following EBI2 devices are supported:

- NAND flash
  - 8/16 bit, single-level cell (SLC)/multi-level cell (MLC) 512/2048-byte page devices
  - DMA support
  - Boot-up capability from the above devices
- Burst NOR flash
  - 16-bit multiplexed AAD burst NOR devices
- 8/16/18-bit (write only) LCD devices (both Motorola and Intel style)

Characteristics of the EBI2 clock are listed below:

- The maximum clock rate is 46 MHz, defined by the AMSS software.
- The EBI2 memory controller operates at HCLK/2.
- Broadcasting and multicasting (QSC6270 only, with MBP1600 IC) are based on:
  - Wideband MediaFLO™, DBV-H, and ISDB-T

6.2.1 EBI2 features

6.2.1.1 EBI2 memory interface features

- Support for WORD, HWORD, and BYTE accesses to 16-bit and 8-bit wide devices on all chip-selects (bus-sizing supported for all interfaces with the exception of NAND)
- Up to four chip-selects; two are configurable GPIOs
- A NAND controller that allows a software-programmable command to be issued to the NAND device
  - This flexibility allows the chip to support other NAND devices that require a new command not supported by the controller.
- EBI2 boot-up from a 16-bit asynchronous interface type device
  - Boot-up will be delayed using a 10-bit counter for 53.3 µs (at 19.2 MHz).
  - Because the timer is monitoring the BUSY_N pin, a pull-up device is necessary.
  - EBI2 operation at HCLK/2 with a maximum frequency of 46 MHz.
  - EBI2 devices such as LCD, NAND are typically slower devices with typical access times around 100 ns.
  - Enabling lower-speed burst devices in the 40 MHz to 50 MHz range allows the more widely available burst-NOR flash devices to be supported.
- Support for only address/data MUXed devices on EBI2

6.2.1.2 LCD device-controller features

The LCD device controller in the xmem_ctlr includes the following features:

- Support for 8/16/18-bit (write-only) Intel (8080) interface type of devices on general-purpose chip-selects and LCD_CS, coupled with the use of LCD_RS
- Support for 8/16/18-bit (write-only) Motorola (6800) interface type of devices on LCD_CS coupled with the use of LCD_EN and LCD_RS
- Support for port-mapped parallel interfaces
- Support for insertion of WAIT and HOLD cycles for both Intel and Motorola interfaces
- Support for E_SETUP and E_HIGH cycles for Motorola interface
- Support for bus-sizing based on the size of the LCD device
- Dedicated LCD signals for chip-select, enable, and register select

6.2.2 EBI2 connections

All EBI2 connections are listed in Table 6-4; a typical EBI2 application appears in Figure 6-5.

Table 6-4 EBI2 connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
EBI2_M_CLK	A2	P2 (1.8/2.6 V)	DO	(5.6/11.2)	EBI2 memory clock
EBI2_WE_N	E6	P2 (1.8/2.6 V)	DO	(5.6/11.2)	EBI2 write enable
EBI2_OE_N	F2	P2 (1.8/2.6 V)	DO	(5.6/11.2)	EBI2 output enable
EBI2_LB_N	C6	P2 (1.8/2.6 V)	DO	(5.6/11.2)	Memory address latch enable; LCD bit 16
EBI2_UB_N	F1	P2 (1.8/2.6 V)	DO	(5.6/11.2)	Memory command latch enable; LCD bit 17
EBI2_CS3_N (GPIO_48)	B7	P2 (1.8/2.6 V)	B	(5.6/11.2)	EBI2 chip-select; EBI2_BUSY_N
EBI2_CS2_N (GPIO_49)	A7	P2 (1.8/2.6 V)	DO	(5.6/11.2)	EBI2 chip-select
EBI2_CS1_N	E5	P2 (1.8/2.6 V)	DO	(5.6/11.2)	Chip-select for NAND and OrNAND
EBI2_CS0_N	E2	P2 (1.8/2.6 V)	DO	(5.6/11.2)	Chip-select for OneNAND and NOR
LCD_CS_N	C7	P2 (1.8/2.6 V)	DO	(5.6/11.2)	LCD chip-select
LCD_EN	F5	P2 (1.8/2.6 V)	B-PU	(5.6/11.2)	LCD enable; EBI2_WAIT_N
LCD_RS	B6	P2 (1.8/2.6 V)	B-NP	(5.6/11.2)	LCD register select; EBI2_ADV_N
EBI2_A_D_15	A5	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 15
EBI2_A_D_14	B5	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 14
EBI2_A_D_13	C5	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 13
EBI2_A_D_12	A4	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 12
EBI2_A_D_11	C4	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 11
EBI2_A_D_10	A3	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 10
EBI2_A_D_9	B4	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 9
EBI2_A_D_8	D3	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 8
EBI2_A_D_7	E3	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 7
EBI2_A_D_6	B3	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 6
EBI2_A_D_5	C2	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 5
EBI2_A_D_4	B1	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 4
EBI2_A_D_3	C1	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 3
EBI2_A_D_2	D2	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 2
EBI2_A_D_1	D1	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 1
EBI2_A_D_0	E1	P2 (1.8/2.6 V)	B-K	(5.6/11.2)	EBI2 address or data bit 0

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

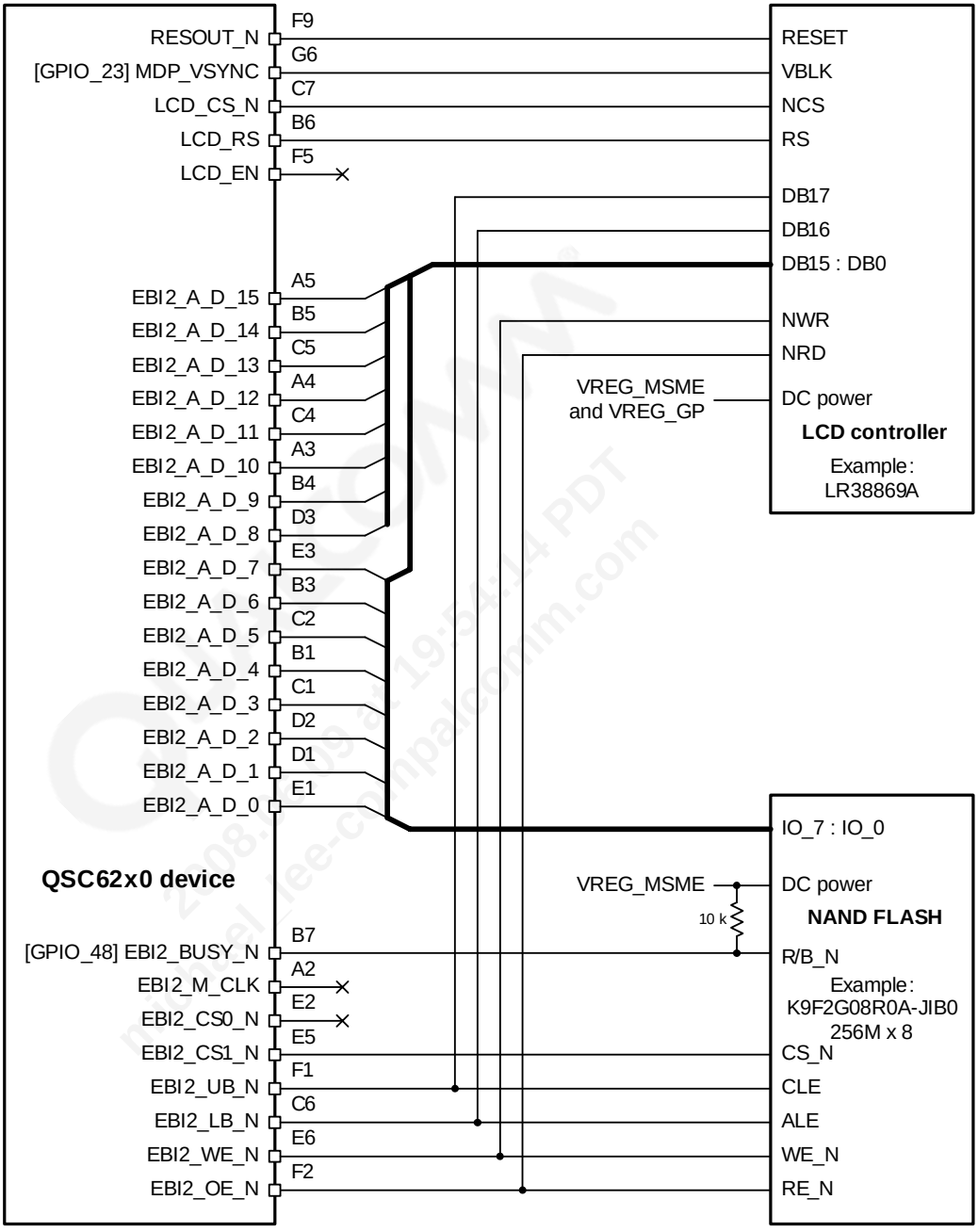


Figure 6-5 Typical EBI2 application example

6.2.3 NAND controller

The NAND controller (Figure 6-6) has five major functional blocks and communicates with on-chip and off-chip functions through two interfaces:

- Five major NAND controller functional blocks:
  - NANDC FSM (NANDC_FSM): This is the serial flash-controller state machine. It interfaces with any serial flash device that has timing and interface characteristics that are similar to NAND flash devices. The controller can be designed as a sequencer with a simple instruction set or as a hardwired state machine.
  - NANDC interface (NANDC_IF): This block generates all the interface signals to the flash device. Serial flash devices are block/page access devices, with read/write (program) operations performed on a page basis. This block contains the buffer (RAM) to fit the flash page data, and is address-accessible through the AHB interface. For higher reliability and to reduce the bit error rate (BER), parity should be added to the data stored in the flash device. The flash controller block has error-correcting code (ECC) logic that generates parity for the data; parity is generated based upon codewords. Each page in the flash device can use single or multiple codewords. The ECC block generates parity when the controller is writing (programming) user data to the flash device and the controller writes the parity at the end of the user data for each codeword. During the read operation, the logic fetches the user data and parity data for each codeword, then the ECC logic decodes the user data and parity data to detect any errors in the data. If errors are detected, it generates error values and the logic corrects the error. For this purpose, the logic has to store the entire codeword during the read operation, a requirement met by the NANDC block. The actual codeword size for QSC62x0 is 522 bytes; NAND flash devices provide 528 bytes for each codeword.
  - NAND data mover interface block (NANDC_DM): This block generates interface signals to the system's data mover block and generates interrupts to the processor to indicate completion of read/write/erase accesses.
  - NAND command decode (NANDC_CMD_DECODE): This block decodes the command opcode and configuration registers and provides the decoded signals to the other blocks in the system.
  - ECC: The ECC block handles error detection, error correction, and parity generation.
- NAND controller interfaces:
  - Xmem controller: NAND flash devices sharing EBI2 pins with other types of memory devices are controlled by the Xmem controller. Signals from the NAND controller go through the Xmem controller's multiplexer logic before reaching the EBI2 pins.
  - AHB interface: This block is a slave interface to the 32-bit AHB bus; it includes all the software control/configuration registers.

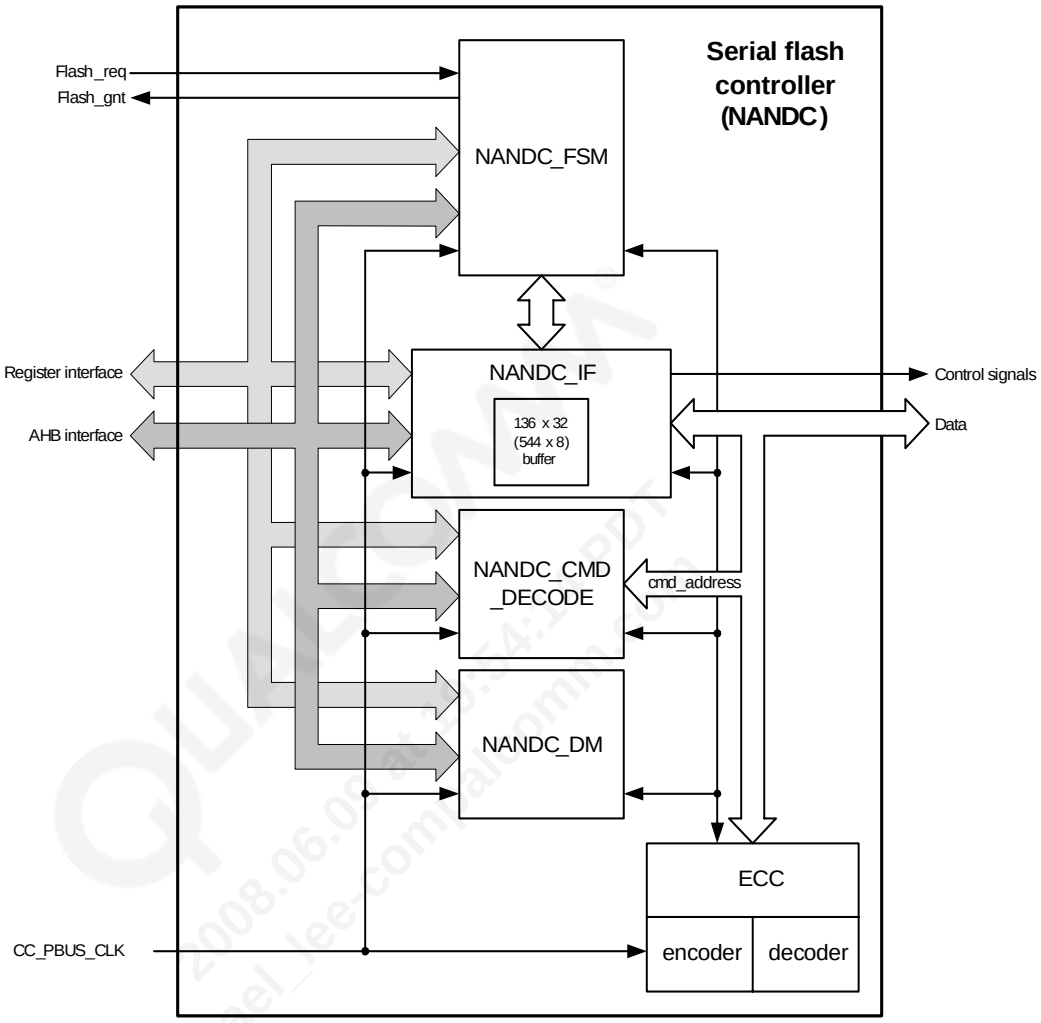


Figure 6-6 NAND controller functional block diagram

6.2.4 NAND flash memory interface

The QSC62x0 device supports NAND flash memory to improve data storage capability, including:

- The generic NAND flash memory interface supports 16-bit and 8-bit NAND flash devices from manufacturers such as Samsung and Toshiba.
- Programmable page sizes of 512 byte and 2 kB are supported (256-byte is not supported).
- The NAND interface uses the EBI2 port, sharing this port with other external devices such as external SRAM.
- The interface performs data transfers one page at a time to and from the NAND device. Sequential page accesses are not supported.

- The interface has an internal SRAM buffer for one data page; the ARM processor directly accesses this internal buffer memory through the AHB bus using 32-bit word accesses.
- Both multilevel cell (MLC) and single-level cell (SLC) mass-storage technologies are supported (MLC NAND devices are currently not supported by AMSS).
- ECC: Reed-Solomon code ECC is used to support MLC-based devices. The Reed-Solomon ECC has four-symbol correction per 512 bytes of data, where a symbol consists of eight bits of user data and two padded zero bits. Therefore, the correction capability is equivalent to four bytes per 512 bytes of user data. A 512-byte page is considered a single ECC codeword, and a 2-kB page is considered four ECC codewords by the ECC engine. The ECC function can be disabled by the QSC processor.

For more information regarding the NAND flash configuration registers, please refer to the *QUALCOMM Single-Chip QSC6240/QSC6270 Software Interface* (80-VF846-2) document.

Additional NAND-related topics are discussed in the following subsections.

6.2.4.1 Support for two NAND flash devices

The NAND controller treats the whole NAND flash storage space as a series of pages. By default, the controller assumes that a single device is used in the system. If there are two NAND devices attached to the QSC device, then the configuration of each device (I/O width and page size) and the boundary between the two devices has to be programmed by software. The boundary of the two devices is defined as the first page of the second device.

The controller's access to each device is completely page-based; once a page access to a device begins, the access must be completed before the controller can start another page access to the same (or other) device. Therefore, the controller never operates on two devices simultaneously.

6.2.4.2 Supported mass storage technologies

Two mass-storage technologies are supported: SLC and MLC.

The SLC technology is the original mass storage solid state technology used in flash devices. Every storage transistor has two different states and, therefore, contains one bit of information. The SLC is sometimes referred to as 2LC since each transistor has two states. Based upon the data reliability level of the SLC technology, vendors recommend the Hamming code with 1-bit correction per 512 bytes of data as the ECC scheme to achieve acceptable overall data reliability.

The MLC technology is a newer solid-state technology where each storage transistor has four states, and therefore contains two bits of information. The MLC is sometimes referred to as 4LC since each transistor has four states. Compared to SLC, the MLC technology provides higher data capacity in the flash device, but slower read/program/erase speeds and lower data reliability. Based upon the data reliability level of the MLC technology, vendors recommend the Reed-Solomon code with 4-bit correction per 512 bytes of data as the ECC scheme to achieve acceptable overall data reliability.

NOTE MLC NAND is supported in hardware in QSC62x0 devices. Software support for MLC NAND devices is currently not available.

6.2.4.3 NAND-flash page arrangement

In NAND devices, data is stored in sectors. Each sector of data is called a page. Any access from the controller to the NAND device results in accessing the entire page. A write or program to the NAND device is always page-oriented. Each page can have multiple codewords; hence, the controller data transfer is always in multiples of a codeword. A codeword for the controller is 512 bytes of user data + 10 bytes of parity data. A NAND flash device has room for 512 bytes of user data + 16 bytes of spare area for the codeword. Each page has one byte (or one word, depending on the I/O interface size) reserved for the bad-block status. To make it simple, the controller always reserves 1 byte/1 word per codeword for the bad-block status. The bad-block status location is not fixed across different NAND flash devices.

The following figures (Figure 6-7 and Figure 6-8) show the page arrangements of different NAND flash devices. The controller has programmable registers to indicate the bad-block status byte location.

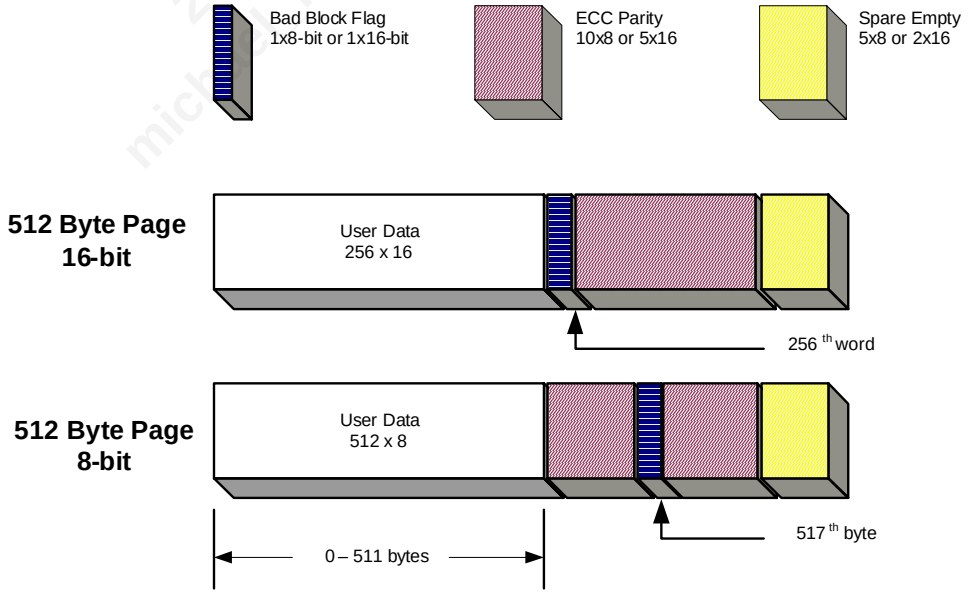


Figure 6-7 Page format description (8-bit and 16-bit — 512-B/page)

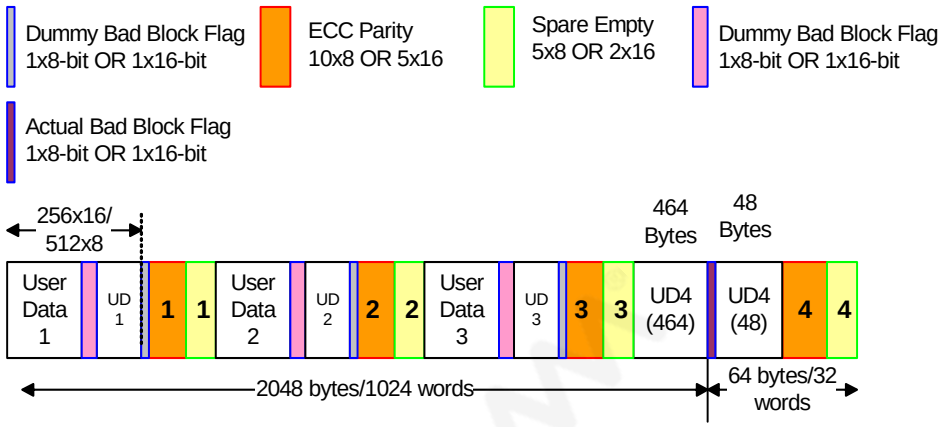


Figure 6-8 Page format description (8-bit and 16-bit — 2 kB/page)

6.2.4.4 ECC for NAND devices

The QSC62x0 device uses Reed-Solomon error correction coding and decoding. The controller supports both 512-byte and 2-kB page sizes. Since each ECC codeword covers 512 bytes of user data, the 512-byte page is arranged to store one ECC codeword, and the 2-kB page is arranged to contain four ECC codewords.

6.2.4.4.1 ECC modes

ECC for NAND 8/16-bit interface (512-B page)

The ECC module provides Reed-Solomon encoding and error correction (decoding) of the 512-byte data packets (plus 10 parity bytes for the decoder), operating in one of two modes (illustrated in Figure 6-7):

- 8-bit wide 512-byte NAND flash device
  - The 512-byte page actually has (512 + 16 = 528) bytes of storage space in the flash device as explained above.
  - Each page is treated as one ECC codeword space.
  - The first 512 bytes of space are used to store user data, followed by five bytes of ECC parity code, followed by one byte bad-block status, followed by five more bytes of parity. The last five bytes are unused.
- 16-bit wide 512-byte NAND flash device
  - 512-byte (16-bit mode) page actually has (256 + 8 = 264) 16-bit words of storage space in the flash device.
  - Each page is treated as one ECC codeword space.
  - The first 256 16-bit words of space are used to store user data, followed by one 16-bit word bad block status, followed by five 16-bit words of ECC parity code. The last two 16-bit words are unused.

The ECC allocations are given in Table 6-5 and Table 6-6 for 8-bit and 16-bit NAND interfaces, respectively.

Table 6-5 ECC allocation – 8-bit NAND interface (512-B page)

Byte	ECC allocation	Notes
Byte_00	ECC Parity_0	10 parity length for Reed-Solomon decoder (8-bit interface)
Byte_01	ECC Parity_1	
Byte_02	ECC Parity_2	
Byte_03	ECC Parity_3	
Byte_04	ECC Parity_4	
Byte_05	Bad-block indicator (0xFF)	
Byte_06	ECC Parity_5	
Byte_07	ECC Parity_6	
Byte_08	ECC Parity_7	
Byte_09	ECC Parity_8	
Byte_10	ECC Parity_9	
Byte_11 to Byte_15	Unused	

Table 6-6 ECC allocation — 16-bit NAND interface (512B page)

Word	ECC allocation	Notes
Word_00	ECC Parity_1 [7:0] ECC Parity_0 [7:0]	
Word_01	ECC Parity_3 [7:0] ECC Parity_2 [7:0]	5 parity length for Reed-Solomon decoder (16-bit interface)
Word_02	ECC Parity_5 [7:0] ECC Parity_4 [7:0]	
Word_03	ECC Parity_7 [7:0] ECC Parity_6 [7:0]	
Word_04	ECC Parity_9 [7:0] ECC Parity_8 [7:0]	
Word_05	Bad-block indicator (0xFFFF)	
Word_6 to Word_7	Unused	

ECC for NAND 8/16-bit interface (2 kB page)

The ECC module provides Reed-Solomon encoding and error correction (decoding) of the 512-bytes data packets (plus 10 parity bytes for the decoder), operating in one of two modes (illustrated in Figure 6-8):

1. 8 bits x 512 input block length (plus 8 bits x 10 parity length for the decoder)
2. 16 bits x 256 input block length (plus 16 bits x 5 parity length for the decoder)

The page is evenly divided into four 528-byte/264 16-bit word sections, with each section arranged similarly to the 512-byte page arrangement. A 2-kB page contains 512 bytes/256 16-bit words of user data, followed by 1 byte/1 16-bit word bad-block status, followed by 10 bytes/5 16-bit words of ECC parity code, and 5 bytes/2 16-bit words of unused data. That completes the first codeword. The second codeword follows as shown in Figure 6-8.

The QSC62x0 device has different ECC/spare area layouts from the previous-generation MSM products.

6.2.4.4.2 ECC programming

The ECC module provides encoding and decoding of the 512-B data blocks. During the encoding process, 10 B of parity are added to the data block. The correction capability of the code is four bytes.

The code is based on the Reed-Solomon family of codes with symbols from the Galois field GF(2^{10}). The codeword is 1023 of the 10-bit symbols, including eight parity symbols. Figure 6-9 shows how 512 bytes of user data and eight 10-bit ECC parity symbols fit into an ECC codeword of 1023 symbols.

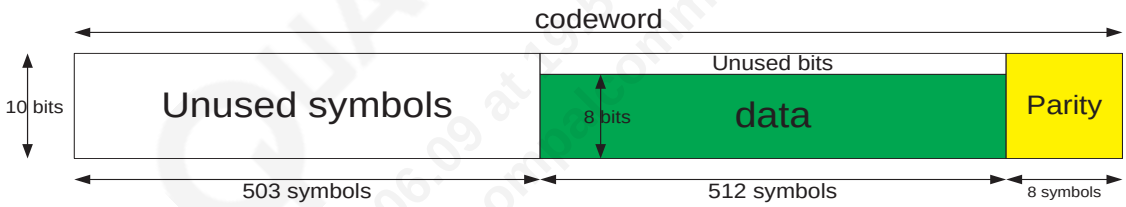


Figure 6-9 ECC codeword

The 10-bit symbols are elements of the Galois field GF(2^{10}) generated by the primitive polynomial $p(x) = 1 + x^3 + x^{10}$. There are two representations of the elements: binary and power index of α representations, where α is the notation for the third element (after zero and one) of the field ("0000000010" = 2 in binary representation). The binary representation is used for all signals in the design. The power index of α representation is used for the constant notations in the multipliers by constant. There are three algebraic operators in this field: addition (+), multiplication (x), and division (/).

The code generation polynomial is:

$$g(x) = (x + \alpha^{b0})(x + \alpha^{(b0+1)\%1023})(x + \alpha^{(b0+2)\%1023})(x + \alpha^{(b0+3)\%1023})$$
$$(x + \alpha^{(b0+4)\%1023})(x + \alpha^{(b0+5)\%1023})(x + \alpha^{(b0+6)\%1023})(x + \alpha^{(b0+7)\%1023})$$

where:

$\alpha^{b0}, \alpha^{(b0+1)\%1023}, \alpha^{(b0+2)\%1023}, \alpha^{(b0+3)\%1023}, \alpha^{(b0+4)\%1023}, \alpha^{(b0+5)\%1023}, \alpha^{(b0+6)\%1023}, \alpha^{(b0+7)\%1023}$

are roots of the generation polynomial, % is the modulus function, and b0 is an arbitrary number between 0 and 1022.

6.2.4.5 Booting from a NAND flash device

When the QSC device powers up, it has to execute the on-chip primary boot loader (PBL). The boot ROM software does not have any knowledge of what type of NAND device is attached to the EBI2 interface. To properly program the NAND flash controller to download the boot loader, the QSC device needs to determine the NAND device-type.

On EBI2, any one of the following flash devices can be connected:

- 512-B page 8-bit NAND device
- 512-B page 16-bit NAND device
- 2-KB page 8-bit NAND device
- 2-KB page 16-bit NAND device

Unless the connected device type is known, the contents of the device cannot be downloaded. For any of the possible devices connected, if the page size and the bus interface width is known, data can be read from it. The EBI2 controller can detect the page size of the flash device connected to it. The device parameters that need to be determined are:

- 8-bit or 16-bit I/O width
- 512-B or 2-kB page size

The addressing schemes, such as number of column address cycles and number of row address cycles, are different for a 512-B page device than a 2-kB device. The controller uses this to detect the device that is connected. Further details are provided in Section 6.2.4.6.

Boot-up using EBI2_BUSY_N

If the BUSY_N pin is at logic 0 at the end of t(ebi2_boot), the boot-up will be further delayed until the assertion of logic 1 on the BUSY_N pin (Figure 6-10). This pin must be driven high when the chip is programmed to boot from EBI2.

Note that NAND devices drive the BUSY_N pin. Therefore, a pull-up device on this pin would be the preferred approach. Care must be taken to ensure that the drive strength of the NAND device is greater than the drive strength of the pull-up device. Also, the pull-up should not be too weak as it affects the boot-up time.

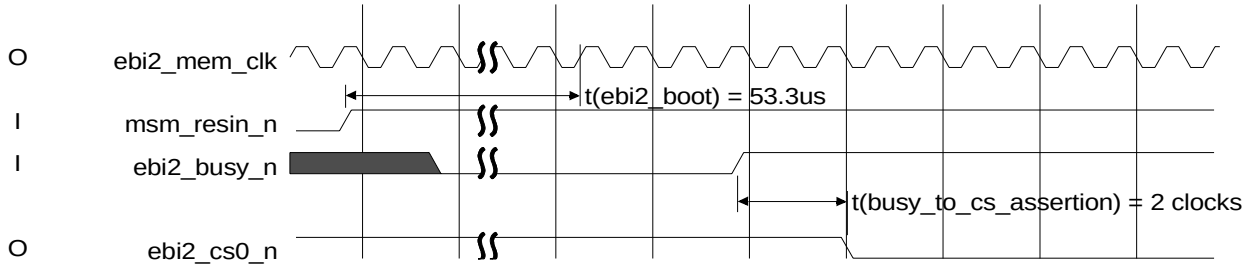


Figure 6-10 EBI2 boot up scheme with BUSY_N monitoring

6.2.4.6 QSC NAND device detection

The NAND device detection process (Figure 6-11) is done automatically. When the AUTO_DETECT bit is set in register NAND_FLASH_CMD, the controller executes this part of the state machine and sets the AUTO_DETECT_DONE bit in register NAND_FLASH_STATUS to indicate that the controller detected the device size. It also sets a bit indicating the page size of the device (NAND_DEV_SIZE) in the NAND_FLASH_STATUS register. The controller has a time-out counter to generate an interrupt in case the flash is not responding.

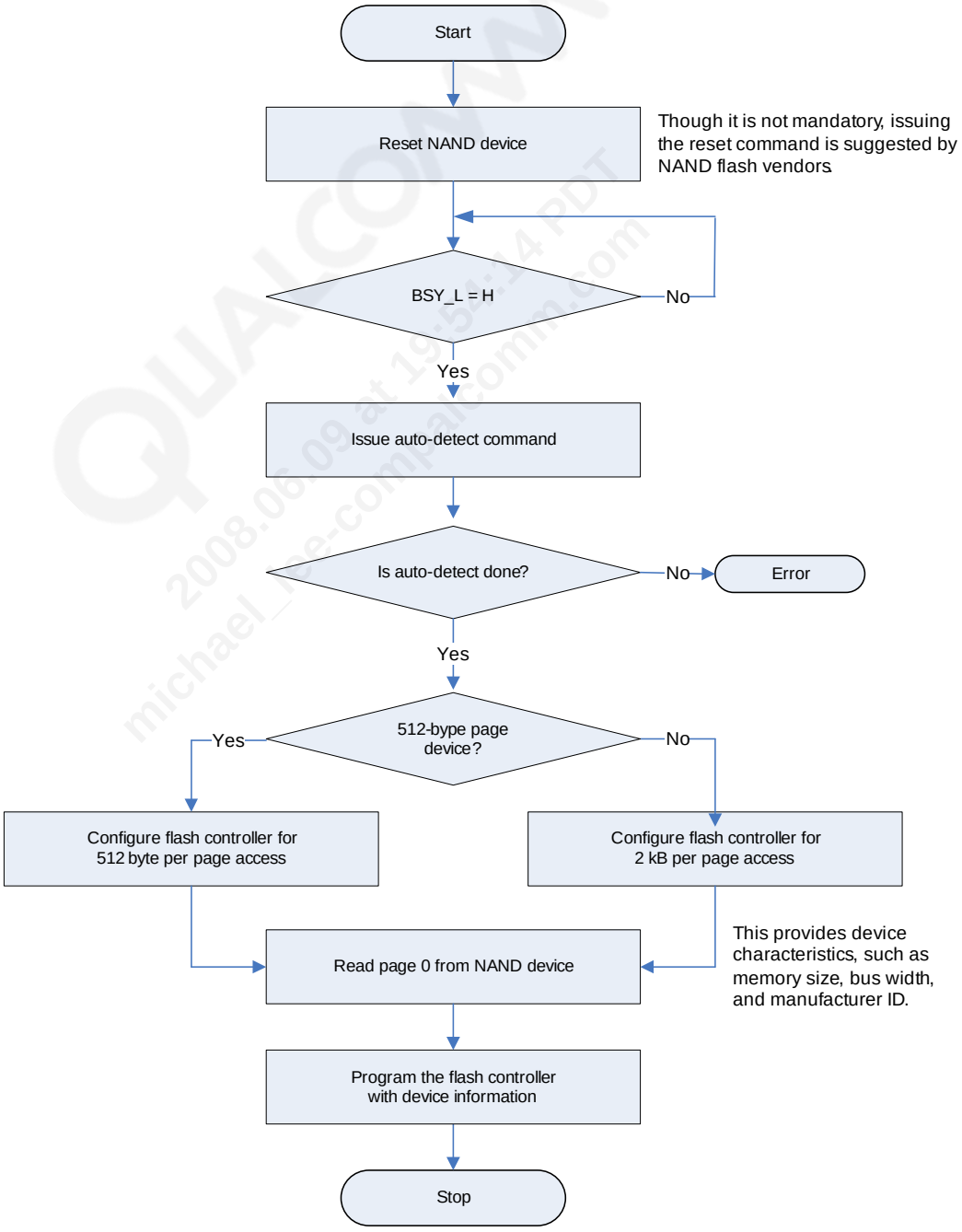


Figure 6-11 NAND flash-detection process

As part of the AUTO_DETECT sequence, the controller can be programmed to fetch data from page 0, block 0 of the flash device. The first step in the detection process is to detect the NAND device page size. The NAND controller uses the difference in the read structure between the 512-B page size NAND devices and the 2-kB page size NAND devices. This difference is used to detect the page size of the device connected. After detecting the device page size, the controller does a read data transfer in 8-bit mode of page 0 (even if the connected device is 16-bit, the controller only uses the lower 8-bit data from the device), then pack the data into the NAND flash buffer. The data read from the flash device can be used to detect the bus size and any other parameters of the flash.

After successfully detecting the page size and getting the data into the flash buffer, the software can read the data and identify the NAND configuration. By detecting the page size of the NAND device, the addressing mechanism (number of ROW, COLUMN address cycles) is known. With this information, the QSC device can address any page of the NAND device.

6.3 Asynchronous/burst controller (EBI1 and EBI2)

The external memory controller (xmem_ctlr) forms the asynchronous/burst controller for both EBI1 and EBI2 in the QSC62x0 device. The controller is generic in terms of its software programmable options and can be customized when used for EBI1 and EBI2. This block has been enhanced in the QSC62x0 device to support 32-bit burst memories and byte masking during write operations.

6.3.1 Features

The memory controller offers the following features:

- General-purpose chip-selects that can support asynchronous/burst NOR/PSRAM memories
  - Maximum memory size supported is 128 MB (on EBI1).
- One dedicated LCD chip-select (on EBI2)
- Support for WORD, HWORD, and BYTE accesses to 32-bit, 16-bit, and 8-bit asynchronous devices on all chip-selects
- Support for WORD, HWORD, and BYTE SINGLE accesses to burst NOR/PSRAM memories
- Support for fixed-length bursts of WORD accesses to burst NOR/PSRAM memories
- Support for undefined length INCR writes to burst NOR/PSRAM memories
- Support for internal address translation based on data-width of the memory device
  - As a result, a 16-bit memory device gets a half-word aligned address and a 32-bit memory device gets a word-aligned address. This is especially important for MUXed-address/data devices, because LSBs of the A_D bus also represent data and need to be connected.

6.3.2 Descriptions of operating modes

6.3.2.1 INIT_LATENCY and WAIT cycles operation

INIT_LATENCY cycles are inserted in the first access when a particular chip-select is enabled for page or synchronous burst operation, or when the access crosses a page boundary. INIT_LATENCY cycles also refers to the cycles inserted at the beginning of every access to a asynchronous memory. A minimum of one INIT_LATENCY cycle must be programmed for all accesses.

The controller supports independent programming of INIT_LATENCY cycles for read and write accesses.

NOTE INIT_LATENCY = 0 is not guaranteed to work and will produce unpredictable behavior.

The WAIT cycles are inserted for all accesses to the memory. The minimum value of WAIT that can be programmed is 0.

The controller supports independent programming of WAIT cycles for read and write accesses.

NOTE The WAIT setting must be set to 0 for burst memories.

Upon power-up, all accesses to any chip-select will incur the following number of clock cycles:

- (INIT_LATENCY_RD + WAIT_RD + HOLD_RD + 1) cycles for reads
- (INIT_LATENCY_WR + WAIT_WR + HOLD_WR + 1) cycles for writes.

Once a chip-select is enabled for a page-mode read operation, the first access to the memory always incurs (INIT_LATENCY_RD + WAIT_RD + 1) cycles. Subsequent sequential accesses within the same page incur (WAIT_RD + 1) cycles.

Example 1:

For a 50-MHz HCLK 70/25 ns page flash device, the following conditions must be satisfied (note that in reality the on-chip delay and the board-routing delay must be taken into consideration; however, we have ignored these considerations for all of the examples here):

- INIT_LATENCY + WAIT + 1 > 70 ns
- WAIT + 1 > 25 ns

So, we can select a WAIT = 1 and INIT_LATENCY = 2 to give us 80 ns for the first access and 40 ns for the sequential access within the page.

Example 2:

For a 54-MHz HCLK, 70/13.5 ns burst-flash device configuration, the following conditions must be satisfied:

- INIT_LATENCY + WAIT + 1 > 70 ns
- WAIT + 1 > 13.5 ns

So, we can select a WAIT = 0 and INIT_LATENCY = 3 to give us 74 ns for the first access and 18.5 ns for the sequential access within the page. There are two types of burst-flash devices that are supported: Intel StrataFlash and AMD parts (does not have multiplexed address/data). The AMD part does not support a WAIT = 1 setting. However, the Intel part allows two cycles for subsequent accesses.

NOTE We recommend that the WAIT values be programmed to 0 and the INIT_LATENCY values be used for all asynchronous non-page memories.

6.3.2.2 HOLD cycles operation

HOLD cycles are the extra cycles inserted after every read/write access to an asynchronous memory. A minimum value of 1 HOLD_WR value must be programmed for every write access. The data from the chip is driven from the time WE_N strobe goes low to the time when CS_N goes high. When a hold cycle value of 1 is programmed, then the CS_N stays active for one extra clock cycle.

Read accesses typically do not require HOLD cycles. The external memory controller, however, supports HOLD_RD and this is used to avoid bus contention in MUXed-address/data devices.

NOTE 1 A fixed HOLD_RD of three cycles is inserted by EBI1 and EBI2 controllers for all asynchronous accesses (except LCD, which is programmable) to avoid bus contention.

NOTE 2 This bit field is ignored for the chip-select if the burst write (CSx_BURST_WR_ENA = 1) is enabled.

NOTE 3 A HOLD_WR value of 0 is not guaranteed to work and will cause unpredictable behavior.

6.3.2.3 RECOVERY cycles operation

Recovery cycles are required to ensure that there is no contention on the data bus. This value can be obtained from the memory data sheet; it is typically defined as the amount of time that the memory continues to drive the data bus after OE_N deassertion.

Example 1:

For a 50-MHz HCLK and flash devices with 30 ns recovery time, the required condition is (RECOVERY > 30 ns). So, three cycles must be programmed for the RECOVERY value to ensure that there is no bus contention in asynchronous mode in the event of another read or write access. In the burst mode, a minimum of three recovery cycles is inserted by the controller. Refer to Table 6-7 to determine the RECOVERY programming value. Recovery cycles are inserted for the following situations:

1. Consecutive access to two different chip-selects with the first access being a read
2. Read access followed by a write access to the same chip-select

The following table provides the number of cycles that a chip-select may be deasserted using the RECOVERY programmable cycles.

Table 6-7 RECOVERY programming vs. CS high cycles

Programmed value	CS high cycles		
	Burst memory	Page memory	Async. memory
0	3	1	1
1	3	1	1
2	3	3	3
3	3	4	4
4	5	5	5
5	6	6	6
n	n+1	n+1	n+1

6.3.2.4 Write precharge (WR_PRECHARGE) cycles operation

This bit field allows for the insertion of extra cycles between a write and a read access to the same chip-select. These cycles are inserted only after a write access to any memory. The COSMORAM memories typically require these cycles and refer to them as write recovery cycles.

NOTE To ensure that write-recovery timing is met, it is important to program the PRECHARGE cycles to handle back-to-back burst writes.

The following table provides the number of cycles that a chip-select may be deasserted using the WR_PRECHARGE programmable value.

Table 6-8 WR_PRECHARGE programming vs. CS high cycles

Programmed value	CS high cycles		
	Burst memory	Page memory (async. write-page read)	Async. memory
0	1	1	1
1	1	1	1
2	3	3	3
3	4	4	4
4	5	5	5
5	6	6	6
n	n+1	n+1	n+1

6.3.2.5 PAGE_SIZE operation

The PAGE_SIZE bit field is used by the controller only if the chip-select is enabled for page or a burst-mode operation. There are two important reasons for setting this bit field correctly:

1. For page or burst memories, any burst access crossing the boundary defined in the PAGE_SIZE bit field will insert extra INIT_LATENCY cycles and begin a new access. However, if a burst is broken, a new access will begin without deasserting the corresponding chip-select.
2. If the memory device has a requirement that an explicit burst-terminate command needs to be issued, then the PRECHARGE_CYC bit field must be programmed. This will ensure that extra cycles are inserted when crossing the page boundary by deasserting the chip-select and restarting the access.

Note that the chip-select is deasserted for a minimum of one cycle when an access crosses a page boundary as defined by PAGE_SIZE.

The PAGE_SIZE bit field must be programmed to a value of 0x00 when WRAP8_MODE is set to 1 for chip-selects whose BURST_xx_ENA is set to 1. This mode automatically assumes a PAGE_SIZE = 16 hwords for 16-bit memories.

Page-read operation

Page-memory devices typically require a long initial access time for completing the first access in a page. Subsequent accesses within the page are much faster. Most of the memories that support page-mode operation usually only support a page read operation.

6.3.2.6 Synchronous burst-memory operation

Burst memories are like page memories: they provide faster access times for accesses within a page. However, burst memories use a clock to begin their accesses (page memories detect an address change to begin their access) and this enables them to operate at much higher frequencies. The following pins are used to support burst-mode memories, in addition to the pin for async/page memories:

- EBI1_ADV_N/LCD_RS (EBI2_ADV_N)
  - This pin indicates to the burst-memory that the address is valid and must be captured by the memory.
- EBI1_DQS_1 (EBI1_WAIT_N)/LCD_EN (EBI2_WAIT_N)
  - This pin is an input to the chip. The burst-memory device asserts this pin low if it wants to assert wait states for any particular access. This pin must be asserted one clock cycle prior to the actual access.
- EBI1_M_CLK/EBI2_M_CLK
  - This provides the clock to the burst-memory device.

The controller uses the INIT_LATENCY settings for the burst-memory device. Note that the EBI1_WAIT_N/EBI2_WAIT_N pin is used to delay a particular access. However, programming an INIT_LATENCY = 3 implies that the first access to a burst memory will take a minimum of INIT_LATENCY cycles, regardless of the value on EBI1_WAIT_N. The BURST controller checks for the wait on the pin only after the initial INIT_LATENCY counters have expired.

6.3.2.7 Burst-memory initialization

All burst-mode devices require the following to be configured for memory during software initialization:

- The rising edge of CLK has to be the active edge for memory.
- Synchronous programming (write) to burst flash is not supported. All writes to burst flash must be asynchronous. This requires that BURST_WR_ENA be set to 0 for all burst flash devices.
- EBI1_RDY/EBI1_WAIT_N or EBI2_RDY/EBI2_WAIT_N must be configured to arrive one clock earlier than the data.
- Only 0 wait states of subsequent operation can be supported. Programming WAIT_RD/WAIT_WR = 0 **is highly recommended**.
- The burst memory **must** be configured when the cache is **disabled**.
- The burst memory **must not** be used for execution/reading while being configured. For example, if CS0 is currently a burst flash operating in asynchronous mode, then the microprocessor has to be executing out of a different chip-select first. The burst flash can then be configured to switch to the synchronous mode as described in the example below.

Example 1:

For a 50-MHz HCLK, 70/13.5 ns AMD burst flash device (AM29BDS640G), the burst-mode configuration register must be programmed first. Some of the parameters such as WRAP operation must be configured based on software requirements. The AMD burst flash requires the following sequence of writes to set the burst-mode configuration register:

1. Write to the address = 0x555 with data = 0xAA. Because the external memory controller performs an address shift based on the data size of the memory device, the ARM microprocessor must perform write operations using byte addresses. This implies that the ARM microprocessor must actually write to ADDR = 0xAAA.
2. Write to the address = 0x2AA with data = 0x55. This corresponds to ADDR = 0x554.
3. Write to the address = 0x(CR)555 with data = C0. CR corresponds to the configuration register bits (A19-A12) of the burst-memory device. Based on the settings below, write to ADDR = 0x46AAA.
   - A19 = 0, synchronous read-enabled
   - A18 = 0, EBI1_RDY active one-clock cycle before data
   - A17 = 1, burst starts and data are output on the rising edge of CLK
   - [A16:A15] = 00, continuous burst. If the WRAP8_MODE feature is used, then set to [10]. This is described in more detail in Section 6.3.2.8.
   - [A14:A12] = 011, configures the wait state settings for the initial access. Because **only** the handshake option is supported, set five clock cycles for 50 MHz operation as specified in the datasheet.

In addition to configuring burst flash, the configuration registers in the burst controller must also be programmed. Ensure that the following conditions are satisfied.

- INIT_LATENCY_RD + WAIT_RD + 1 > 70 ns
- WAIT_RD + 1 > 13.5 ns

WAIT_RD = 0 implies that the clock can be as fast as 1/13.5 ns = 74 MHz, assuming the burst flash can support it; the AMD part mentioned above supports a maximum speed of 54 MHz only. The INIT_LATENCY wait states must now be programmed correctly.The AMD datasheet specifies that the EBI1_RDY pin is controlled by the burst flash, indicating when the data will actually be available. Because handshake mode is supported, the INIT_LATENCY cycles need to be programmed to a value ≥ 1. Follow these steps to configure CS0 to be a burst flash operating in synchronous mode:

1. Set INIT_LATENCY_RD = 2, and WAIT_RD = 0 for correct operation by writing to the EBI1_CS1_CFG0/EBI2_CS2_CFG0 register based on the chip-select of interest. Because the HANDSHAKING mode of the burst memory is used, the EBI1_RDY/EBI1_WAIT_N or EBI2_RDY/EBI2_WAIT_N will go high to indicate valid data.

2. Next, the AMD burst flash must be configured as described in the datasheet to meet the conditions listed previously. Note that because the memory is 16-bit wide, all accesses to the memory are 16-bit (half word) accesses. An example sequence of operations is as follows:

```
LDR  R3, =EBI1_CS0_BASE
LDR  R4, =0xAAA
LDR  R5, =0xAA
STRH R5, [R3,R4]

LDR  R3, =EBI1_CS0_BASE
LDR  R4, =0x554
LDR  R5, =0x55
STRH R5, [R3,R4]

LDR  R3, =EBI1_CS0_BASE
LDR  R4, =0x46AAA
LDR  R5, =0xC0
STRH R5, [R3,R4]
```

3. Set the CS0_BURST_RD_ENA bit to 1 in the EBI1_CS0_CFG1 register.

Example 2:

For a 50-MHz HCLK and a 54-MHz Micron synchronous burst flash MT28F322D20 (80 ns/17 ns), program the read configuration register (RCR) first. This Micron part requires the following sequence of writes to put it into burst mode.

1. Write to the address = RCD (read configuration data) with data = 0x60. The RCD value based on the parameters listed below implies that 0x21CF must be presented on the address bus to the burst memory.
   - A15 = 0, synchronous burst access mode
   - A14 = 0, reserved
   - [A13:A11] = 100, latency counter = 4 for 48-MHz operation
   - A10 = 0, reserved
   - A9 = 0, hold data out for one clock only
   - A8 = 1, EBI1_WAIT_N is asserted one clock cycle before data is available
   - A7 = 1, data is always addressed linearly
   - A6 = 1, rising edge of the clock is the active edge
   - [A5:A4] = 00, reserved
   - A3 = 1, burst does not wrap within the burst length
   - [A2:A0] = 111, continuous burst. The Micron burst memory does not support a burst length of 16 words, so the WRAP8_MODE feature **cannot** be used.
2. Write to the address = RCD (in this case: 0x439E) with data = 0x03.

Now, configure the INIT_LATENCY_RD and WAIT_RD parameters for burst memory.

- Because the clock period is 20 ns and the burst flash has an access time of 17 ns, there is an approximately 3-ns margin that should account for any external board delays. A value of WAIT_RD = 0 can thus be programmed. This is **only** an example. The system designer must do the analysis to ensure that the timing can be met from the AC characterization data in the device specification document.
- For the 54-MHz part, the INIT_LATENCY_RD = latency configuration code + 1 = 5

Follow the steps in this example to configure CS0 to be a burst flash operating in synchronous mode:

1. Set INIT_LATENCY_RD = 5, and WAIT_RD = 0 for correct operation by writing to the EBI1_CS1_CFG0 register.
2. Next, the Micron burst flash must be configured as described in the datasheet to meet the conditions listed previously. Note that because the memory is 16 bits wide, all accesses to the memory are 16-bit (half-word) accesses. The sequence of operations is as follows:

```
LDR  R3, =EBI1_CS0_BASE
LDR  R4, =0x439E
LDR  R5, =0x60
STRH R5, [R3,R4]

LDR  R3, =EBI1_CS0_BASE
LDR  R4, =0x439E
LDR  R5, =0x03
STRH R5, [R3,R4]
```

3. Now, set the CS0_BURST_RD_ENA bit to 1 in the EBI1_CS0_CFG1 register.

6.3.2.8 WRAP8 READ/INCR burst optimization

Setting the WRAP8_RD_MODE provides the external memory controller with the information that the memory is in WRAP8 mode for read operations. However, write accesses are assumed to be INCR accesses.

In the WRAP8_RD mode of operation, the memory will keep wrapping at the 8-word (16-hword) boundaries. This ensures that only the first access in a cache line-fill will be as short as possible and all remaining locations of that line can be single-cycle accesses. If the WRAP8_RD_MODE is not set, the memory will keep supplying the data continuously until the controller terminates the burst. The external memory controller uses the EBI1_RDY (EBI1_WAIT_N) signal provided by the burst memory to insert extra wait cycles when required.

Set the WRAP8_MODE bit for a particular chip-select when the following conditions are satisfied:

1. The chip-select is hooked up to a burst device.
2. The burst device supports a page size of 16 hwords or greater.
3. The burst device supports wrap within burst length.

The following changes to the BCR are required to configure the burst memory into WRAP8 mode (example 2).

1. Enable wrap within burst length; A3=0
2. Set burst length = 16 hwords; A2:A0 = 011

NOTE Although setting this bit to 1 optimizes accesses for the port with ARM as a master, the other masters on PORT0 and PORT1 do not perform WRAP8 operations and will suffer a hit in their performance. Therefore, we recommend setting this bit to 1 only after analyzing the system.

Step 3 of the initialization sequence described in the previous section must be modified as follows to support WRAP8 optimization. The WRAP8_MODE bit of EBI1_CS1_CFG1 must also be set to 1.

```
LDR  R3, =EBI1_CS1_BASE
LDR  R4, =0xAAA
LDR  R5, =0xAA
STRH R5, [R3,R4]

LDR  R3, =EBI1_CS1_BASE
LDR  R4, =0x554
LDR  R5, =0x55
STRH R5, [R3,R4]

LDR  R3, =EBI1_CS1_BASE
LDR  R4, =0x64AAA ; <----Configure AMD part for WRAP8 operation
LDR  R5, =0xC0
STRH R5, [R3,R4]
```

6.3.2.9 Burst-memory program operation

The program and erase operation of the burst memory **must be performed in asynchronous mode**. For example, the AMD burst flash also supports synchronous program operation. However, the QSC62x0 core will not support it. Software will have to ensure that CSx_BURST_WR_ENA is set to 0.

Other requirements for AMD program/erase operations:

- Set PWRSAVE_MODE to 1

  This ensures that the EBI1_M_CLK/EBI2_M_CLK pin going to the burst memory will be kept toggling only during read operations to the AMD memory after it is configured for burst operation.

6.3.2.10 Precharge cycles operation

The precharge cycles, as the name implies, are used by any pseudo-static SRAM to ensure that the precharge time is met. The important difference from recovery cycles is that precharge cycles are always applied to the same chip-select.

These cause extra cycles to be inserted between consecutive read or write access to the same chip-select when crossing page boundaries. The precharge cycles apply to chip-selects configured in either burst (BURST_xx_ENA = 1) or page mode (PAGE_xx_ENA = 1) or asynchronous mode.

The controller inserts a minimum of one precharge cycle for all precharge scenarios.The following are the three scenarios:

- Read followed by another read access to the same chip-select.
- Write followed by another write access to the same chip-select.
- A page boundary crossing during an access to a burst/page memory.

Note that precharge cycles are not inserted between bus-sized accesses.

Table 6-9 lists the number of precharge cycles inserted for a read-read scenario.

Table 6-9 PRECHARGE_CYC programming in a read-read scenario

Programmed value	CS high cycles		
	Burst memory	Page memory	Async. memory
0	3	1	1
1	3	1	1/2 [1]
2	3	3	3
3	3	4	4
4	5	5	5
5	6	6	6
n	n+1	n+1	n+1

[1] The chip-select is high for two cycles between accesses that belong to an AHB burst beat. Expect that the chip will stay high for one cycle between single accesses. It is recommended that a programmable value suitable for all AHB accesses be chosen.

Table 6-10 lists the number of precharge cycles inserted for a write-write scenario.
Table 6-11 lists the number of precharge cycles inserted for a page boundary scenario.

Table 6-10 PRECHARGE_CYC programming in a write-write scenario

Programmed value	CS high cycles		
	Burst memory	Page memory	Async. memory
0	1	-	1
1	1	-	1/2 [1]
2	3	-	3
3	4	-	4
4	5	-	5
5	6	-	6
n	n+1	n+1	n+1

[1] The chip-select is high for two cycles between accesses that belong to an AHB burst beat. Expect that the chip will stay high for one cycle between single accesses. It is recommended that a programmable value suitable for all AHB accesses be chosen.

Table 6-11 PRECHARGE_CYC programming in a page-boundary scenario

Programmed value	CS high cycles			
	Burst memory		Page memory	
	Read access	Write access	Read access	Write access
0	1	1	1	–
1	2	2	2	–
2	3	3	3	–
3	4	4	4	–
4	5	5	5	–
5	6	6	6	–
n	n+1	n+1	n+1	

6.3.2.10.1 WRAP8_RD_MODE = 1

For burst or page-read accesses in this mode, always assume the size of the page is configured to:

- 16 hwords for 16-bit PSRAM
- 8 words for 32 -bit PSRAM

Write accesses continue to be treated as linear (continuous) mode writes. If you need to break write accesses across a page boundary, the PAGE_SIZE bit field needs to be programmed.

Whenever an access crosses 32 bytes (for both x16 or x32 PSRAMs), a precharge cycle is inserted regardless of the PAGE_SIZE bit field. See Section 6.3.2.8 for more details.

6.3.2.10.2 WRAP8_RD_MODE = 0

In this mode, the memory is configured to operate in the linear (continuous) addressing mode. So, if the master (ARM, DSP, etc.) does a linear burst, such as INCR4, INCR8, etc., the accesses will all be continuous and treated as one burst. If the memory device has a restriction on crossing a page boundary, then the size of that boundary must be programmed in the PAGE_SIZE bit field. The precharge cycles will then be inserted whenever a burst access crosses this boundary.

For example, if the size of the page is configured to be 16 half-words (sixteen 16-bit words), then a burst crossing this boundary will be broken up by deasserting the chip-select and a new burst will be issued with the address corresponding to the new page.

If the ARM does a WRAP8 operation, then the burst will be broken up into two bursts at the wraparound boundary (832-bit word boundary). Note that if the PRECHARGE_CYC bit field is set to 0, then the CS_N would be deasserted for one cycle and CS_N and ADV_N strobe are asserted for the new wraparound address.

When the memory is configured for page or burst-mode read or write operation, the precharge cycles are inserted only when a multiword transaction addresses locations on both sides of a page boundary.

- For a read->write or write->read access to the same chip-select, there is one arbitration clock cycle inserted between accesses that should provide the necessary precharge time.
- For a read->write access, recovery cycles have to be programmed to ensure that enough cycles are inserted between accesses.
- For a write->read access to the same chip-select configured to be in asynchronous mode, and HOLD_WR=1, two precharge cycles (1 arbitration cycle + 1 hold) will be inserted. More cycles between the write->read will require increasing the value for HOLD_WR. For a write->read access to the same chip-select configured to be in burst mode, the WR_PRECHARGE bit field must be programmed to ensure sufficient cycles between write->read.

Example:

If PAGE_SIZE is programmed, the chip-select is deasserted for a minimum of one clock cycle.

NOTE 1 Precharge cycles also get inserted for accesses to burst memory devices when going across page boundaries. If this is not required, ensure that the PAGE_SIZE value for that chip-select is set to 0x0.

NOTE 2 Do not program IGN_WAIT_RD/IGN_WAIT_WR = 1 and PAGE_SIZE = 0x0 simultaneously. This would imply that the controller would not insert precharge cycles and would also ignore the wait_n pin.

6.3.2.11 ADDR_HOLD_ENA

This bit is used to support MUXed-address/data interfaces to the external memory. When this bit is 0, the address is driven only for the duration of the ADV_N active period. In case greater hold times are needed, setting ADDR_HOLD_ENA to 1 implies that the address will be held for one extra cycle after ADV_N rise. For non-MUXed address/data interfaces, there are no issues with address hold, because the address is driven all the time.

NOTE 1 For burst memories, the controller asserts the address either for a single cycle or two cycles based on the ADDR_HOLD_ENA setting. For the remaining duration of the burst access, the controller drives 0x0 on the address bus.

NOTE 2 We recommend programming ADDR_HOLD_ENA=1 for dual-EBI mode.

NOTE 3 If ADDR_HOLD_ENA = 1, ADV_OE_RECOVERY must be set to at least 1 to avoid bus contention. In addition, INIT_LATENCY_RD/INIT_LATENCY_WR must be programmed to at least two cycles.

6.3.2.12 ADV_OE_RECOVERY

This bit field assures OE_N with respect to ADV_N. A value of 0 implies that ADV_N rise and OE_N fall will occur on the same clock cycle. A value of 1 ensures that OE_N will be asserted one clock cycle after ADV_N deasserts.

This bit field must be programmed appropriately to avoid bus contention after the chip stops driving the address bus. This address bus is reused as the data bus when supporting MUXed-address/data memories.

NOTE INIT_LATENCY_RD must always be programmed greater than the ADV_OE_RECOVERY value.

6.3.2.13 WE_TIMING

Setting this bit to 1 ensures that WE_N remains active for the duration of the burst access. The controller by default only supports a single clock pulse on WE_N when performing burst-write operations. The COSMORAM memories originally did not support the pulse WE_N timing. The WE_TIMING is mainly intended for COSMORAM memories.

In addition, some burst NOR flash devices also support a synchronous write timing operation, which requires WE_N to be asserted for the entire access. In such cases, WE_TIMING must be set to (1).

6.3.2.14 IGN_WAIT_FOR_WR

The purpose of this bit is to ignore the EBI1_WAIT_N/EBI2_WAIT_N pin during burst-write operation.

This bit is useful for burst PSRAM memories that do not drive the WAIT_N pin during a burst-write operation. The controller by default always expects the burst memory to drive the WAIT_N even for a write access.

So, this bit must be set to 1 for memories where the WAIT_N pin is HI-Z during a burst-write operation. It is important to program the INIT_LATENCY_WR value to match the write-latency corresponding to the memory device.

NOTE If this bit is programmed to 1, ensure that PAGE_SIZE is programmed to an appropriate value.

6.3.2.15 IGN_WAIT_FOR_RD

The purpose of this bit is to ignore the EBI1_WAIT_N/EBI2_WAIT_N pin during burst-read operations.

This bit is useful for burst PSRAM memories that do not drive or incorrectly drive the WAIT_N pin during a burst-read operation. The controller by default always expects the burst memory to drive the WAIT_N for a read access.

So this bit must be set to 1 for memories where the WAIT_N pin is HI-Z during a burst-read operation. It is important to program the INIT_LATENCY_RD value to match the read-latency corresponding to the memory device.

NOTE If this bit is programmed to 1, ensure that PAGE_SIZE is programmed to an appropriate value.

6.3.2.16 PWRSAVE_MODE feature

When burst-flash is used, preventing the EBI1_M_CLK/EBI2_M_CLK pin from toggling when no accesses are being done to the burst memory can save power. This is achieved by setting the PWRSAVE_MODE to high, which ensures that the respective memory clock (burst clock) coming out of EBI1 or EBI2 will be toggling only during a valid burst access.

Once the ARM begins execution out of the burst flash (in synchronous mode), it can always go into sleep/low-power mode and upon wake-up can continue execution out of the burst flash. It is not necessary to switch to asynchronous mode prior to going into sleep.

NOTE On reset, the memory clock is always toggling. Also, note that EBI1_M_CLK and EBI2_M_CLK can be independently controlled using register bits in EBI1_CFG and EBI2_CFG, respectively.

6.3.2.17 BYTE_DEVICE_ENA

This bit needs to be set to 1 by software when an 8-bit memory is connected to the external interface. This setting will help the controller take care of BYTE, HWORD and WORD access to an 8-bit device.

6.3.2.18 x32_DEVICE_ENA

This bit needs to be set to 1 by software when a 32-bit device is connected to the external memory. This bit needs to be used along with BURST_RD_ENA/BURST_WR_ENA in the case of a 32-bit burst memory.

6.3.2.19 Low-power mode control using UXMC_PSRAM_CRE_CFG

The UXMC_PSRAM_CRE_CFG register can be used to control the pin through software.

This is very useful for connecting to PSRAMs that support a deep powerdown or deep-sleep mode using an extra pin. For example:

- cs pin on the Toshiba PSRAM
- zz pin on the Cypress PSRAM
- cre pin on the Micron PSRAM
- mode pin on the NEC Mobile RAM
- zz pin on the Samsung UtRAM

This pin can also be used for any PSRAMs that operate in the synchronous mode and to issue commands to the RAM to program the mode register.

6.3.2.20 LCD_INTERFACE_TYPE

This register bit needs to be programmed based on the LCD interface connected to the LCD_CS_N pin of the chip. The controller supports both Intel (8080) and Motorola (6800) interfaces. Set this bit to 1 when using a Motorola interface LCD device.

NOTE 1 Any of the regular EBI2 chip-selects may be used as a second LCD chip-select. However, these may be used only as an Intel (8080) interface.

NOTE 2 Although the external memory controller itself supports both memory-mapped and port-mapped LCD devices, the EBI interface supports only port-mapped devices.

NOTE 3 LCD_RS can be independently controlled regardless of the EBI2 chip-select used.

6.3.2.21 LCD_RECOVERY

This bit field needs to be programmed when recovery cycles are required between an LCD read access and an access to any other EBI2 chip-select. A maximum of 15 recovery cycles is supported. By default, the controller always inserts a recovery cycle. In effect, if this field is programmed to *r*, then *r+1* recovery cycles are inserted.

NOTE This field applies to both Intel and Motorola types of LCD interfaces.

6.3.2.22 LCD HOLD cycles operation

The LCD hold cycles operation is similar to the operation of hold cycles in an asynchronous interface. It is recommended that the LCD_HOLD_WR field be programmed to a minimum of one cycle for both the Intel and Motorola interfaces. As per the LCD_HOLD_RD field, a value of 0x0 would work for most of the LCD devices. Please refer to the device datasheets to determine the exact programming values.

If a value of *h* is programmed, the controller inserts *h* hold cycles.

NOTE This field applies to both Intel and Motorola types of LCD interface.

6.3.2.23 LCD WAIT cycles operation

The LCD wait cycles operation is no different from that of its asynchronous counterpart. LCD_WAIT_WR determines the number of cycles during which WE_N is asserted low for a write access to a LCD device of the Intel interface type. LCD_WAIT_RD determines the number of cycles OE_N that will be asserted low for a read access to a LCD device of the Intel interface type.

Programming this field to *w* results in the respective signal being asserted for *w+1* cycles.

NOTE This field applies to the Intel type of LCD interface only.

6.3.2.24 LCD_RS support

Table 6-12 lists the external-memory controller multiple programmable options to drive the LCD_RS pin.

Table 6-12 Programmable options to drive LCD_RS

LCD_RS_MODE	LCD_RS_VAL	Comments
00	0 or 1	LCD_RS_VAL drives the LCD_RS pin.
01	N/A	Bit 20 of AHB address for the current LCD access drives the LCD_RS pin.
10	N/A	Bit 19 of AHB address for the current LCD access drives the LCD_RS pin.
11	N/A	Bit 18 of AHB address for the current LCD access drives the LCD_RS pin.

NOTE LCD_RS can be independently controlled regardless of the EBI2 chip-select used.

6.3.2.25 LCD_EN support

LCD_EN pin is driven when the LCD_INTERFACE_TYPE = 1, the Motorola-type LCD interface. The controller supports LCD_E_SETUP and LCD_E_HIGH cycles. Read and write accesses can be independently controlled using the appropriate register bit fields.

LCD_E_SETUP controls the number of cycles during which LCD_EN is driven low. If LCD_E_SETUP is programmed *es*, LCD_EN will be driven low for *es* cycles.

NOTE 1 The minimum programmable value for LCD_E_SETUP_READ and LCD_E_SETUP_WRITE is 1.

NOTE 2 LCD_E_HIGH controls the number of cycles during which LCD_EN is driven high. If LCD_E_HIGH is programmed *eh*, LCD_EN will be driven high for *eh + 1* cycles.

NOTE 3 LCD_CS is asserted for *es + eh + h + 1* cycles.

6.3.2.26 LCD_BYTE_DEVICE_ENA

Program this bit when an 8-bit LCD device is connected to the LCD interface. The external memory controller supports bus-sized accesses for both 8-bit and 16-bit devices.

6.3.2.27 LCD_18_BIT_DEVICE_ENA

Program this bit to 1 when an 18-bit LCD device is connected to the LCD chip-select.

7 Air Interfaces

The QSC62x0 device performs the signal processing necessary to support WCDMA and GSM air interfaces; adding the RGR6240 IC to either QSC device enables GPS position location. These air-interface support functions are entirely embedded within the device — there are no air interface-specific I/Os to be discussed in this chapter. The airlink-related I/Os necessary to control analog/RF and PM functions are covered in Chapter 3 (RF Signal Paths and LO Circuits), Chapter 12 (Baseband - Analog/RF-PM Interfaces), and Chapter 16 (PM Interfaces and MPPs).

The supported air-interface standards and features include:

- UMTS/WCDMA/GSM/GPRS/EDGE Specification Release '99 (3GPP R99)
- GSM/GPRS/EDGE Specification Release 4 (3GPP R4)
- UMTS/WCDMA Specification Release 5 (3GPP R5, QSC6270 only)
  - HSDPA and equalizer; 3.6 Mbps
- Enhanced GPS position location using gpsOne (with RGR6240 IC)
  - Integrated gpsOne functionality, featuring enhancements by SnapTrack®, Inc., to enable a wide variety of location-based services and applications, including points of interest, personal navigation, and friend finder
  - Simultaneous-GPS (processes GPS using dedicated circuitry while voice and/or data signals continue to be processed separately)
  - 1024x searcher, direct facility termination (DFT) accelerator, off-chip RAM for measured data storage

7.1 WCDMA air interfaces

The QSC62x0 UMTS analog and digital baseband subsystem performs the data conversions and signal processing necessary to maintain the WCDMA air interface between the handset and the base station (and WCDMA network). Supported modes include WCDMA R99 and HSDPA.

7.1.1 WCDMA R99

QSC6240 and QSC6270 WCDMA FDD release 99 support includes the following features:

- All modes and data rates for WCDMA frequency division duplex (FDD), with the following restrictions:
  - The downlink supports the following specifications:
    - Up to four physical channels, including the broadcast channel (BCH), if present
    - Up to three dedicated physical channels (DPCHs)
    - Spreading factor (SF) range support from 4 to 256
    - Transmit diversity modes: space-time transmit diversity (STTD), time-switched transmit diversity (TSTD), closed-loop feedback transmit diversity (CLTD)
  - The uplink supports the following specifications:
    - One physical channel, eight TrCH, and 16 TrBks starting at any frame boundary
    - 384 kbps maximum data rate
    - Full SF range from 4 to 256
- SMS (CS and PS)
- PS data rate: 384 kbps DL/384 kbps UL
- CS data rate: 64 kbps DL/64 kbps UL
- AMR (all rates)

7.1.2 HSDPA (QSC6270 only)

The QSC6270 device also supports HSDPA release 5/6, including:

- HS-DSCH (HS-SCCH, HS-PDSCH, and HS-DPCCH) and the R99 transport channels as defined in the 3GPP specifications
- A maximum of four simultaneous HS-SCCH channels as defined in the 3GPP specifications
- A maximum of 10 HS-PDSCH channels and both QPSK and 16-QAM modulation; UE categories 5 and 6
- CQI and ACK/NACK on the HS-DPCCH channel as defined in the 3GPP specifications
- All incremental redundancy versions for HARQ as defined in the 3GPP specifications
- Switching between HS-PDSCH and DPCH channel resources as directed by the network
- Power class 3 or power class 4 as defined in the 3GPP R5 specifications

- Network activation of compressed mode by SF/2 or HLS on the DPCH for conducting interfrequency or inter-RAT measurements when the HS-DSCH is active
- STTD on both associated DPCH and HS-DSCH simultaneously
- CLTD mode 1 on the DPCH when the HS-PDSCH is active
- STTD on HS-SCCH when either STTD or CLTD mode 1 are configured on the associated DPCH
- TFC selection limitation on the UL factoring in the transmissions on the HS-DPCCH as required in TS 25.133

7.2 GSM air interfaces

The QSC62x0 GSM analog and digital baseband subsystem performs the data conversions and signal processing necessary to maintain the GSM air interface between the handset and the network. Supported modes include GSM R99, GPRS, and EDGE.

7.2.1 GSM R99

The QSC62x0 device supports the following GSM modes and data rates while conforming to release 99 specifications:

- Voice features
  - FR
  - EFR
  - AMR
  - HR
  - A5/1, A5/2, and A5/3 ciphering
- Circuit-switched data features
  - 9.6 k
  - 14.4 k
  - Fax
  - Transparent and nontransparent modes for CS data and fax
  - No subrates supported

7.2.2 GPRS

- Packet-switched data (GPRS)
  - DTM (simple class A) operation
  - Multislot class 12 data services
  - CS schemes: CS1, CS2, CS3, and CS4
  - GEA1, GEA2, and GEA3 ciphering
- Maximum of four Rx timeslots per frame

7.2.3 EDGE

- EDGE E2 power class for 8 PSK
- DTM (simple class A), multislot class 12
- Downlink coding schemes: CS1 - CS4, MSC1 - MCS9
- Uplink coding schemes: CS1 - CS4, MSC1 - MCS9
- BEP reporting
- SRB loopback and test mode B
- 8-bit, 11-bit RACH
- PBCCH support
- 1-phase/2-phase access procedures
- Link adaptation and IR
- NACC, extended UL TBF

7.3 GPS air interface

gpsOne technology merges GPS satellite and network information to provide a high-availability solution that offers industry-leading accuracy and performance. This solution performs well even in very challenging environmental conditions where conventional GPS receivers fail. It provides a platform to enable wireless operators to address both location-based services and emergency mandates, such as the United States FCC E911 mandate.

When a request for position location is issued in mobile-assisted mode, the available network information is provided to the location server (e.g., cell-ID), and assistance is requested from the location server. The location server sends the assistance information to the handset. The handset/mobile unit measures the GPS observables and provides the GPS measurements along with available network data (that is appropriate for the given air interface technology) to the location server. The location server then calculates the position location and returns the results to the requesting entity.

In mobile-based mode, the assistance data provided by the location server encompasses not only the information required to assist the handset in measuring the satellite signals,

but also the information required to calculate the handset's position. Therefore, rather than provide the GPS measurements and available network data back to the location server, the mobile calculates the location on the handset and passes the results to the requesting entity.

In standalone (autonomous) mode, the handset demodulates the data directly from the GPS satellites. This mode has some reduced cold-start sensitivity and a longer time to first fix as compared to the assisted modes. However, it requires no server interaction and works in out-of-network coverage situations.

This combination of GPS measurements and available network information provides:

- A high-sensitivity solution that works in all terrains: indoor, outdoor, urban, and rural
- High availability that is enabled by using both satellite and network information

Therefore, while network solutions typically perform poorly in rural areas and areas of poor cell geometry/density, and while unassisted, GPS-only solutions typically perform poorly indoors, the QUALCOMM gpsOne solution provides optimal time-to-fix, accuracy, sensitivity, availability, and reduced network utilization in both of these environments, depending on the given condition.

The gpsOne solution in assisted modes provides a cold-start GPS sensitivity improvement of approximately 20 to 30 dB over unassisted, conventional GPS receivers.

Compared to network solutions that require equipment at each cell site, the gpsOne solution integrates a complete GPS receiver in the RGR6240 and QSC62x0 chipset combination. This means that each handset is capable of position location without requiring expensive cell site equipment. This solution not only can be used to help operators address the FCC E911 mandate in the United States (and mandates planned for other countries), but also provides a ubiquitous platform for location-based applications. The gpsOne solution enables consumer-priced, position-capable handsets for location-based services worldwide.

8 Multimedia Blocks and Applications

Multimedia topics are discussed in this chapter, including:

- Camera interface and video front-end
- Mobile display processor
- Additional multimedia support: video, audio, graphics, and messaging

8.1 Camera interface and video front-end

The camera interface (CAMIF) connects the QSC62x0 device directly to a camera sensor. Typical applications include Qcamera, Qcamcorder, and Qvideophone. The CAMIF delivers the raw 10-bit Bayer pattern data (preferred) to the video front-end (VFE) that performs the required image processing (RGB-triplet generation, color-space conversion, auto-white balance, auto exposure, gamma correction, etc.) and prepares the image for capture or transmission. The QSC device also supports a YUV 4:2:2 input from the sensor (8-bit, 2:1 MUX YUV or CCIR656 YUV).

The video capabilities of the two QSC devices varies slightly, with the QSC6270 providing higher performance than the QSC6240 device — greater resolution, higher capture and streaming rates, etc.

8.1.1 CAMIF and VFE features

The camera interface supports CMOS and CCD sensors, and Bayer and YUV data. The resolution is up to 2.0 MP for the QSC6240 device, and up to 3.0 MP for the QSC6270 (Bayer only). The combination of CAMIF and VFE delivers a wide variety of pixel manipulation, camera modes, image effects, and postprocessing techniques:

- High resolution: QSC6240 = 2.0 MP; QSC6270 = 3.0 MP
- JPEG and MPEG-4 encoding
- 30 fps QCIF viewfinder

- Pixel manipulation
  - Noise suppression
  - Edge enhancement
  - Blemish/pixel bad detection and correction
  - Color interpolation, correction, and space conversion
  - HW-based independent white balance (auto, daylight, cloudy, indoor/flicker)
  - 10-bit gamma correction (programmable lookup table)
  - Sharpness control
  - Auto-exposure and auto-focus
  - Hardware-based Bayer roll-off correction support
- Multiple operating modes
  - Continuous digital zoom
  - Multiple shot (regular exposure or bracketing – with adequate memory support)
  - Macro mode
  - Flash support
- Effects
  - Image effects (sepia, B/W, blackboard, whiteboard)
  - Still image edit (rotation, scaling, H/V flip, interpolation)
- Image post-processing
  - Hand jitter reduction/anti-shake (single frame)
  - Luma adaptation
  - Chroma suppression

8.1.2 CAMIF connections

An example camera interface is shown in Figure 8-1, and all CAMIF connections are listed in Table 8-1.

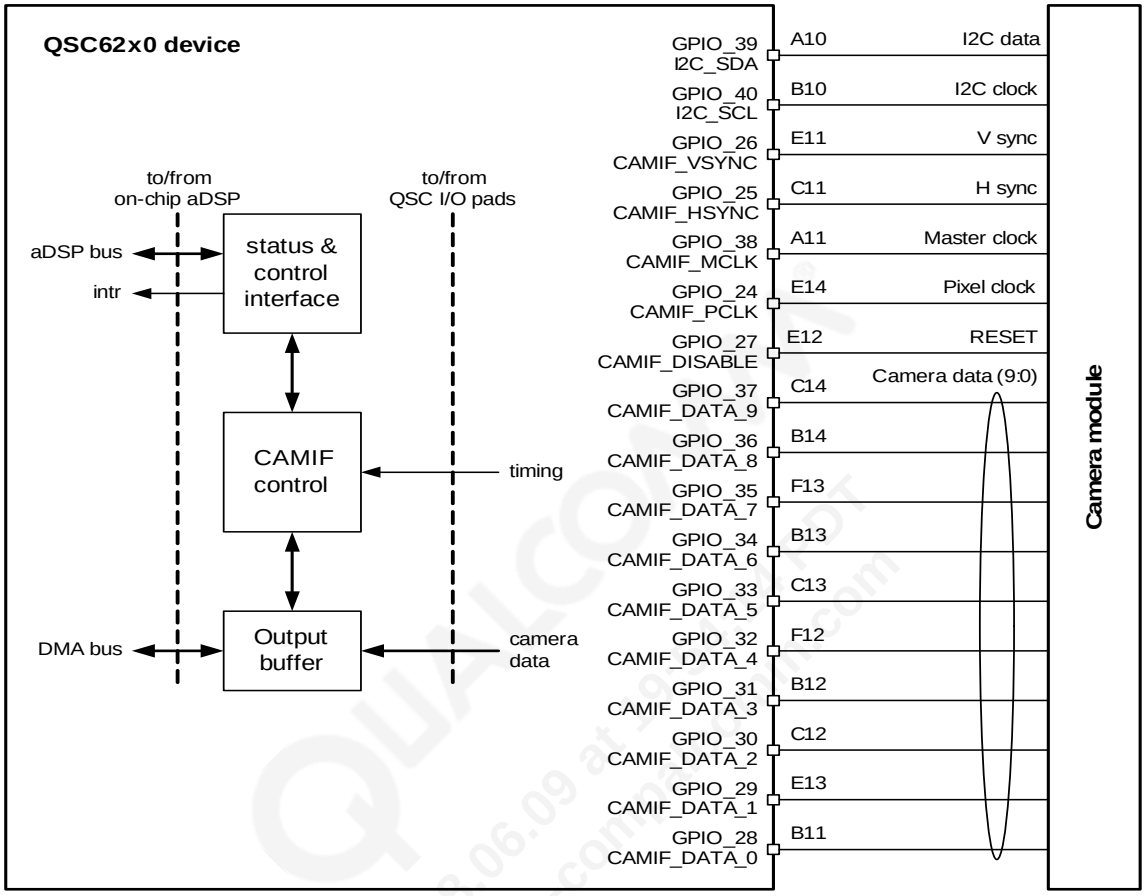


Figure 8-1 Camera interface connections

Table 8-1 Camera interface connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
CAMIF_DATA_9 (GPIO_37)	C14	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 9
CAMIF_DATA_8 (GPIO_36)	B14	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 8
CAMIF_DATA_7 (GPIO_35)	F13	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 7
CAMIF_DATA_6 (GPIO_34)	B13	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 6
CAMIF_DATA_5 (GPIO_33)	C13	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 5
CAMIF_DATA_4 (GPIO_32)	F12	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 4
CAMIF_DATA_3 (GPIO_31)	B12	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 3
CAMIF_DATA_2 (GPIO_30)	C12	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 2
CAMIF_DATA_1 (GPIO_29)	E13	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 1
CAMIF_DATA_0 (GPIO_28)	B11	P5 (1.8 V)	DI	–	Parallel camera interface pixel data bit 0
CAMIF_MCLK (GPIO_38)	A11	P5 (1.8 V)	DO	1-8 (1)	Master clock to the camera sensor
CAMIF_PCLK (GPIO_24)	E14	P5 (1.8 V)	DI	–	Pixel clock from the camera
CAMIF_HSYNC (GPIO_25)	C11	P5 (1.8 V)	DI	–	Horizontal synchronization from the camera
CAMIF_VSYNC (GPIO_26)	E11	P5 (1.8 V)	DI	–	Vertical synchronization from the camera
CAMIF_DISABLE (GPIO_27)	E12	P5 (1.8 V)	DO	1-8 (1)	Turns off the camera
I2C_SDA (GPIO_39)	A10	P5 (1.8 V)	B	1-8 (1)	I^2C data
I2C_SCL (GPIO_40)	B10	P5 (1.8 V)	B	1-8 (1)	I^2C clock

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

Each major component of the CAMIF is described more completely in the following paragraphs.

CAMIF_MCLK: master clock

This is used as the main clock into the camera module. In general, the sensor's MCLK is generated by the QSC device (CAMIF_MCLK) or through an external clock source (where the module is responsible for sourcing this clock). The output pixel clock, CAMIF_PCLK, which is normally generated by the camera module, is derived from this clock. The typical frequency of this clock varies depending on the requirements of the camera module.

CAMIF_DISABLE: optional sensor enable/reset input

This is the optional enable/reset line provided from the QSC device into the camera module. The signal can be active high or active low, depending upon the camera module requirements.

Camera data signals

CAMIF_DATA[9:2]: 8 MSBs of the output data bus from the camera module for a 10-bit interface, or the full 8 bits for an 8-bit interface.

CAMIF_DATA[1:0]: 2 LSBs of the output data bus from the camera module for a 10-bit interface; idle and not connected for an 8-bit interface.

CAMIF_PCLK: pixel clock

This is the output pixel clock from the camera module. The QSC device uses this clock to latch in the output data from the camera module. It is assumed that this clock is continuous, with a programmable function to gate the clock off as power-saving options. The PCLK can run as fast as 80 MHz.

Camera synchronization

CAMIF_HSYNC: Horizontal synchronization signal — this is the line synchronization signal output from the camera module. The dedicated hardware allows automatic tracking of this signal by the QSC62x0 device. See Section 8.1.5 for more details.

CAMIF_VSYNC: Vertical synchronization signal — this is the frame synchronization signal output from the camera module. The dedicated hardware allows automatic tracking of this signal by the QSC62x0 device. See Section 8.1.5 for more details.

I^2C control bus

I2C_SDA: Serial data line of the I^2C bus — the standard required pull-up resistor is placed on the QSC device side; a pull-up resistor is not required in the camera module.

I2C_SCL: Serial clock line of the I^2C bus — the standard required pull-up resistor is placed on the QSC device side; a pull-up resistor is not required in the camera module.

8.1.3 Clock requirements

The camera module is driven by a master clock input, provided either from the QSC device or an external oscillator. It is important to verify the jitter on the clock input to the camera does not violate the sensor specifications. In addition, since the PCLK input to the QSC device is derived from the MCLK, it is important to verify the PCLK quality, and verify that any jitter carried over from the MCLK does not violate the QSC specifications.

8.1.3.1 CAMIF_MCLK

If a QSC clock source is chosen, the CAMIF_MCLK output from GPIO_38 should be used. This dedicated clock is specifically designed to provide a clock input to sensors. It has several convenient features such as SW gating, which allows the user to turn the clock on and off cleanly, without glitches. Note that this timing constraint will limit the maximum output frequency. Since it is intended specifically for sensor use, and since the PCLK input to the QSC device may very well be derived from it, the CAMIF_MCLK will be characterized and tested along with the rest of the camera interface to ensure correct and reliable operation.

NOTE If the QSC device is selected as the camera's master clock source, use the CAMIF_MCLK output at GPIO_38.

The CAMIF_MCLK output signal can be derived from various clock sources that can then be further divided by a modulo divider or by an M/N:D counter. Designers must be aware of M/N counter limitations and test their desired clock output to verify it conforms to the sensor requirements. For minimal jitter and minimal duty cycle degradation, even integer dividers should be used. If other dividers are wanted, QCT will work with designers to evaluate different M/N values and the resulting clock quality.

The CAMIF_MCLK settings are configured in the **CAMCLK_PO_CLK_MD** and **CAMCLK_PO_CLK_NS**.

For more detailed information regarding these registers, please refer to the *QSC6240/QSC6270 QUALCOMM Single Chip Software Interface* document (80-VF846-2).

8.1.4 Frame synchronization capabilities

The data output on the 10-bit (or 8-bit) bus must be synchronized to a set of frame and line synchronization signals that are recognizable by the device. These synchronization signals are:

- HSYNC – Horizontal synchronization signal, indicating the start and end of a line of video
- VSYNC – Vertical synchronization signal, indicating the start and end of a frame of video

The usual mode of operation is active physical synchronization. This is where the QSC device receives the HSYNC and VSYNC signals as inputs from the camera module. It then uses those signals in the internal VFE to correctly parse and rebuild the frames.

The preferred type of data format is raw Bayer RGB data (though YCbCr 4:2:2 is also supported).

8.1.4.1 Frame synchronization timing

Figure 8-2 shows the relationship between the vertical and horizontal synchronization signals. The timing of the arrival of the first valid active video line with respect to VSYNC signal is programmable. All synchronization signals throughout this document are shown as active low. However, the QSC device can be programmed to accept both active high or active low pulses for HSYNC, VSYNC, and PCLK signals.

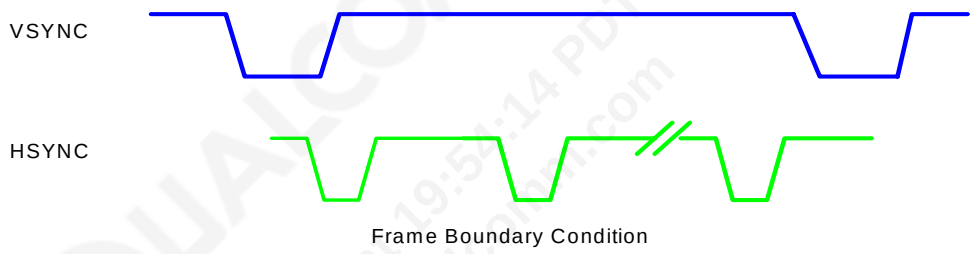


Figure 8-2 Horizontal and vertical synchronization signals

Figure 8-3 and Figure 8-4 indicate the start of frame conditions. In Figure 8-3, it is assumed that the camera module outputs its first active line immediately after the VSYNC pulse, where as Figure 8-4 shows the condition where a number of inactive lines can precede the first active line of the frame. The QSC device can interface with modules having either capability.

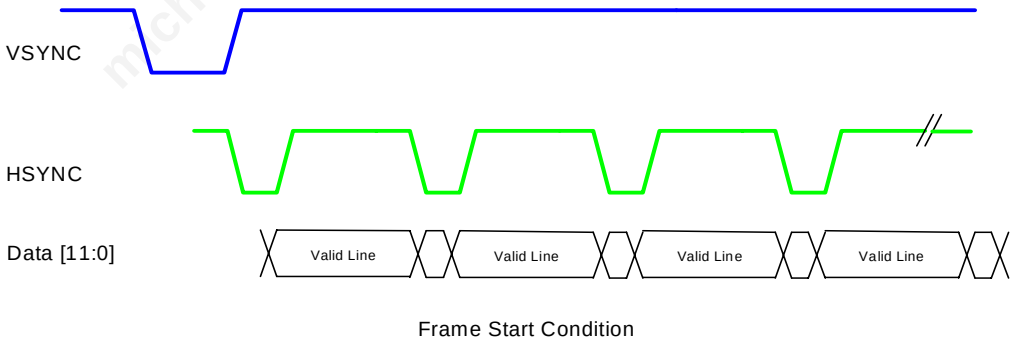


Figure 8-3 Frame start condition

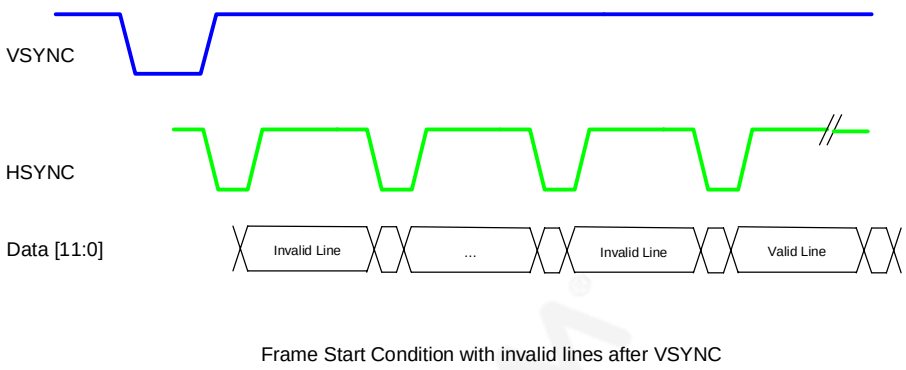


Figure 8-4 Frame start condition with invalid lines after VSYNC

The QSC device is also flexible in handling the different end-of-frame conditions, as indicated in Figure 8-3 and Figure 8-4. In Figure 8-3, it is assumed that the camera module outputs active lines up until immediately before the VSYNC pulse, with no invalid lines present during the active VSYNC period. Figure 8-4 shows the condition where a number of inactive lines can precede the VSYNC pulse. The QSC device can interface with modules having either capability.

8.1.5 Video front-end hardware

The QSC62x0 device features dedicated video front-end hardware that enables:

- PCLK rate of up to 80 MHz
- Maximum resolution of up to:
  - 2.0 MP for QSC6240 (regardless of input format)
  - 3.0 MP for QSC6270 (Bayer only)

NOTE The maximum snapshot FPS rate supported is determined by data type and resolution; it is hard-limited by the 80 MHz maximum data rate.

- Wide variety of camera sensors
  - The QSC device tracks the HSYNC and VSYNC inputs — a feature that is very designer-friendly and greatly reduces the interface design effort.
  - There are fewer sensor restrictions and more interface flexibility since HSYNC and VSYNC are allowed to vary.

However, there are a few restrictions on the HSYNC and VSYNC inputs that should be noted:

- The VSYNC blanking period must be no less than five PCLKs; the HSYNC blanking period must be no less than two PCLKs; and there must be 16 PCLKs between two valid lines (see Figure 8-5).
- The actual number of valid bytes during the active portion of a line must remain constant. This restriction is not new and is based on the fact that the number of valid pixels given to the VFE should stay constant when in one mode of operation.

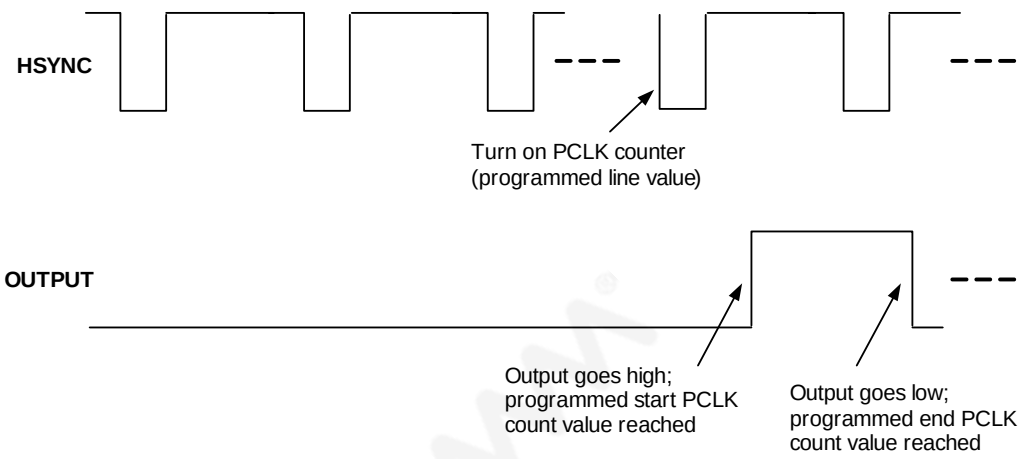


Figure 8-5 Synchronous timer operation

8.1.6 Multimedia applications using CAMIF

This section describes the different multimedia applications that can be used on the QSC62x0 device. The following applications use the camera interface, and possibly the MDP.

NOTE For more detailed information regarding the capabilities and implementations of the applications mentioned below, refer to the appropriate multimedia software documentation.

8.1.6.1 Qcamera

The Qcamera application is the most popular and standard of the multimedia applications. It integrates the camera sensor into the handset, using the QSC CAMIF for digital camera applications capable of taking, formatting, and storing digital pictures. While the hardware blocks and the interfaces were described in previous sections, the Qcamera application capabilities are summarized in Table 8-2.

Table 8-2 QCamera capabilities summary

Parameter	Capture	Preview
Interface	10 bits (8 optional)	10 bits (8 optional)
Maximum resolution	2.0 MP – QSC6240 device 3.0 MP – QSC6270 device	QCIF – 30 fps
Maximum PCLK	80 MHz	80 MHz

8.1.6.2 Qcamcorder

The Qcamcorder application uses the same hardware setup and interface as Qcamera, but it is used to record whole segments of digital film. The Qcamcorder application capabilities are summarized in Table 8-3.

Table 8-3 Qcamcorder capabilities summary

Parameter	QSC6240	QSC6270
Video codecs	MP4 Simple Pro, level 0b; H.263 Profile 0, level 45	
Audio codecs	AMR-NB with .3gp and .mp4 file formats	
Encode performance	15 fps at QCIF, 192 kbps	15 fps at QCIF, 192 kbps

8.1.6.3 Qtv

The Qtv application is used to decode and playback video clips from memory and for streaming content. The Qtv application capabilities are summarized in Table 8-4.

Table 8-4 Qtv capabilities summary

Parameter	QSC6240	QSC6270
Video codecs	MP4 Simple Pro, level 0b; H.263 Profile 0, level 45; H.264 Baseline v1.0; Windows Media Player v9; Real Player v10	
Audio codecs	AMR-NB, AMR-WB, AMR-WB, AAC, AAC Plus, Enhanced AAC Plus (audio-only playback), Windows Media Audio v8/9, Real Audio 8 (G2)	
File formats	.3gp, .mp4, .m, .wma, .wmv, .asf	
Streaming protocols	RTSP/SDP over TCP; TRP/RTCP over UDP; RDT over RTSP	
Playback performance		
MP4/H.263 [1]	15 fps at QCIF, 384 kbps	30 fps at QVGA, 384 kbps
H.264 [2]	15 fps at QCIF, 192 kbps	30 fps at QVGA, 384 kbps
Windows or Real (video+audio)	15 fps at QCIF, 128 kbps	15 fps at QVGA, 192 kbps
Streaming performance		
MP4/H.263 [1]	15 fps at QCIF, 256 kbps	15 fps at QVGA, 384 kbps
H.264 [2]	15 fps at QCIF, 128 kbps	15 fps at QCIF, 384 kbps
Windows or Real (video+audio)	15 fps at QCIF, 128 kbps	15 fps at QCIF, 128 kbps
Audio performance	128 kbps at 48 kHz	
Software supported	MMS frame work; video store file system; video capture driver; video display driver; MMC/SD drivers	

[1] With AMR, EVRC, QCELP, AAC, AAC+, EAAC+.

[2] With AMR, EVRC, AAC, AAC+, EAAC+.

8.1.6.4 Qvideophone

The Qvideophone application enables the handset to conduct videophone sessions. The Qvideophone application capabilities are summarized in Table 8-5.

Table 8-5 Qvideophone capabilities summary

Parameter	QSC6240	QSC6270
Video codecs	MP4 Simple Pro, level 0b; H.263 Profile 0, level 45; H.263 Profile 3, level 45 (decode only)	
Audio codecs	AMR-NB	
Protocols	H.324M protocol stack; H.245 control protocols; H.223 multiplexing	
Performance	15 fps at QCIF, 64 kbps	

8.2 Mobile display processor

The MDP is a hardware accelerator primarily responsible for transferring an updated image from the QSC memory subsystem to the LCD module. The transferring of an updated image is an operation that is shared between software, video processing, and graphics processing, so a common block helps to reduce redundant circuitry. The MDP is designed with the assumption that the LCD panel has an embedded LCD controller and a frame buffer. The image transfer is then the copying of the image from the QSC memory system to the frame buffer within the LCD module.

While the MDP is transferring an image to the LCD module, it can perform a final set of operations to the image. The set of operations that the MDP can perform has been chosen to maximize the efficiency of the QSC memory subsystems, typically removing two or more copy operations of the image to and from memory.

The MDP reduces redundant circuitry and offloads the ARM and aDSP from memory-transfer operations and a certain set of graphics and video operations.

8.2.1 MDP features

The MDP can support a set of graphics and video operations using various interfaces. These are summarized below:

- Frame update synchronization – To fully take advantage of this feature, the LCD must be capable of providing an LCD_VSYNC output to the QSC device that the MDP uses to calculate the location of the frame buffer's read pointer.
- Rotation (90, 180, and 270 degree) and flip (left/right and top/bottom) – The MDP is capable of providing 90, 180, and 270 degree rotations, and left-right and top-bottom flips as it transfers an image from memory to the LCD panel.

- Video functions
  - Chroma upsampling
  - Color conversion (such as YCbCr 4:2:0 to RGB 6:6:6 format)
  - Image size downscale by n/2, n/4, and 3n/4
  - Image size upscale by two, and upscale from QVGA (320 x 240) to WQVGA (400 x 246)

NOTE This refers only to display update capability and not multimedia application capabilities.

- Graphics functions
  - Alpha-blend
  - Transparency
  - Text overlay
- Flexible LCD interface
- Script processing

8.2.2 MDP LCD module interface

A block diagram of the interface between the MDP block and the LCD module is shown in Figure 8-6.

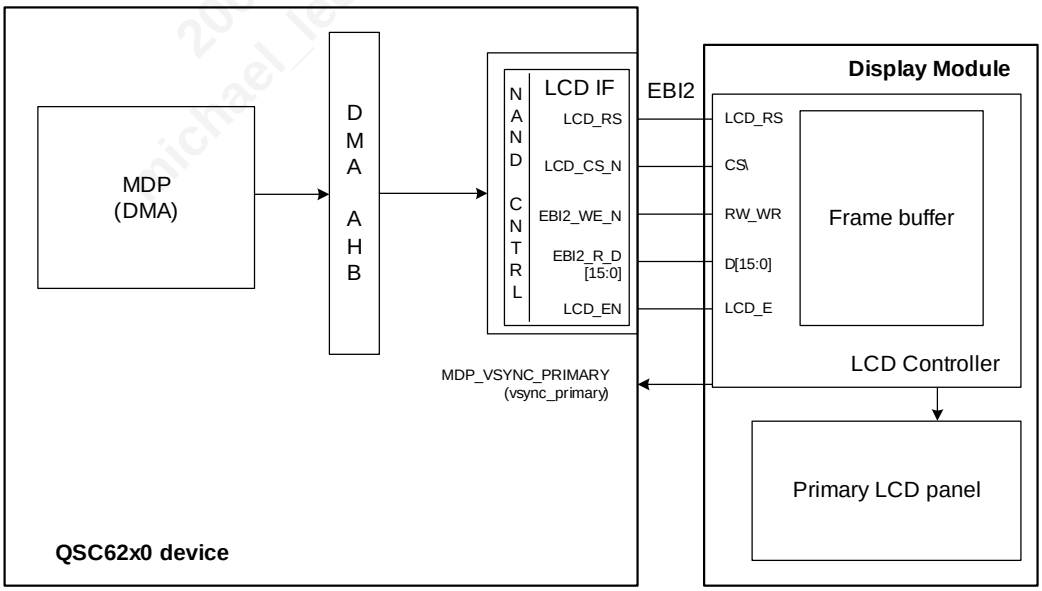


Figure 8-6 MDP-to-LCD interface functional block diagram

8.2.3 MDP-related design considerations

Design considerations relating the MDP include:

- The MDP is designed to work with LCD panels that have an LCD controller and a frame buffer.
- The MDP will not work with memory-mapped LCD modules.
- The recommended, and typical, refresh rate for the LCD is 60 Hz. Note that changing the refresh rate (in either direction) can lead to tearing.
- The MDP update rate should be greater than one refresh cycle but less than two refresh cycles.
- MDP is tuned primarily for the CIF display (288 x 352). The maximum width of the display is limited to 352. However, the length can increase with the cost of performance degradation.
  - WQVGA is supported only in portrait mode.
- In order to take full advantage of the capabilities of the MDP, it is strongly recommended to use LCD modules that output an external VSYNC signal. This signal is connected to the QSC62x0 device via GPIO_23 (MDP_VSYNC_P) and is what the MDP uses to calculate the read pointer location and to prevent tearing. MDP uses the VSYNC to calculate the approximate location of the LCD frame-buffer read pointer. This is sufficient to prevent tearing and is necessary to prevent the frame-buffer write pointer (MDP writes) from getting ahead of the frame-buffer read pointer.

8.3 Additional multimedia support

Supported video, audio, graphics, and messaging features are listed.

8.3.1 Video support

See Section 8.1.6.

8.3.2 Audio support

Table 8-6 Audio features supported

Feature	Description
Concurrency	Basic audio concurrency with all codecs: codec + voice call, codec + viewfinder, etc.
Speech codecs	ADPCM 8, 16, and 32 kHz AMR narrowband - all rates; AMR wideband (speech) G.711, G729A (TBD feature) GSM FR/HR/EFR
Multimedia codecs	MP3 decode only - up to 320 kbps at 48 kHz stereo/MP3 2.5 variable rate (VBR) AAC - up to 128 kbps; AAC Plus - up to 128 kbps; enhanced AAC Plus - up to 128 kbps (all at 48 kHz stereo) RealNetworks® Audio v8 Windows Media Audio v8/9 AMR wideband+ (multimedia)
Microphone	8 kHz and 16 kHz sampling rate, 16 bit 3 inputs - handset, headset, and auxiliary (all differential)
Speaker and buzzer	Up to 48 kHz sampling rate, 16 bit Mono - headset speaker, headset, and auxiliary Stereo - headset
Enhanced echo cancellation	For speakerphone applications
Vibrator	Via PM functional block
Audio AGC	Ensures less saturation of loud voice/sounds while simultaneously being able to amplify soft voices/sounds to reasonable audible levels
MIDI and multimedia support	CMX - up to 128 poly with time-synchronization support of MIDI, PNG, animation (SVG 1.1 and 1.2), WAV, text, LED/vibrate. XMF/DLS, MFi (requires DoCoMo license), SP-MIDI, and SMAF® audio playback through CMX player.
Audio playback from MMC/SD or network download OTA	MP3 or AAC decoding in SW, playback through stereo DAC MMC file system/API; MMC/SD drivers Media player
Multimedia services	MMS framework and file system RTP/RTSP
Streaming audio/video	Audio/Video synchronization
Voice memo conversation/record	PureVoice®
3D sound	QConcert (surround sound) - for headsets and speakers
Audio effects	Spectrum analyzer and equalizer PCM post-processing
Positional audio	QAudioFX™
AV synchronization	MP3, AAC, PCM, EVRC, AMR, Real, and Windows Any decoder required by Qtv (except PCM)

8.3.3 Messaging support

Table 8-7 Messaging features supported

Feature	Description
Short messaging services (SMS)	Text messages, maximum 160 characters UTF-8; UTF-16 (Unicode text encoding for SMS) ISO/IEC 10646 UCS-2 (UCS text encoding for SMS)
Multimedia messaging services (MMS)	Combination of video (MPEG-4), still image (JPEG), voice tag (AMR), and text sent as a message Software: MMS Framework; User Profile; SMIL 2.0 Basic; WAP/WSP/WTP and HTTP; UDP/TCP/IP and WAP/WDP Media types: text; graphics (GIF87a, GIF89a up to CIF); JPEG (encode and decode); MP3/AAC decode; MPEG-4 (encode and decode) - Simple Profile level 0, Simple Profile level 2, Advanced Simple Profile, and Simple Scalable Profile; H.263 Profile 0 level 10 (encode and decode)
Applications / OS	Java HW acceleration through ARM926EJS Jazelle core MIDP 2.0, CLDC/KVM 1.0.4, plus extensions for 3D graphics, location support, and video BREW version 3.1.5 or greater

9 Connectivity

QSC62x0 devices support many interfaces that provide handset users with connectivity to other people and data systems. All QSC connectivity functions are discussed in this chapter, including:

- Keypad
- USB-OTG
- High-speed UART
- USIM/MUSIM
- Secure digital I/O interfaces
- I^2C interface
- Auxiliary PCM interface
- SPI
- NFC
- Bluetooth
- FM radio
- WLAN

9.1 Keypad interface

QSC62x0 devices provide a keypad interface (Figure 9-1) that supports five sense lines, or columns, and (typically) five keypad rows (though more can be software defined). The columns are a set of five QSC pins that are used for sensing (KEYSENSE_xN); another set of pins are connected to the keypad's rows and are used for driving (KYPD_x). The sensed columns reveal when any keypad button is pressed, and then the rows are driven sequentially to determine precisely which keypad button was pressed.

The QSC device detects when any keypad button is pressed by ORing all keypad column signals together using a set of KEYSENSE_N pins. All KEYSENSE_N inputs are pulled high internally, and before a keypad button is pressed all rows are driven low. Without any keypad buttons being pressed all sensed inputs are high.

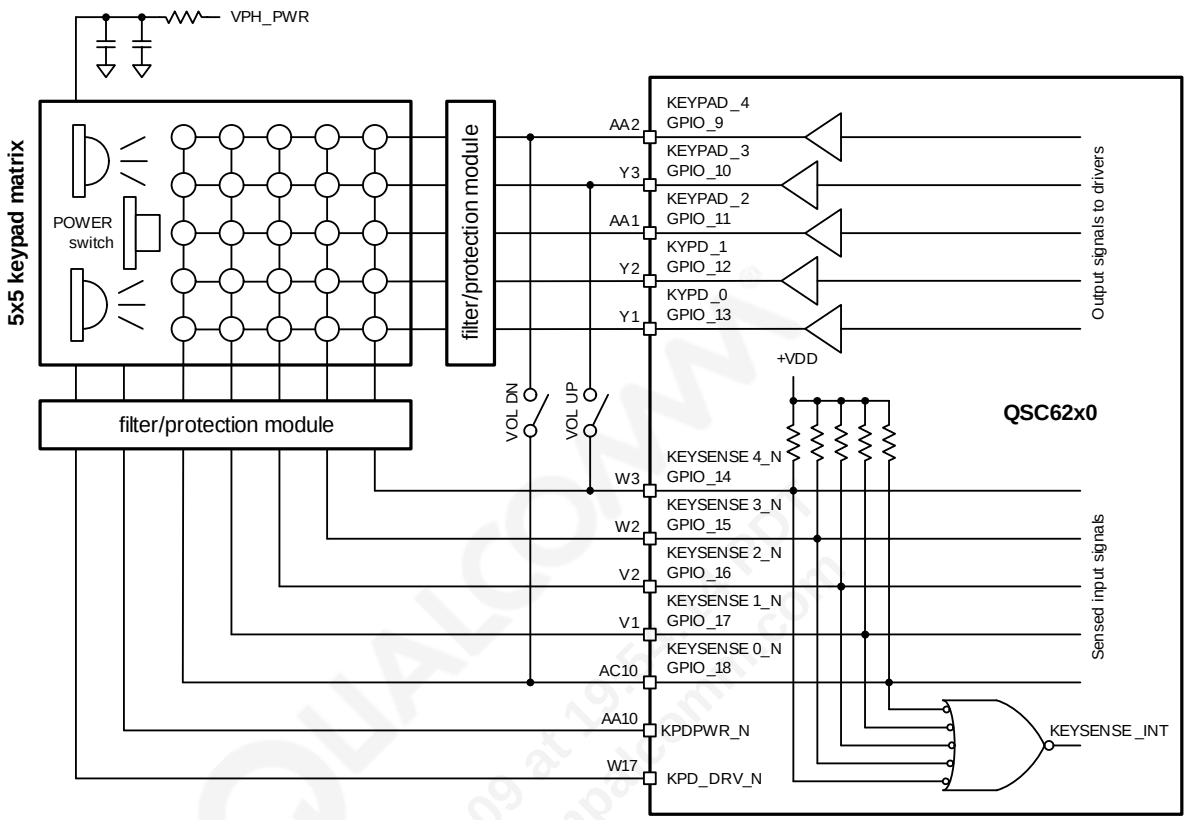


Figure 9-1 Example keypad connections - 5 x 5 matrix

When a keypad button is pressed, its corresponding column is pulled low (since all rows are low). Since it is an active low sense line, the KEYSENSE_INT signal is asserted whenever any keypad button, from any column, is pressed. When the interrupt signal is received, the QSC device begins scanning the keypad. During a scan, each row is sequentially driven low, one at a time. As each row is driven low, the columns are sensed. The precise keypad button being pressed is revealed when that button's column reads low while that button's row is driven low.

In addition to the switch matrix, the keypad interface includes:

- VPH_PWR (DC power) and KPD_DRV_N (current sink) for lighting the keypad backlight
- Support for volume up and volume down buttons
- Detection of the power on/off button via KPDPWR_N

The KPD_DRV_N and KPDPWR_N functions are addressed within the power management sections.

9.1.1 Keypad features

The QSC62x0 device provides 10 pins for the keypad interface, supporting a 5 × 5 keypad matrix. Supplemental functions such as volume up, volume down, power on/off, and backlighting are supported as well.

9.1.2 Keypad connections

Ten pins (Table 9-1) are used for supporting up to a 5 × 5 keypad matrix; five pins for sensing, five for driving.

Table 9-1 Keypad connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
Keypad interface					
KEYPAD_4 (GPIO_9)	AA2	P1 (1.8 V)	DO	1-8 (1)	Bit 4 drive to the pad matrix
KEYPAD_3 (GPIO_10)	Y3	P1 (1.8 V)	DO	1-8 (1)	Bit 3 drive to the pad matrix
KEYPAD_2 (GPIO_11)	AA1	P1 (1.8 V)	DO	1-8 (1)	Bit 2 drive to the pad matrix
KEYPAD_1 (GPIO_12)	Y2	P1 (1.8 V)	DO	1-8 (1)	Bit 1 drive to the pad matrix
KEYPAD_0 (GPIO_13)	Y1	P1 (1.8 V)	DO	1-8 (1)	Bit 0 drive to the pad matrix
KEYSENSE4_N (GPIO_14)	W3	P1 (1.8 V)	DI	–	Bit 4 for sensing key press on pad matrix
KEYSENSE3_N (GPIO_15)	W2	P1 (1.8 V)	DI	–	Bit 3 for sensing key press on pad matrix
KEYSENSE2_N (GPIO_16)	V2	P1 (1.8 V)	DI	–	Bit 2 for sensing key press on pad matrix
KEYSENSE1_N (GPIO_17)	V1	P1 (1.8 V)	DI	–	Bit 1 for sensing key press on pad matrix
KEYSENSE0_N (GPIO_18)	W1	P1 (1.8 V)	DI	–	Bit 0 for sensing key press on pad matrix

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

9.2 USB interface

9.2.1 Introduction

The QSC device contains a USB interface. This interface is compliant with the USB 2.0 specification.

When two devices are connected via a USB interface, one of the devices must act as a host and the other device must act as a peripheral. The host is responsible for initiating and

controlling traffic on the bus. The USB specification requires PCs to act as hosts, and other devices such as printers, keyboards, mice, etc. to act as peripherals.

The USB 2.0 specification requires hosts such as PCs to support all three USB speeds, namely low-speed (1.5 Mbps), full-speed (12 Mbps) and high-speed (480 Mbps). The USB 2.0 specification allows peripherals to support any one or more of these speeds.

The QSC device is compliant with the USB 2.0 specification. The QSC device supports full-speed and high-speed when acting as a peripheral and low-speed, full-speed, and high-speed when acting as a host.

There are two USB controllers embedded in QSC6240:

1. Primary USB controller
2. Secondary USB controller

Both are high-speed USB controllers. The primary USB port has an integrated controller and PHY and can operate at all three speeds. The secondary USB port has an integrated controller. It can only operate at full speed. Figure 9-2 shows the architecture for both primary and secondary USB controllers.

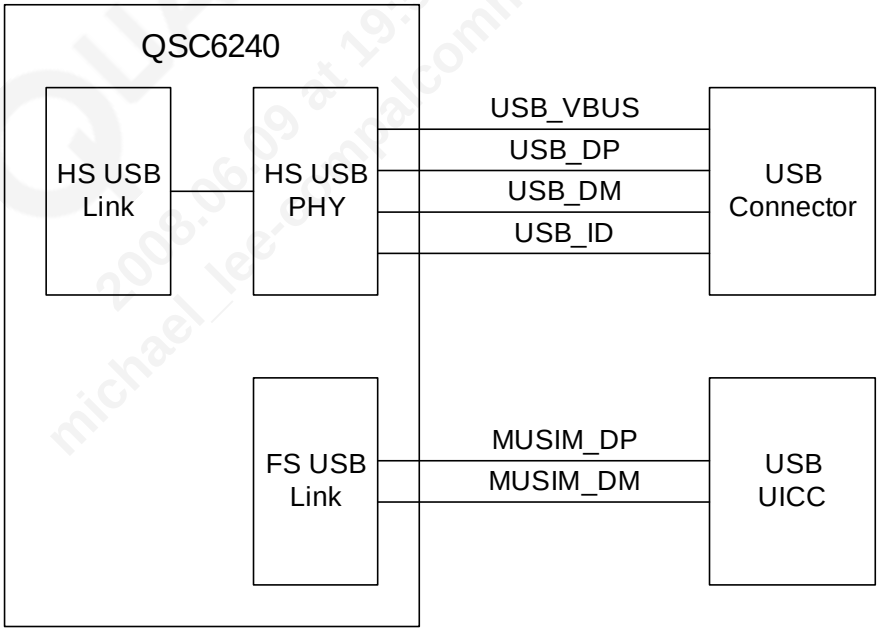


Figure 9-2 Primary and secondary USB controller architecture in QSC6240

9.2.1.1 Primary USB controller

The primary USB controller contains the following features:

- It has a high-speed USB controller (specification 2.0).
- It acts as both a host and a peripheral.
- In peripheral mode, the controller has eight in/out pairs of endpoints.

- In host mode, the number of possible endpoints is determined by software and can be greater than 8.
- Built-in high speed PHY
- Four USB pins dedicated for high speed USB interface. They are:
  - USB_VBUS
  - USB_DP
  - USB_DN
  - USB_ID

9.2.1.2 Peripheral applications

Given the appropriate software, the primary USB controller can support the following applications while acting as a peripheral:

- Mass storage
- MTP for music and video content transfers
- CDC/ACM for modem
- CDC/ECM for data services
- CDC/OBEX for NMEA and diagnostic
- SICD for PictBridge

9.2.1.3 Host applications

Given the appropriate software, the primary USB controller can support the following applications while acting as a host:

- HID-supporting keyboard
- Mouse and gamepad controller connectivity
- Mass storage supporting USB flash drive and HDD connectivity

9.2.1.4 Endpoints

The primary controller supports eight in/out endpoint pairs when acting as peripheral. The number of endpoints when acting as a host is determined by software and can be greater than eight.

Each endpoint in either host or peripheral mode can be configured for control, interrupt, or bulk. Isochronous endpoints are supported in hardware but are not yet used by any software application.

9.2.2 Secondary USB controller

Below are the secondary USB controller features:

- The secondary controller has the same logic and software interface as the primary USB port.
- The secondary port does not have an integrated high-speed PHY.
- The secondary port has smaller buffers than the primary port and only a full-speed transceiver interface. Thus, the secondary port can only support full-speed operation.
- The secondary controller acts as both a host and a peripheral.
- In peripheral mode, the controller has four in/out pairs of endpoints.
- In host mode, the number of possible endpoints is determined by software and can be greater than four.

9.2.2.1 Host application

The secondary USB controller can support USB-UICC operation while acting in host mode. Due to a maximum buffer-size limitation in the secondary USB controller, the USB-UICC interface's maximum speed is up to that of the full-speed USB interface (12 Mbps). Two pins are dedicated to this operation along with the three USIM pins.

9.2.2.2 Peripheral application

The secondary USB controller can only be used in host mode to communicate with a UICC. It cannot be used in peripheral mode.

9.2.3 USB charging

The QSC device supports a variety of charging source options, including USB charging. Details are provided in the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* (80-VF846-5).

9.2.4 Powering peripherals and accessories

The OTG Supplement requires that an OTG device be capable of outputting at least 8 mA at 5 V to a USB peripheral. The QSC device includes a boost regulator that generates 5 V from the battery voltage, and is capable of outputting 200 mA on USB_VBUS.

9.2.5 USB signaling

When the USB_OE_N pin is high, QSC circuits are receiving data from the USB interface and the baseband interface circuits operate as shown in Table 9-2.

Table 9-2 USB receive operation

Suspend	Inputs		Outputs		Comments
	D+	D-	DAT_RXD	SE0_TXD	
0	0	0	diff rcv (1)	1	USB DAT don't care
0	1	0	1	0	
0	0	1	0	0	
0	1	1	diff rcv (1)	0	Not valid
1	0	0	0	1	
1	1	0	1	0	
1	0	1	0	0	
1	1	1	1	0	

When the suspend bit is set, the USB transceiver is in a low-power mode and is using the single-ended receivers to detect activity.

When the USB_OE_N signal is low, QSC circuits are transmitting data onto the USB interface and the baseband interface circuits operate as shown in Table 9-3.

Table 9-3 USB transmit operation

Inputs		Outputs		Comments
DAT_RXD	SE0_TXD	D+	D-	
0	0	0	1	Speed: 0 = low-speed 1 = full-speed
1	0	1	0	
0	1	0	0	
1	1	0	0	

9.2.6 SRP and HNP

SRP and HNP are described in the OTG supplement, and are supported by the QSC62x0 device.

A USB cable has a type-A plug (std-A or mini-A) on one end and a type-B plug (std-B or mini-B) on the other end. An OTG device is defined as a device that has a mini-AB receptacle, which can accept either a mini-A plug or a mini-B plug. An OTG session is defined as a period of time during which the A-device is providing power on VBUS.

When a mini-A plug is inserted into an OTG device, that device becomes an A-device. The A-device provides power during a session and defaults to being a USB host at the start of each session. When a mini-B plug is inserted into an OTG device, that device becomes a B-device. The B-device is allowed to draw power during a session and defaults to being a USB peripheral at the start of each session.

SRP is a protocol that allows a B-device to wake up an A-device that is not supplying power on VBUS and request that it start a session. HNP is a protocol that allows a B-device to negotiate with an A-device for the role of USB host. The handshaking

associated with SRP and HNP requires that an OTG device be capable of connecting and disconnecting the following resistors from the VBUS, D+ and D- lines, as highlighted in Figure 9-3:

- VBUS charge
- VBUS discharge
- D+ pull-up
- D+ pull-down
- D- pull-up
- D- pull-down

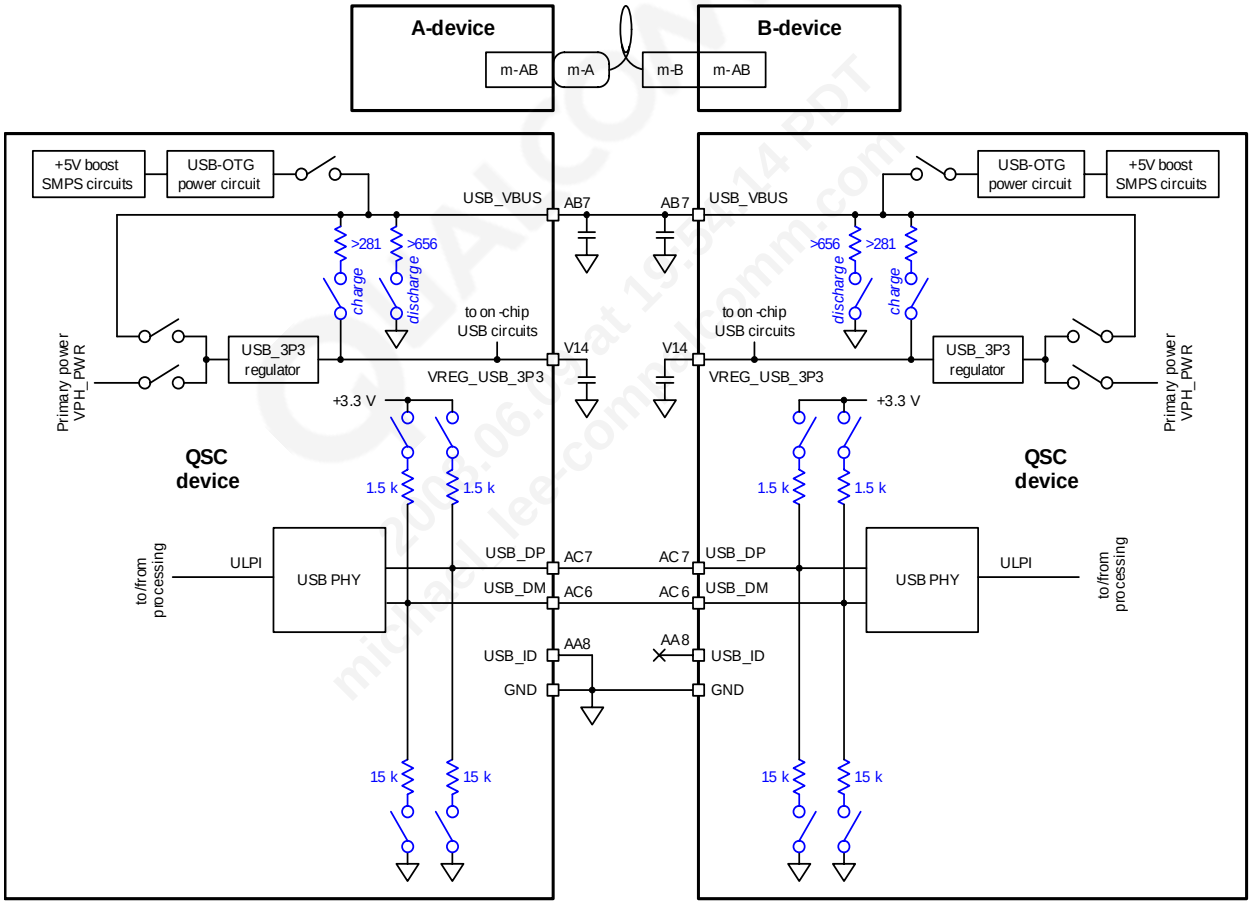


Figure 9-3 HNP handshaking resistors

The QSC62x0 device integrates the above resistors internally and supports both SRP and HNP. These switches are controlled via software.

9.2.7 UART signaling

The USB analog interface requires that a phone be able to output the UART transmit data (TXD) signal on D-, and be able to input the UART receive data (RXD) signal on D+. As shown in Figure 9-2, this mode is supported by the QSC62x0 device and enabled via software.

9.2.8 USB audio

The USB pins intended to interface with external circuits, USB_DP and USB_DM, can be configured as an audio output capable of driving a USB audio interface (Figure 9-4).

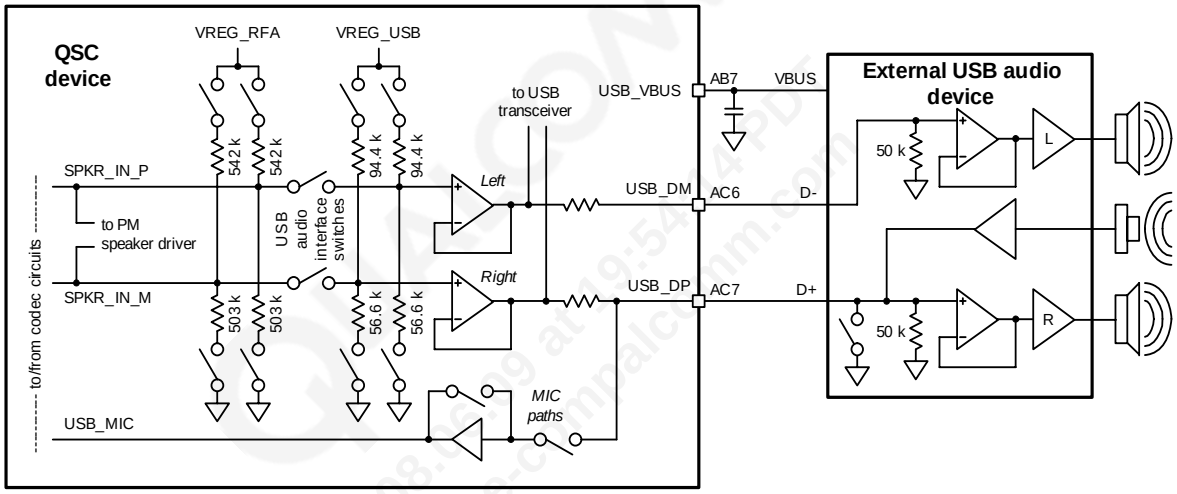


Figure 9-4 USB audio interface

The codec's audio output is delivered to the speaker driver input that is also connected to the USB audio buffer circuits. The USB audio path includes USB analog interface switches that are low-impedance transmission gates. The pull-up and pull-down resistors on the D- line generate a DC bias for the speaker signal and are used in both stereo and mono mode. The bias voltage generated by these resistors is nominally 1.25 V. The pull-up and pull-down resistors on the D+ line perform the same function but are only used in stereo mode.

Only one audio circuit is enabled at a time – either the speaker driver or the USB audio circuit. When the USB audio circuit is enabled, the USB transceiver circuits are disabled and switches at the output of the buffers route the audio signals to the USB transceiver I/O pins USB_DP and USB_DM. The buffers provide unity gain, isolate the speaker driver from the USB digital data, and maintain the speaker driver's proper input impedance.

When the USB audio circuits are disabled, they are powered down to save power. The right and left buffers can be enabled together or individually. The right control enables the USB_DP output and the left control enables USB_DM. When a buffer is enabled, its input and output are both biased around 1.25 V. The USB analog interface connection is DC-coupled so AC-coupling capacitors should not be inserted at the USB_DM and USB_DP pins.

Mono or stereo signals are routed through the USB circuits as follows:

- USB stereo:
  - Audio input signals from LINE_ON (right channel) and LINE_OP (left channel)
  - Left and right channel USB buffers are enabled
  - LINE_ON -> SPKR_IN_M -> USB_DP (right channel)
  - LINE_OP -> SPKR_IN_P -> USB_DM (left channel)
- USB mono, single-ended:
  - Audio input signal from LINE_OP only
  - Left channel USB buffer is enabled; right channel USB buffer is disabled and powered down
  - LINE_OP -> SPKR_IN_P -> USB_D M (left channel)
  - MIC_EN is set to enable the USB headset to phone audio
  - USB_DP -> USB_MIC
- USB mono, differential – not supported

While in the UART mode, the D- line is used to send TXD data to the USB analog interface and the D+ line is used to receive RXD data from the USB analog interface.

While in mono mode, the D- line is used to send the speaker signal to the USB analog interface and the D+ line is used to receive the microphone signal from the USB analog interface. The phone is able to interrupt the USB analog interface by injecting a short positive pulse on the D- line. Likewise, the USB analog interface is able to interrupt the phone by injecting a short negative pulse on the D+ line. These pulses are too short to cause a click or pop on the speaker. Using coding, these pulses enable data transmissions without interfering with the audio. This feature is referred to as data-during-audio in the USB analog interface specification.

While in stereo mode, the D- line is used to drive the left speaker and the D+ line is used to drive the right speaker.

The USB ID line is not required when using a 4-line interface.

9.3 High-speed UART

The QSC62x0 device includes one UART port that can be used in the low-speed, full-speed, and high-speed modes. The UART communicates with serial data ports that conform to the RS-232 interface protocol. With a properly written and user-defined download program, the UART can be used as the handset's serial data port for testing and debugging, and can support additional interface functions such as an external keypad or ringer. If the handset uses EEPROM or flash memory, the UART can be used to load and/or upgrade system software.

The UART interface is used for data transport during Bluetooth 2.0 operation when the BTS4020™ SoC is used.

The serial data port is a UART channel (Figure 9-5). The UART processes both transmitted and received data with separate transmit (Tx) and receive (Rx) FIFOs (512 bytes each). Supporting circuits include interrupt control, clock source, bit-rate generator (BRG), and microprocessor interface. Each of these UART functions is described later in this chapter.

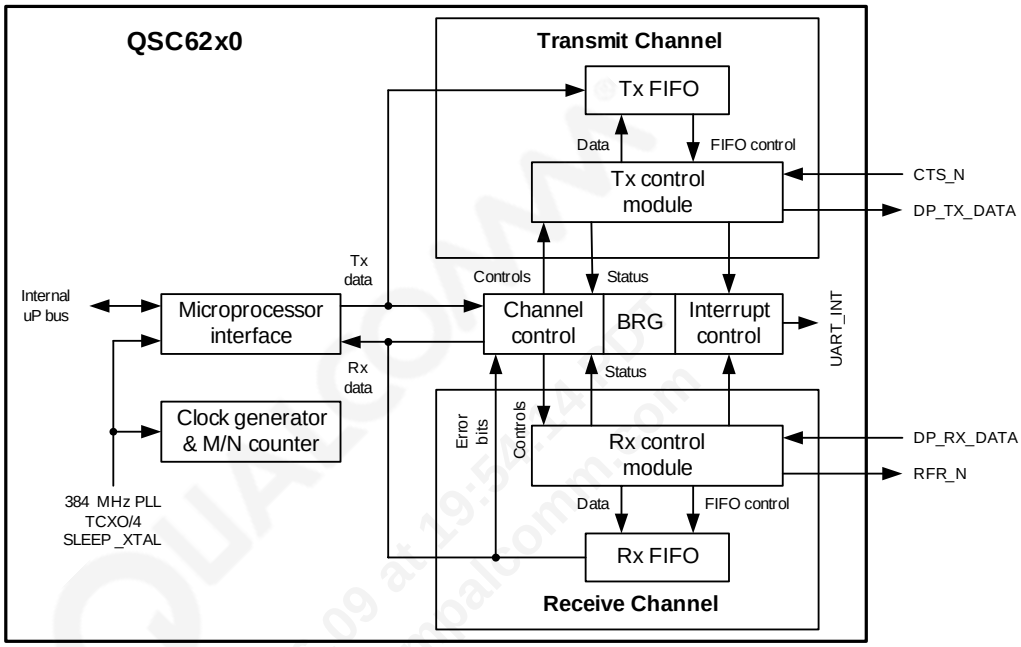


Figure 9-5 UART functional block diagram

9.3.1 UART features

The UART has several features that are common to both transmit and receive modes:

- UART supports the external BTS4020 Bluetooth System-on-Chip using its high-speed (4 Mbps) mode
- Hardware handshaking
- Programmable parameters
  - Data size
  - Stop bits
  - Parity
  - Bit rate
  - Selectable clock source
    - Command registers that control the broad UART functions such as enable, disable, and reset of the Tx and/or Rx channel, plus start and stop breaks

Some UART signals use GPIOs; this allows them to be configured for alternate functions as discussed throughout this chapter. In fact, the UART Rx data pins are high-voltage tolerant, making them even more flexible.

9.3.2 UART connections

UART connections are shown in an example Bluetooth application in Figure 9-6 and then listed in Table 9-4.

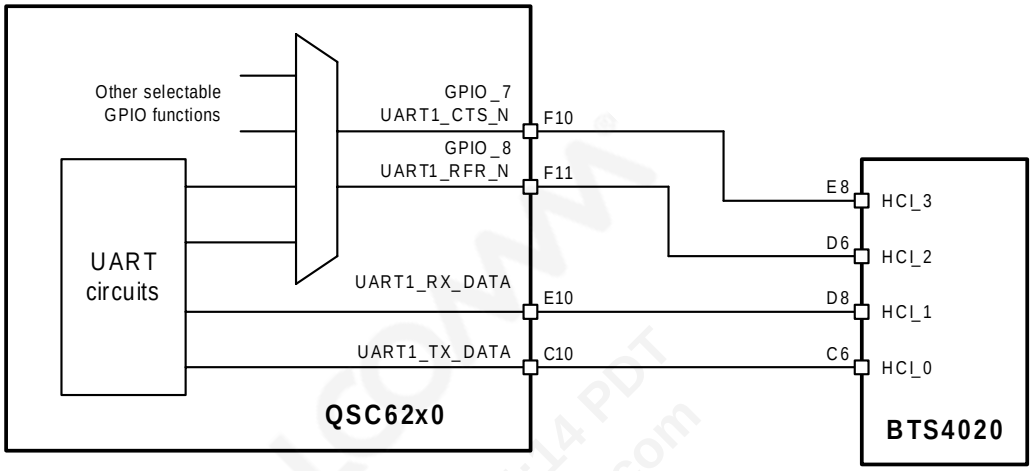


Figure 9-6 UART connections

Table 9-4 UART connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
UART1_RXD	E10	P5 (1.8 V)	DI (HV)	–	High-speed UART receive data input
UART1_TXD	C10	P5 (1.8 V)	DO	1-8 (1)	High-speed UART transmit data output
UART1_RFR_N (GPIO_8)	F11	P5 (1.8 V)	DO	1-8 (1)	High-speed UART ready for receive signal
UART1_CTS_N (GPIO_7)	F10	P5 (1.8 V)	DI	–	High-speed UART clear to send signal

1 The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

2 Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

Additional details about the BTS-QSC Bluetooth UART interface is available in the *BTS4020 System-on-Chip (SOC) User Guide* (80-VE132-3).

9.3.3 UART transmitter

The UART transmit channel contains a Tx control module plus a 512-byte Tx FIFO. The Tx FIFO accepts parallel data from the microprocessor. The Tx control module reads data from the Tx FIFO and sends it out serially, adding a start bit, optional parity bits, and stop bit(s).

The TXRDY bit in the status register (SR) sets (1) whenever the Tx FIFO has space available. The TXLEV interrupt is asserted when the Tx FIFO has fewer than the number of characters programmed in the transmit FIFO watermark register (TFWR).

Idling or disabling the Tx channel holds the DP_TX_DATA pin in a marking state (high). Enabling the Tx channel with a character waiting in the Tx FIFO loads that character into the Tx shift register. Next, a start bit is transmitted (Tx low) for one bit time, followed by each bit in the character (LSB first), then an optional parity bit followed by stop bits (Tx high).

If the Tx FIFO is not empty after the previous transmission, the Tx control module begins another transmission by loading the next Tx FIFO character into the Tx shift register. If the Tx FIFO is empty after the previous transmission, the SR's TXEMT (Tx empty) bit is set (1). Setting TXEMT indicates an under-run in the Tx channel. The TXEMT bit clears (0) once the transmit shift register has a new character from the Tx FIFO.

The transmitting channel can also be enabled or disabled using the command register (CR). When disabled, the Tx channel continues transmitting any characters in the Tx shift register until the register is empty. When transmission ends, the DP_TX_DATA pin goes into a marking state (high).

If the CR issues a start break command after the Tx channel transmits all the characters in the Tx FIFO and shift register, the Tx channel forces the DP_TX_DATA pin low. The DP_TX_DATA pin remains low until the CR issues a stop-break command.

When the CR issues a reset-transmitter command, the Tx channel ceases transmissions. The DP_TX_DATA pin enters a marking state and flushes the Tx FIFO.

And finally, the CTS control feature can be turned on by programming the UART mode register 1 (MR1). When on, the Tx channel checks the CTS_N input before transmitting a character. If CTS_N is high, the Tx channel stops transmitting and continues marking. If CTS_N is low, transmission begins or continues. If CTS_N goes high in the middle of character transmission, the Tx channel waits for a completed transmission before entering the marking state. The Tx channel can generate a programmable interrupt whenever CTS_N changes states.

9.3.4 UART receiver

The UART receive channel contains an Rx control module plus a 512-byte Rx FIFO. Each byte in the Rx FIFO has two bits of corresponding status information. Serial data is input via the DP_RX_DATA pin which is pulled low internally. The Rx control module converts the serial data into parallel data and loads it into the Rx FIFO.

If the Rx FIFO is not full, the control module writes the received character to the Rx FIFO. When the Rx control module writes a received character to an empty Rx FIFO, the RXRDY bit in the SR is set (1). If the Rx FIFO becomes full, the RXFULL bit sets (1). The RXFULL bit clears (0) when a character is read from the Rx FIFO. The RXRDY bit clears (0) when the Rx FIFO becomes empty once again. When the Rx FIFO has more characters than what was programmed in the receive FIFO watermark register (RFWR), the Rx channel asserts a RXLEV interrupt. If a character is waiting in the Rx FIFO for a programmed time period (STALE_TIMEOUT), an RXSTALE interrupt occurs. The Rx channel asserts an RXHUNT interrupt whenever the Rx channel receives a character that matches the value found in the hunt character register (HCR).

Two status bits are associated with each 10-bit data word in the Rx FIFO. One status bit defines the received break condition; the other status bit is the logical-OR of a parity error or framing error. MR2:ERROR_MODE determines the character error mode or the block error mode. In character error mode, the SR's error bits apply only to the byte waiting to be read from the Rx FIFO. In block mode, the SR's error bits are the logical-OR of all incoming error bits since the command register last issued a reset-error-status command. Whether in character error mode or block mode, reading the status register has no effect on the Rx FIFO.

If the Rx FIFO is full and the Rx channel receives a new character, that character remains in the Rx shift register until a position becomes available in the Rx FIFO. Any additional incoming characters are lost until space becomes available in the Rx FIFO. When characters are lost, an overrun error occurs and SR:OVERRUN is set (1). Once SR:OVERRUN is set, it remains set until the CR issues a reset error status command. An overrun error does not affect the Rx FIFO's contents.

The receive channel can also be disabled using the CR. Incoming characters are now ignored, but characters already in the Rx FIFO remain unchanged and can be read by the microprocessor. Having the CR issue a reset-receiver command flushes the Rx FIFO and clears (0) the status bits. The receive channel remains disabled until it is re-enabled by setting the appropriate bits on the CR.

The automatic ready-for-receiving (RFR_N) mode can be enabled on the UART's mode register 1 (MR1). In this mode, the RFR_N pin goes high when the Rx FIFO level is the same or greater than the value programmed in MR1. When the Rx FIFO level falls below the programmed level, the RFR_N pin returns to a low state. This feature prevents overruns by connecting the channel's RFR_N pin to a transmitting device CTS_N pin.

9.3.5 UART clock source

The UART interface allows independent selection of its clock source and programmable M/N counter to sets its rate. Available sources include:

- PLL0 (384 MHz)
- PLL1 (460.8 MHz)
- TCXO (19.2 MHz)
- SLEEP_CLK (32.768 kHz)

The UART clock is derived from its selected clock source using the clock generator and M/N counter circuits within the UART block. Writing the value 0 (zero) to the M registers cause the M/N counter to output a 0 Hz clock, effectively turning off the clock and putting the UART into its power-save mode. When the clock is disabled, the M/N counter registers are the only accessible UART registers.

The 384 MHz PLL source can be selected to generate a 64 MHz base frequency; this clock is needed to support Bluetooth 2.0 ASICs at the desired 4 Mbps rate.

9.3.6 UART bit rate generator

The desired bit rate for the receive and transmit channels is selected using the UART_CSR register. Depending upon the selected clock source and selected M/N values, the fields in the UART_CSR register should be programmed to achieve the desired bit rate for either direction (transmit or receive). For normal UART operation (non-Bluetooth), the maximum supported bit rate is 230.4 kbps; 4 Mbps is supported for Bluetooth 2.0.

The normal UART BRG generates enables to the Tx and Rx channels that are 16 times the nominal bit rate. The BRG selects one of the 16 possible bit rates as defined in the clock select register (Table 9-5) and sends the selected bit rate to the Tx channel (CSR bits [3:0]) and Rx channel (CSR bits [7:4]). The bit rate values listed in the table assume a system clock frequency of 1.8432 MHz and are generated by dividing the UART system clock.

Table 9-5 UART bit rate selection

CSR bits	Settings	Bit rate (bps) with XO/4 selected	Bit rate (bps) with XO selected
Rx: bit [7:4]	0000	75	300
Tx: bit [3:0]	0001	150	600
	0010	300	1.2 k
	0011	600	2.4 k
	0100	1.2 k	3.6 k
	0101	2.4 k	4.8 k
	0110	3.6 k	7.2 k
	0111	4.8 k	9.6 k
	1000	7.2 k	14.4 k
	1001	9.6 k	19.2 k
	1010	14.4 k	28.8 k
	1011	19.2 k	38.4 k
	1100	28.8 k	57.6 k
	1101	38.4 k	115.2 k
	1110	57.6 k	230.4 k
	1111	115.2 k	460.8 k [1]

[1] Not supported by QSC62x0 devices.

9.3.7 UART interrupts

The UART can assert six separate interrupts: DELTA_CTS, TXLEV, RXLEV, RXSTALE, RXBREAK, and RXHUNT. The interrupt status register (ISR) shows each interrupt's status bit independent of the interrupt mask register (IMR) bit state. The mask interrupt status register (MISR) returns the bitwise AND of the ISR and IMR registers. The interrupt functions in the UART_ISR register can be enabled or disabled by setting the appropriate bits in the IMR. Furthermore, the conditions for the RXSTALE interrupt can be defined by the user on the UART_IPR register.

9.4 USIM

With proper configuration of its GPIOs, the QSC62x0 device supports SIMs, including UMTS modules (USIM) and large storage capacity modules (USB-UICC). The identity modules are smart cards for wireless handset applications; they provide personal authentication information that allows the mobile station or handset to be connected with the network. The SIM card can be inserted into any QSC-equipped handset to enable its user to receive or make calls and receive other subscribed services.

The internal power management circuits, USIM circuity, and USIM pads allow for implementing both 1.8 V and/or 3 V cards via a direct connection.

9.4.1 USIM features

Key USIM features are:

- USIM, SIM, and large capacity (USB-UICC) support
- Data rates up to 4 MHz
- Support for dual-voltage cards (3 V and 1.8 V)

All USIM signals use GPIOs; this allows them to be configured for alternate functions as discussed throughout this chapter. The associated pads are powered off using VREG_USIM and can be either 1.8 V or 2.85 V (nominal).

9.4.2 USIM connections

Recommended GPIOs for implementing USIM functions are listed in Table 9-6.

Table 9-6 USIM connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
USIM_DATA (GPIO_47)	AC5	P3 (1.8/2.85V)	B	1-8/2-16 (6/12)	USIM data
USIM_CLK (GPIO_46)	AB6	P3 (1.8/2.85V)	DO	1-8/2-16 (6/12)	USIM clock
USIM_RESET (GPIO_45)	AC4	P3 (1.8/2.85V)	DO	1-8/2-16 (6/12)	USIM reset
MUSIM_DP (GPIO_73)	AB8	P3 (1.8/2.85V)	B	1-8/2-16 (6/12)	USB-UICC data plus line
MUSIM_DM (GPIO_72)	AA7	P3 (1.8/2.85V)	B	1-8/2-16 (6/12)	USB-UICC data minus line

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

9.4.3 USIM software implementation

On powerup or after a soft reset, the clock and data lines to the USIM circuits are active as they go through the initialization process. After initialization, the operation of the slot depends on whether a card is detected or not. When operating with a card, the data line always stays in the active high state (the marking state). Even though the line is not actively transmitting between accesses, it is still active. The clock, on the other hand, is only on when actively reading the card, and is turned off between accesses. The state of the clock when it is turned off depends on the characteristics of the card, and may be high or low. However, even between accesses, the interface is still active. The data, reset, and power lines all remain high (when reset is active low). The clock is turned off in order to save power and is turned on to access the card.

If for some reason the slot cannot be communicated with (card not inserted, card not recognized, broken connection, etc.), the interface is deactivated. In this state, all lines are low (again, clock state depends on card characteristic), and there is no chance to operate or communicate with the card without reinitializing.

Once a USIM card is inserted and initialized on powerup, the interface is always on (as described above), even when the QSC is in its sleep mode. This characteristic can give rise to current consumption problems. It is important to understand that only on powerup of the QSC, and not during regular operation, can a USIM card be recognized and initialized.

A card is recognized on powerup as follows:

- All signals and power are off (low)
- Power up the interface at 1.8 V (VREG_USIM = 1.8 V). Wait for reply from card.
- If a card is detected, finish initialization and move to the next device driver. If no card is detected, power off the interface and regulator.
- Power up again to the 3.0 V range (2.85 V nominal). Wait for reply from card.
- If a card is detected, finish initialization and move to next device driver. If no card is detected, power off the interface and regulator.

At this point, the QSC will not try to detect a card again. It assumes neither a 1.8 V nor a 3.0 V card is present, and the UIM interface will remain off until reset or power is cycled.

9.5 Secure digital interfaces

In general terms, a secure digital (SD) memory card is a flash-based memory card that is specifically designed to meet the security, capacity, performance, and environment requirements inherent in newly emerging audio and video consumer electronic devices.

As discussed in this section, the QSC62x0 device supports two SD interfaces that conform to the secure digital physical layer specification.

9.5.1 Secure digital features

As mentioned, the QSC secure digital interfaces support the SD physical layer specification. The following features are implemented:

- Two SDIO interfaces
  - One intended for WLAN connections (external SoC)
  - One intended for media/audio connections
- Supports 4-bit SD and 1-bit SD
- SW configurable edge latching, data/command value changes (rising or falling edge)
- Clock gating for power saving (can even turn the clock off when the bus is idle)
- Flow control option to prevent overflow and underflow
- MCLK output up to 52 MHz

9.5.2 Secure digital connections

SD functions are implemented using GPIOs and one MPP. Two examples are shown in Figure 9-7 (SDIO #1 interfacing with a microSD card and SDIO #2 interfacing to a WLAN SoC). All SDIO pins are listed in Table 9-7.

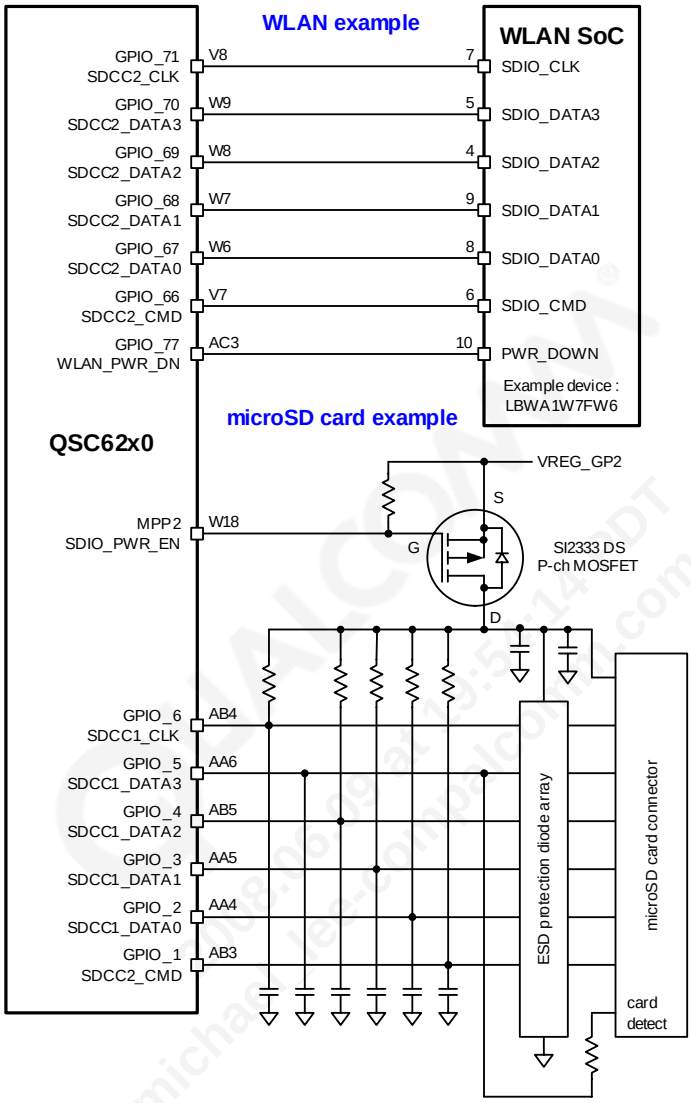


Figure 9-7 Secure digital connections

Table 9-7 Secure digital connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
Secure digital I/O #1					
SDCC1_CLK (GPIO_6)	AB4	P7 (2.6 V)	DO	2-16 (8)	Output clock for SDIO1 device
SDCC1_DATA3 (GPIO_5)	AA6	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO1 data bit 3
SDCC1_DATA2 (GPIO_4)	AB5	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO1 data bit 2
SDCC1_DATA1 (GPIO_3)	AA5	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO1 data bit 1
SDCC1_DATA0 (GPIO_2)	AA4	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO1 data bit 0
SDCC1_CMD (GPIO_1)	AB3	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO1 command bit
SDIO_PWR_EN (MPP2)	W18	–	DO	–	Power-enable to the secure digital card
Secure digital I/O #2					
SDCC2_CLK (GPIO_71)	V8	P7 (2. 6V)	DO	2-16 (8)	Output clock for SDIO2 device
SDCC2_DATA3 (GPIO_70)	W9	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO2 data bit 3
SDCC2_DATA2 (GPIO_69)	W8	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO2 data bit 2
SDCC2_DATA1 (GPIO_68)	W7	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO2 data bit 1
SDCC2_DATA0 (GPIO_67)	W6	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO2 data bit 0
SDCC2_CMD (GPIO_66)	V7	P7 (2.6 V)	B (HV)	2-16 (2)	SDIO2 command bit
WLAN_PWR_DN (GPIO_77)	AC3	P7 (2.6 V)	B (HV)	2-16 (2)	Powerdown signal to the WLAN device

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

9.5.3 Clock output

Other baseband circuits supply the master clock outputs through GPIO_6 and GPIO_71 (SDCC1_CLK and SDCC2_CLK). These clocks are designed to work with SD interfaces and can go up to 52 MHz, enabling maximum data rates of 100 Mbps.

The SDCC clock outputs are configured by the SDCCx_CLK_MD and SDCCx_CLK_NS registers. These registers are used to select a source and then divide it using either a modulo divider (div2 or div4) or an M/N:D counter. The M/N value is essentially the fraction of the source frequency intended at the output frequency. The D value is used to select a specific duty cycle; for a 50% duty cycle, D should be set to N/2.

For the cleanest output, an even divisor (divide by 2, 4, 6, 8, etc.) should be used, or at least whole integer divisors. If fractions are used, the output will jitter depending upon the M and N values used. It is important to make sure the required frequency can be achieved with acceptable jitter to the clocked device.

NOTE Upon request, QCT will help optimize and verify M, N, and D values.

9.6 I²C Interface

I²C is a two-wire bus for inter-IC communication that supports any IC fabrication process (NMOS, CMOS, bipolar, etc.). Two wires (or lines), serial data (SDA), and serial clock (SCL), carry information between the connected devices. Each device is recognized by a unique address (whether it's a microcontroller, memory, LCD driver, stereo DAC, or keyboard interface) and can operate as either a transmitter or receiver, depending on the device function.

A simplified description of I²C bus operation is given below:

- The master generates a START condition, signaling all ICs on the bus to listen for data.
- The master writes a 7-bit address, followed by a read/write bit to select the target device and to define whether it is a transmitter or a receiver.
- The target device sends an acknowledge bit over the bus. The master must read this bit to determine whether the addressed target device is on the bus.
- Depending on the value of the read/write bit, any number of 8-bit messages can be transmitted or received by the master. These messages are specific to the I²C device used. After 8 message bits are written to the bus, the transmitter will receive an acknowledge bit. This message and acknowledge transfer continues until the entire message is transmitted.
- The message is terminated by the master with a STOP condition. This frees the bus for the next master to begin communications.

Both SDA and SCL are bidirectional lines, connected to a positive supply voltage via a current-source or pull-up resistor. When the bus is free, both lines are high. The output stages of devices connected to the bus must have an open-drain or open-collector to

perform the wired-AND function. External pull-up resistors for the open-drain output are required.

Data on the I^{2}C bus can be transferred at rates of up to 100 kbps in the standard mode and up to 400 kbps in the fast mode. The number of interfaces connected to the bus is solely dependent on the bus capacitance limit of 400 pF.

The QSC62x0 implementation of its I^{2}C controller is discussed in the rest of this section.

NOTE For more information about the I^{2}C interface, see the *I^{2}C Controller and I^{2}C Bus Specification in MSM6xxx Devices* (80-V7836-1).

9.6.1 I^{2}C features

- Two-wire bus for inter-IC communication
- Support for external devices fabricated using any process (1.8 V only)
- Support for external functions such as camera sensors, microcontrollers, FM radio ICs, near-field communicator, LCD driver, stereo DAC, and keyboard interface
- Two operating modes with different transfer rates
  - Standard-mode: up to ~100 kbps
  - Fast-mode: up to ~400 kbps

9.6.2 I^{2}C connections

GPIO pins support the I^{2}C implementation; examples are shown in Figure 9-8 for NFC, FM radio, and camera sensor applications. I^{2}C pins are listed in Table 9-8.

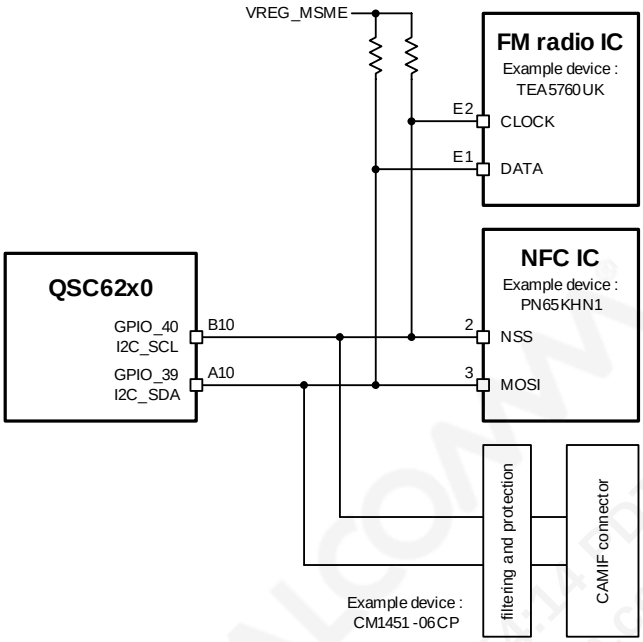


Figure 9-8 Example I²C connections

Table 9-8 I²C connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
I2C_SDA (GPIO_39)	A10	P5 (1.8 V)	B	1-8 (1)	I²C data
I2C_SCL (GPIO_40)	B10	P5 (1.8 V)	B	1-8 (1)	I²C clock

1 The parameters listed under the Pad type column are defined in the QSC6x20 device specification (80-VF846-1)

2 Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

9.6.3 I²C registers

Most of the defined configuration options and status information regarding the I²C operation can be found in the following registers: I2C_WRITE_DATA, I2C_CLK_CTL, I2C_STATUS, I2C_READ_DATA. For more information on these, refer to the *QUALCOMM Single-Chip QSC6240/QSC6270 Software Interface* document (80-VF846-2).

9.6.4 I²C controller

The QSC62x0 I²C controller handles the I²C protocol and frees up the on-chip processor. The controller is an I²C-compliant, high-speed mode (HS-mode) compliant, master-only device. This means the controller can access all available I²C slaves on the bus, but cannot

be accessed by any other masters on the bus. Via the I2C controller, the QSC device is able to interface with such devices as stereo codec (control registers only), external LCD controllers, FM radio ICs, camera sensors, near-field communicator, etc.

QSC62x0 devices support only standard and fast modes.

The I2C interface uses GPIOs that can be configured as open drain outputs. The pull-up resistor for the open drain output is provided at the slave device.

Conceptually, the I2C controller consists of four main blocks: a clock control block, a data control block, a microprocessor block that supervises the controller operation, and a hardware interface block. The clock control and data control functions are discussed in detail below.

9.6.4.1 I2C clock control

Figure 9-9 shows the various possible transitions in the different states of the clock control state machine. Descriptions of the states and their transitions follow after the figure.

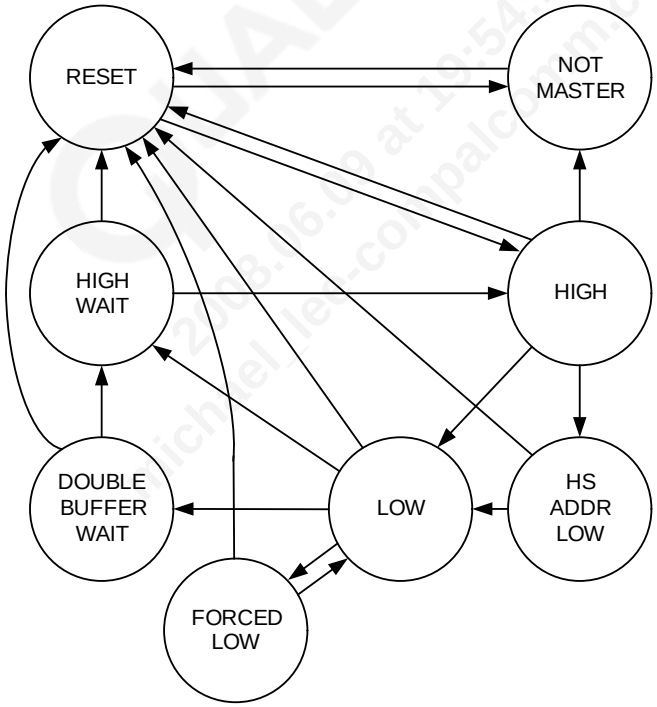


Figure 9-9 Clock control states

Reset bus idle state

There is no activity on the I2C bus when in the RESET_BUSIDLE state. The controller leaves this state when a START condition is detected; START initiates bus transactions and determines the bus master. If sda_out = 0, this controller is the master and transitions to the HIGH state. If sda_out = 1, there is a different master so this controller transitions to the NOT_MASTER state.

The bus should be inactive during this state, but if activity is detected on the bus prior to a START condition (scl_in = 0), an error is generated, the data control block generates an interrupt, and the controller waits until bus activity has ceased.

Not master state

When another master is controlling the I^2C bus, this controller is in the NOT_MASTER state. There are two ways of reaching this state:

- From the RESET_BUSIDLE state, when it is detected that another master has requested control of the bus
- From the HIGH state, when it is detected that this controller has lost arbitration for the bus

Once control of the bus has been relinquished by the other master (STOP condition detected), this controller returns to the RESET_BUSIDLE state.

High state

This is the state that outputs the high phase of scl_out. Four conditions are checked:

- STOP condition: if detected, this controller releases control of the bus and returns to the RESET_BUSIDLE state.
- Loss of arbitration: this controller sets the arb_lost bit, which triggers an interrupt in the data control block, releases control of the bus, transitions to the NOT_MASTER state, and waits for the other controller to release the bus.
- Unexpected START condition: if detected, this controller releases the bus, transitions to the NOT_MASTER state, and flags this as a bus error. There should not be any START conditions detected in this state.
- Toggling to the low phase of scl_out: this can occur after this controller has counted off its high phase (count_zero = 1), or if another master has pulled the bus low. This occurs during arbitration.

In HS mode, a 2:1 low-to-high phase ratio is required. Thus, an extra-low-phase state has been added (HS_ADDR_LOW state). In normal mode, the transition goes to LOW state.

Low state

This state counts off the low phase of scl_out. Once the low phase has been counted off, if the data control block requests that the clock be stalled (force_low = 1), the controller goes into the FORCED_LOW state and waits until the data control block can continue its normal operation, at which time the controller returns to this state and counts off another low period. If there is no need to stall the bus, the controller transitions to the DOUBLE_BUFFER_WAIT state, which extends the low phase of the clock by two i2c_clk cycles. This is done because the double buffering of the serial-in signals causes an extension of the high phase of the clock by two i2c_clk cycles. When in HS mode, double buffering is bypassed so the controller can transition directly to the HIGH_WAIT state.

Double buffer wait state

This state is used to extend the low phase of the clock by two i2c_clk cycles because double buffering the inputs causes the state machine to extend the high phase of the clock's two i2c_clk cycles. Once the two cycles have been counted off, the controller transitions to the HIGH_WAIT state.

High wait state

This state monitors the bus and identifies whether another controller is holding the clock line low. This occurs during arbitration where the other master's natural scl frequency is slower than this controller's frequency. Once it is recognized that the clock line has been released, the controller transitions to the HIGH state to begin counting off its high phase.

Forced low state

This state holds scl_out low for as long as requested by the data control block.

HS address low state

This state counts off an extra low phase of scl_out so that a 2:1 low to high scl_out ratio is achieved, per the I^2C specification.

9.6.4.2 I^2C data control

Figure 9-10 shows the various possible transitions in the different states of the data control state machine. Descriptions of the states and their transitions follow the figure.

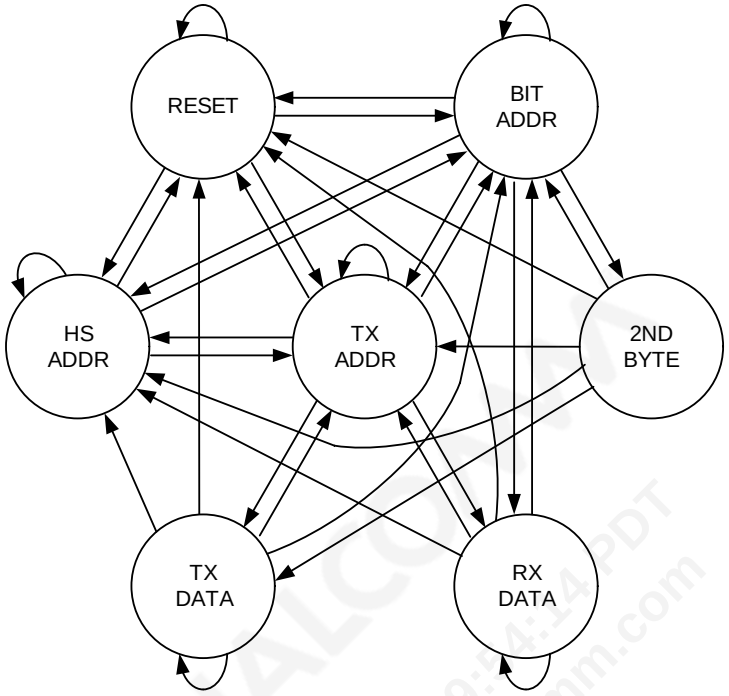


Figure 9-10 Data control states

Reset wait state

There are two main sections to this state:

- Issuing the STOP command to terminate transmissions
- Identifying when the bus is free to commence a transmission

The first section within the code issues the STOP command to the bus. When it is determined (in the previous state) that the STOP command must be issued, the stop signal is set high. Because the transition to the RESET_WAIT state is done during the high clock phase of the ACK cycle, the controller must wait until there is a low phase of the serial clock and then issue the STOP command during the subsequent high phase.

The second section of this block identifies when the controller can attempt to take control of the bus. Data must be present in the buffer, the bus cannot be presently active, and all interrupts must have been serviced (failed_tmp = 0). If any of these conditions are not met, the controller remains in the RESET_WAIT state until it can proceed with a transmission. After all the conditions have been met, the controller decodes the data in the I2C_WRITE_DATA register. Bytes not flagged with addr_byte bit set in this state are considered invalid writes and an interrupt is generated. If the addr_byte bit is set, the address is decoded and the controller proceeds to the appropriate state.

Tx address state

This state transmits the 7-bit I^2C slave address. There are three main sections to this state:

- Force low
- Issue start
- Transmit byte

The first section, where force_low_tmp = 1, occurs after the 7-bit address has been transmitted over the I^2C bus. The controller would enter this section if the 7-bit address was ACKed and there was no new byte buffered, or if the 7-bit address was NACKed, but the byte in buffer is flagged with addr_byte set. The controller forces the serial clock line low (force_low is an output to the clock control block) and waits for the buffer to fill or the interrupt to be serviced, whichever is the case. The next state is dictated by the information in the write buffer: if the addr_byte is not set, then it is data for the accessed slave; if the addr_byte is set, then the address is decoded and the controller selects the appropriate next state.

The second section generates the START condition that precedes any I^2C address on the bus. When transitioning to this state from any other, start is set and this is the first section of code executed.

The third section shifts the data onto the bus. The internal signal WAIT_SIG is used to ensure that events are executed only once per i2c_clk, since the width of scl_in will be a multiple of i2c_clk. Data is updated while the serial clock is low (scl_in = 0); the only action during serial clock high is the reading of ACK (read_ack = 1). At this point, the next transition must be determined.

If the packet is ACKed, the last_byte_bit has precedence overall because it means software has decided to terminate transmission and release control of the bus. Next, the LSB of the 7-bit address byte (rd_wr_n) is checked to allow the controller to release the serial data line, thereby allowing the slave to take control.

If data is to be transmitted (last_byte_bit = 0 and rd_wr_n = 0), the controller checks whether there is data available in the buffer. If not, the controller is set (force_low) and waits until the write buffer is filled (see first section). If the buffer is full, then the next byte is decoded and the appropriate state is selected. If the 7-bit address was NACKed, the controller transitions to the RESET_WAIT state and issues a STOP, unless there is an address byte in the write buffer (wr_buffer_full = 1 and addr_byte = 1), at which time the force_low signal is set and the address byte is not transmitted until the interrupt is processed. The interrupt is generated on the reception of the ACK regardless of its value — this means that a transfer was completed on the bus.

Tx HS address state

This state is very similar to the TX_ADDR state, so only the differences are outlined here.

The most significant difference is that the byte following the transmission of the high-speed master code must be an address, and thus needs to be flagged with the addr_byte set. In the TX_ADDR state, if addr_byte = 0, it produced a transition to TX_DATA state. In this case, if addr_byte = 0 for the next byte due for transmission, it is

flagged as an invalid write and an interrupt is generated to notify software of the error. Also, the high-speed master code must not be ACKed by any slave device. If an ACK occurs, regardless of the next byte in the buffer, this causes an error and the controller transitions to RESET_WAIT state and issues a STOP to release the bus. In the transmit byte section, last_byte and rd_wr_n are not checked.

Tx data state

This state is very similar to the TX_ADDR state, so only the differences are outlined here.

The TX_DATA state does not issue START commands on the bus, so the start = 1 section has been removed. In the transmit byte section this state does not need to examine the rd_wr_n bit; the controller already knows it is writing to the slave.

Tx 10-bit address state

This state varies only slightly from the TX_ADDR state.

In this state, if the byte in the write buffer is not flagged with addr_byte set, the transition is to the TX_2ND_ADDR_BYTE state.

Tx second address byte state

This state (TX_2ND_ADDR_BYTE) is identical to the TX_DATA state. It was added to make it easier to identify when the second byte of the 10-bit address is being sent, thereby making it easier to debug.

Rx data state

In this state, the I^2C is receiving data from the slave. Reception is terminated if either the last_byte is set or the addr_byte is set (which means a new slave is selected for a new transaction, keeping this I^2C controller as bus master).

9.7 Auxiliary PCM

The QSC auxiliary PCM function can be used to interface with the Bluetooth SoC, an external codec, and/or an external stereo DAC. All three applications are discussed below.

9.7.1 Bluetooth PCM interface

In Bluetooth applications using the BTS4020 SoC, a PCM interface is used to transfer audio data between the QSC and BTS devices. An example auxiliary PCM Bluetooth interface is shown in Figure 9-11 and then the connections are listed in Table 9-9.

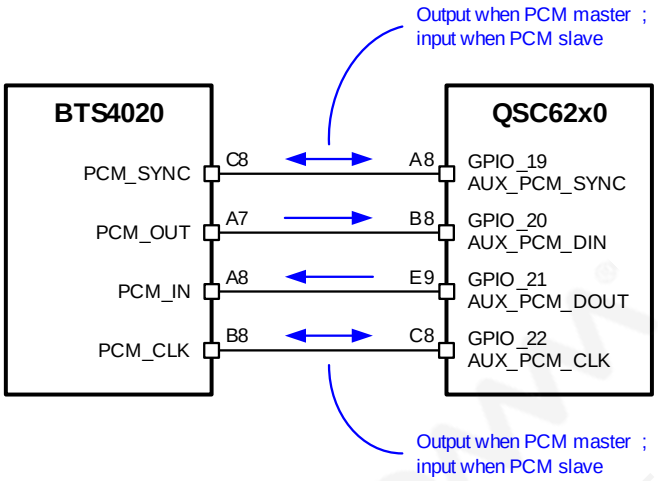


Figure 9-11 Example PCM-Bluetooth interface

Table 9-9 Auxiliary PCM connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
AUX_PCM_DOUT (GPIO_21)	E9	P5 (1.8 V)	B	1-8 (1)	Auxiliary PCM I/F data out (if QSC master)
AUX_PCM_DIN (GPIO_20)	B8	P5 (1.8 V)	B	1-8 (1)	Auxiliary PCM I/F data in (if QSC master)
AUX_PCM_CLK (GPIO_22)	C8	P5 (1.8 V)	B	1-8 (1)	Clock for the auxiliary PCM interface
AUX_PCM_SYNC (GPIO_19)	A8	P5 (1.8 V)	B	1-8 (1)	Auxiliary PCM interface sync

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

Additional details about the BTS-QSC Bluetooth PCM interface is available in the *BTS4020 System-on-Chip (SOC) User Guide*, 80-VE132-3.

9.7.2 External codec

The auxiliary PCM interface enables communication with an external codec to support hands-free applications. Linear, μ-law, and a-law codecs are supported by the auxiliary PCM interface.

The auxiliary codec port operates with standard long-synchronization timing and a 128 kHz clock. The AUX_PCM_SYNC runs at 8 kHz with a 50% duty cycle. Most μ-law and a-law codecs support the 128 kHz AUX_PCM_CLK bit clock.

The ONES_DETECT state machine is clocked by CLKOUT of the internal ARM926EJ-S microprocessor. The logic is controlled by CODEC_CTL:ONES_POLARITY with a single output of WEB_MISC_RD:ONES_DETECT. If ONES_POLARITY = 0, then a high on AUX_DIN guarantees ONES_DETECT = 0. If ONES_POLARITY = 1, then a low on AUX_DIN guarantees ONES_DETECT = 1.

In addition, this logic generates an AUX_PCM_DIN_INT interrupt that is asserted when the value on the AUX_PCM_DIN pin matches that of the ONES_POLARITY bit. This interrupt is enabled when the bit AUX_PCM_DIN_INT_EN within the IRQ_MASK_0 or FIQ_MASK_0 register is set (1); clear (0) this bit to disable the interrupt.

9.7.3 External stereo DAC

The QSC62x0 pins used for the AUX_PCM interface can also be used to interface with an external stereo DAC (SDAC) to play stereo sound or music (MP3 or MIDI, for example). The pertinent pin assignments are listed in Table 9-10. The I2S bus in output mode provides a serial link specifically for digital audio; it handles the transfer of audio data and transports it to the interface.

Using the I2S bus in input mode with AAC is not supported.

Table 9-10 GPIO assignments for AUX_PCM and SDAC interfaces

Pin #	Pin name	AUX_PCM functionality	SDAC interface functionality
A8	GPIO_19	AUX_PCM_SYNC	SDAC_L_R
C8	GPIO_22	AUX_PCM_CLK	SDAC_CLK
B8	GPIO_20	AUX_PCM_DIN	SDAC_MCLK
E9	GPIO_21	AUX_PCM_DOUT	SDAC_DOUT

The stereo DAC has a 16-bit per channel output at several sampling rates from 8 to 48 kHz. The SDAC interface will output PCM data based on the sample rate of the AAC contents. Additional SDAC interconnect details are explained below.

SDAC_DOUT: The serial PCM data stream for both channels are output from the QSC device through this pin. The data is transmitted in the two's complement format with the MSB first. This minimizes the loss of data when the transmitter word length (16 bits) is different from the receiver device's word length, and is handled in one of two ways:

- When the transmitted word length is greater than the receiver word length, the bits after the receiver's LSB are ignored — the rest of the transmitter LSBs are ignored.
- When the transmitted word length is less than the receiver word length, the receiver's missing LSB will be set to zero initially, so they will remain at zero.

These scenarios are true specifically when the system consists of just one transmitter (the master). In a complex system where there are several transmitters, it is possible that the system word length will be greater than a specific transmitter's word length. In that case, the LSBs of the system word will be set to 0 for data transmission.

SDAC_L_R: This signal specifies the present data stream's intended stereo channel: 1 specifies the left channel, 0 specifies the right.

SDAC_CLK: This is the bit clock that can be generated by the QSC device (as explained in the next subsection) and then supplied to the external stereo DAC. Alternatively, it can be generated by the external stereo DAC and then provided to the QSC device.

The SDAC_CLK frequency is dependent on the number of bits per channel and the selected sampling rate. For example, for two channels, 16 bits each, and a chosen 48 kHz sampling frequency, the frequency would be:

FSDAC_CLK = (16 + 16) * 48k = 1536000 b/s = 1.536 MHz

Table 9-11 provides example SDAC_CLK frequencies for a 2-channel, 16-bit output at various sampling rates.

Table 9-11 Sampling frequency and SDAC_CLK rates

Sampling frequency	Required SDAC_CLK rate
32 kHz	1.024 MHz
44.1 kHz	1.4112 MHz
48 kHz	1.536 MHz

NOTE The I2S bus supports a maximum frequency of 12.888 MHz.

SDAC_MCLK: An optional clock output from the QSC device to the external stereo DAC. This clock does not affect the QSC device; it merely provides handset designers with the option of using one of the QSC outputs to provide the external device with a master clock signal (MCLK).

If the QSC62x0 device is chosen as the clock source for the SDAC_CLK, the SDAC_CLK signal can be configured using bits [4:3] of the MSM_CLK_SRCSEL3 register. An external clock source, a 1.024 MHz clock, or an M/N:D counter output can be selected. For the M/N:D counter output, the SDAC_MNCNTR_NOT_N_M, SDAC_MNCNTR_M, and SDAC_MNCNTR_D registers are used to set its configuration. The M/N:D counter can be clocked from the 19.2 MHz TCXO or 384 MHz PLL clock sources; each source is divided down to the desired frequency by configuring appropriate M, N, and D values.

When using the M/N:D counter, the M/N is the division ratio and the M and N values must be programmed correctly. The D value controls the output duty cycle. For a duty cycle of 50%, D should equal N/2. Using the M/N:D counter has limitations and can cause mild-to-serious jitter on the clock signal. In order to guarantee minimal jitter, integer divisors should be used.

For convenience, a 1.024 MHz clock is available from a 2.048 MHz source divided by two; in this case the M/N:D counter is bypassed.

To ensure proper implementation and adequate output clock quality, please contact QUALCOMM for all M/N ratios.

9.8 SPI

When the SPI (master only) system is enabled by setting the control register's SPI enable (SPE) bit, the SPI block provides a duplex, synchronous, serial communication link with the QSC baseband circuits (controller or master). Software can poll the SPI status flags or operation can be interrupt driven.

9.8.1 SPI features

- Master support
- Serial clock with programmable polarity and phase
- Clock rates up to 26 MHz
- One chip-select
- 1.8 V operation

9.8.2 SPI connections (master only)

QSC62x0 pin assignments for SPI connections are listed in Table 9-12.

Table 9-12 Serial peripheral interface connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
SPI_CLK (GPIO_53)	E8	P2 (1.8/2.6 V)	B	(5.6/11.2)	SPI clock
SPI_CS_N (GPIO_52)	F8	P2 (1.8/2.6 V)	B	(5.6/11.2)	SPI chip-select; not mandatory in a point-to-point connection
SPI_MOSI_DATA (GPIO_51)	F7	P2 (1.8/2.6 V)	B	(5.6/11.2)	SPI master out/slave in data
SPI_MISO_DATA (GPIO_50)	E7	P2 (1.8/2.6 V)	B	(5.6/11.2)	SPI master in/slave out data

1 The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

2 Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

9.8.3 SPI configurations

The QSC SPI must be configured as the master.

Master configuration

The SPI is configured as the master when the MSTR bit is set. Only a master SPI module can initiate transmissions. A transmission begins by writing to the master SPI data register. The data begins shifting out on the SDO/MOSI pin under the control of the serial clock.

9.8.4 SPI operating modes

The SPI operates in three modes:

- Run mode – the basic mode of operation.
- Wait mode – SPI operation in wait mode is a configurable low-power mode that is enabled by the control register's wait bit. During wait mode, if the SPI wait bit is clear the SPI operates similarly to the run mode. But if the SPI wait bit is set, the SPI goes into a power conservative state with the SPI clock generation turned off.
- Stop mode – The SPI is inactive in stop mode for reduced power consumption. If the SPI is configured as a master, any transmission in progress stops, but resumes after the controller returns to run mode.

9.9 NFC

The QSC62x0 device supports NFC through four GPIO pins that interface with a NFC IC. An example NFC interface is shown in Figure 9-12 and then the connections are listed in Table 9-13.

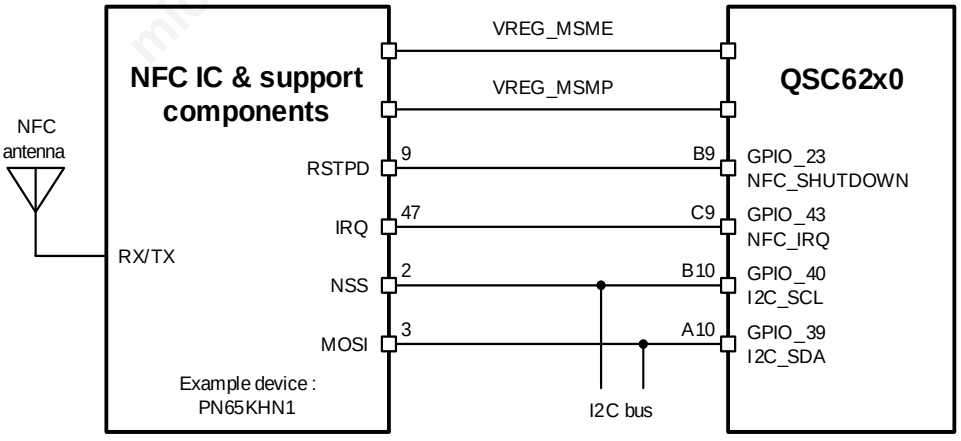


Figure 9-12 NFC interface

Table 9-13 NFC connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description [3]
NFC_IRQ (GPIO_43)	C9	P5 (1.8 V)	DI	–	Allows NFC to interrupt the QSC
NFC_SHUTDOWN (GPIO_23)	B9	P5 (1.8 V)	DO	1-8 (1)	Powerdown signal to NFC device
I2C_SDA (GPIO_39)	A10	P5 (1.8 V)	B	1-8 (1)	I^2C data for NFC
I2C_SCL (GPIO_40)	B10	P5 (1.8 V)	B	1-8 (1)	I^2C clock for NFC

1 The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

2 Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

3 An extra GPIO is needed to support handshake mode. NFC devices other than PN65KHN1 may not require a GPIO for handshake mode.

9.10 Bluetooth

Bluetooth technology provides short-range wireless connections between mobile phones, computers, personal digital assistants (PDAs), and other such devices, allowing users to coordinate the activities of all these devices within range of their piconet. A piconet is established automatically as Bluetooth-enabled devices enter and leave radio proximity.

A Bluetooth piconet can handle both data and voice communications simultaneously. This enables mobile phone applications such as a hands-free headset for voice calls, printing and fax capabilities, and synchronized operation with a PDA or laptop. An automobile hands-free system allows the driver to make mobile phone calls directly through the car's speakers while keeping both hands on the steering wheel. Another popular phone feature is Internet access, even when the user is out of range of a hotspot or wired connection. This extended hot spot is available wherever the phone has access to its networks.

A functional block diagram showing the BTS4020 solution is shown in Figure 9-13.

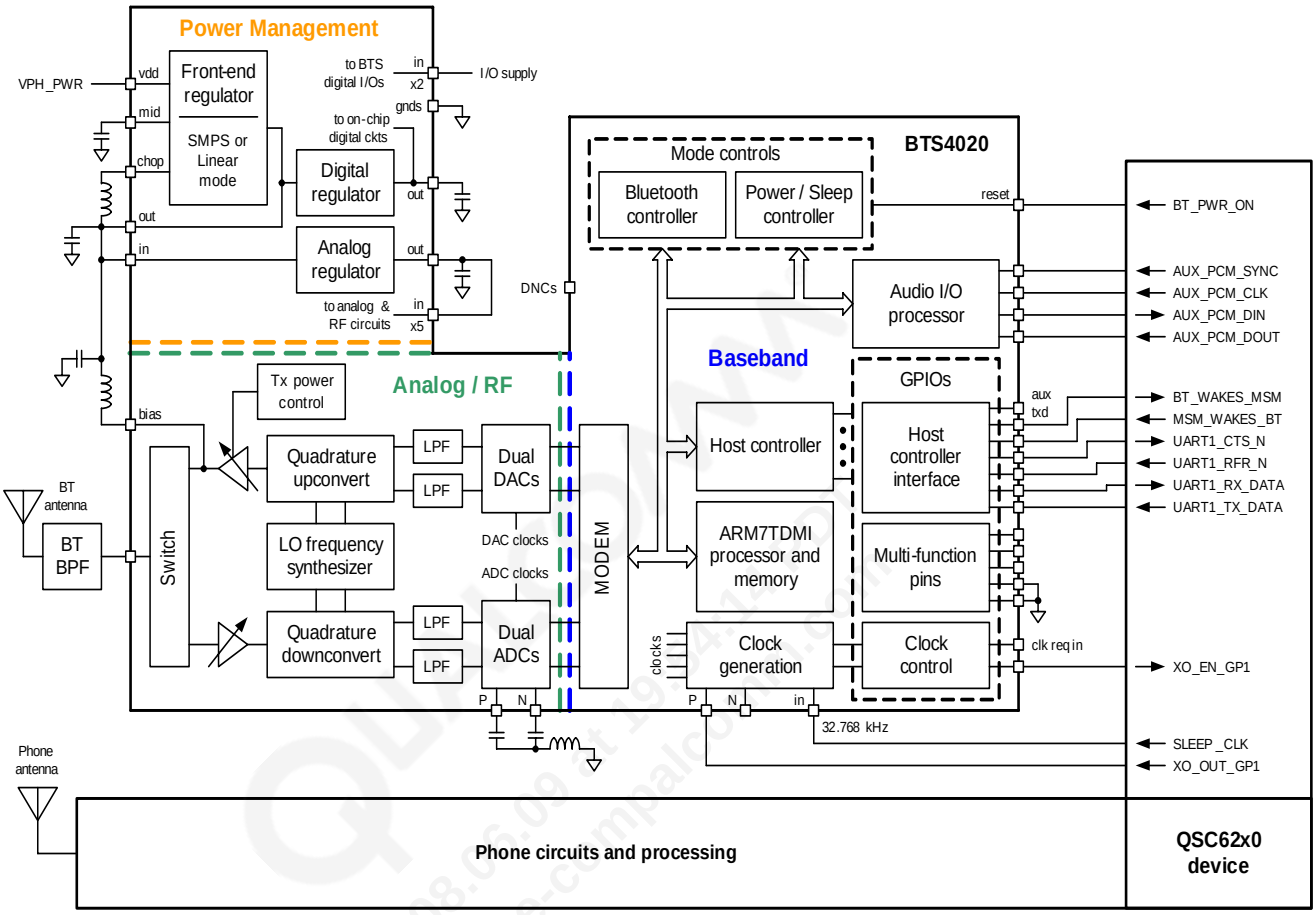


Figure 9-13 Example BTS4020 + QSC62x0 Bluetooth functional block diagram

This application is described more completely in the *BTS4020 System-on-Chip (SoC) Design Guidelines* document (80-VE132-5); some high-level comments follow:

- The handset includes a dedicated Bluetooth antenna; it is connected to a band-select filter that suppresses Rx out-of-band interference and Tx out-of-band spurious emissions while passing the desired Bluetooth signals.
- BTS RF circuits are independent of the phone RF circuits, even though they share the same processor (in this case, the QSC62x0 device).
- Rx and Tx data is transferred between the BTS4020 IC and the QSC device using a UART interface.
- The BTS4020 IC provides a serial interface to support PCM formatted data. The PCM interface exchanges data with any compatible device, such as a codec, including the QSC device. The BTS device can serve as PCM bus master or slave.
- Dedicated one-to-one BTS/QSC connections supplement status and control signaling.
- The QSC device provides the BTS4020 IC's master clock (19.2 MHz XO_OUT_GP1) and sleep clock (32.768 kHz SLEEP_CLK).

- The BTS4020 IC includes integrated power management (PM) functions that allow a wide variety of DC power-supply implementations. The example uses the following on-chip PM functions:
  - The phone's primary power supply voltage (VPH_PWR) powers the BTS IC's front-end regulator (in this case, operating in its SMPS mode).
  - The front-end regulator output is connected internally to a digital supply regulator and externally to an analog supply regulator.
  - The digital and analog regulators provide supply voltages to their corresponding on-chip circuits. The digital circuits are connected internally; analog externally.

A complete set of BTS4020 documents are available at the CDMA Tech Support website (https://support.cdmatech.com), including:

- *BTS4020 System-on-Chip (SoC) Device Specification* (80-VE132-1)
- *BTS4020 System-on-Chip (SoC) User Guide* (80-VE132-3)
- *BTS4020 System-on-Chip (SoC) Revision Guide* (80-VE132-4)
- *BTS4020 Bluetooth System-on-Chip (SoC) Design Guidelines*, (80-VE132-5)
- *BTS4020 HCI Vendor Specific Interface Control Document*, (80-VE132-7)
- *BTS4020 NVM Configuration Parameters Application Note*, (80-VE132-13)

9.10.1 Bluetooth features

The QSC62x0 plus BTS4020 combination is compliant with the Bluetooth Specification Version 2.0 + EDR, and is fully backward compatible with Bluetooth Specification Versions 1.1 and 1.2. In addition, the following features are supported:

- General features
  - Enhanced data rate (EDR) support for up to 3 Mbps
  - As few as nine external components (plus the BTS4020 SoC) in a minimum configuration
  - RoHS compliant
  - Supported device connections:
    - One SCO or eSCO connection
    - Up to seven total connections
    - Up to 3.5 piconets (master, slave, and page scanning)
- Analog / RF features of the BTS4020 SoC
  - 2.4 GHz Bluetooth radio
  - Excellent blocking and low spurious emissions in cellular frequencies
  - No RF tuning required in production

- Baseband features of the BTS4020 SoC
  - High-performance, highly efficient ARM7TDMI processor
  - Host Controller Interface (HCI) support through UART, SPI (master only), and SDIO
    - UART autobaud detection
  - PCM interface for peripherals such as codecs
  - Integrated packet traffic arbitration (PTA) support for coexistence with 802.11b/g
  - Bluetooth wake-up interval can be synchronized with the host wake-up interval
  - Direct reference clock input with frequency auto selection, supports frequencies from 12 to 48 MHz
  - 16 ROM patches available for firmware update without hardware change
- Power management features of the BTS4020 SoC
  - Integrated front-end regulator for direct battery connection
  - Analog regulator for on-chip and off-chip RF circuits
  - Digital regulator for on-chip digital circuits
  - Precision on-chip voltage reference
  - 1.2 V digital core, 1.45 V analog core, and 1.62 to 3.63 V peripheral

9.10.2 Bluetooth connections

The Bluetooth interface between the BTS4020 SoC and the QSC62x0 device is shown in Figure 9-14 and then the connections are listed in Table 9-14.

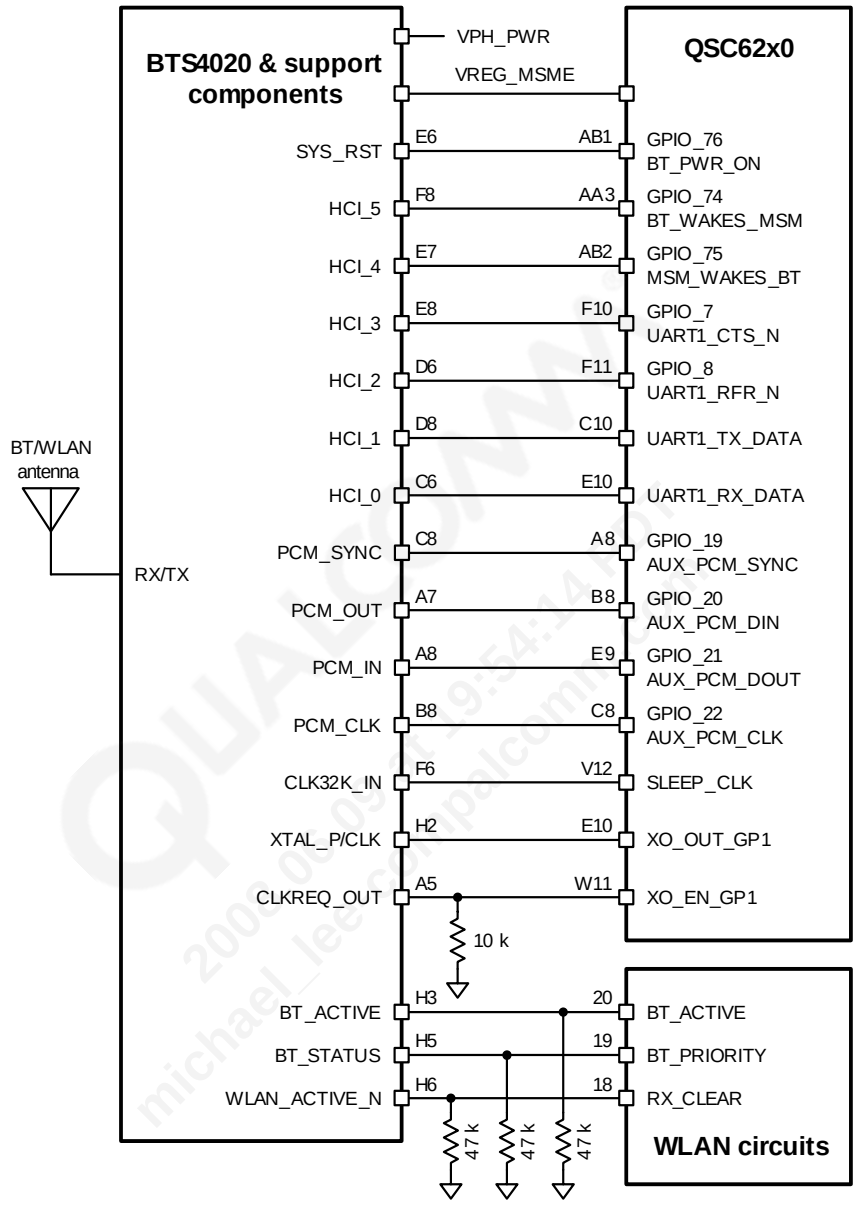


Figure 9-14 Bluetooth interface (BTS4020 example)

Bluetooth and WLAN coexistence is supported by the connections shown at the bottom of Figure 9-14.

Table 9-14 Bluetooth connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
BT_PWR_ON (GPIO_76)	AB1	P1 (1.8 V)	DO	1-8 (1)	Poweron signal for the Bluetooth device
BT_WAKES_MSM (GPIO_74)	AA3	P1 (1.8 V)	DI	–	BT signal to wake up the MSM (QSC)
MSM_WAKES_BT (GPIO_75)	AB2	P1 (1.8 V)	DO	1-8 (1)	Signal to wake up the Bluetooth device
XO_EN_GP1	W11	–	DO	–	Clock request from Bluetooth device
XO_OUT_GP1	W10	–	DO	–	Master clock to Bluetooth device
SLEEP_CLK	V12	–	DO	–	Sleep clock to Bluetooth device
UART1_RXD	E10	P5 (1.8 V)	DI (HV)	–	HS UART receive data from BT device
UART1_TXD	C10	P5 (1.8 V)	DO	1-8 (1)	HS UART transmit data to BT device
UART1_RFR_N (GPIO_8)	F11	P5 (1.8 V)	DO	1-8 (1)	HS UART ready for receive to BT device
UART1_CTS_N (GPIO_7)	F10	P5 (1.8 V)	DI	–	HS UART clear to send from BT device
AUX_PCM_DOUT (GPIO_21)	E9	P5 (1.8 V)	B	1-8 (1)	Bluetooth PCM I/F data out (if QSC master)
AUX_PCM_DIN (GPIO_20)	B8	P5 (1.8 V)	B	1-8 (1)	Bluetooth PCM I/F data in (if QSC master)
AUX_PCM_CLK (GPIO_22)	C8	P5 (1.8 V)	B	1-8 (1)	Clock for Bluetooth PCM interface
AUX_PCM_SYNC (GPIO_19)	A8	P5 (1.8 V)	B	1-8 (1)	Bluetooth PCM interface sync

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

9.11 FM radio

The QSC62x0 device supports FM radio using six pins – three dedicated lines and three GPIOs. An example FM radio interface is shown in Figure 9-15 and then the connections are listed in Table 9-15.

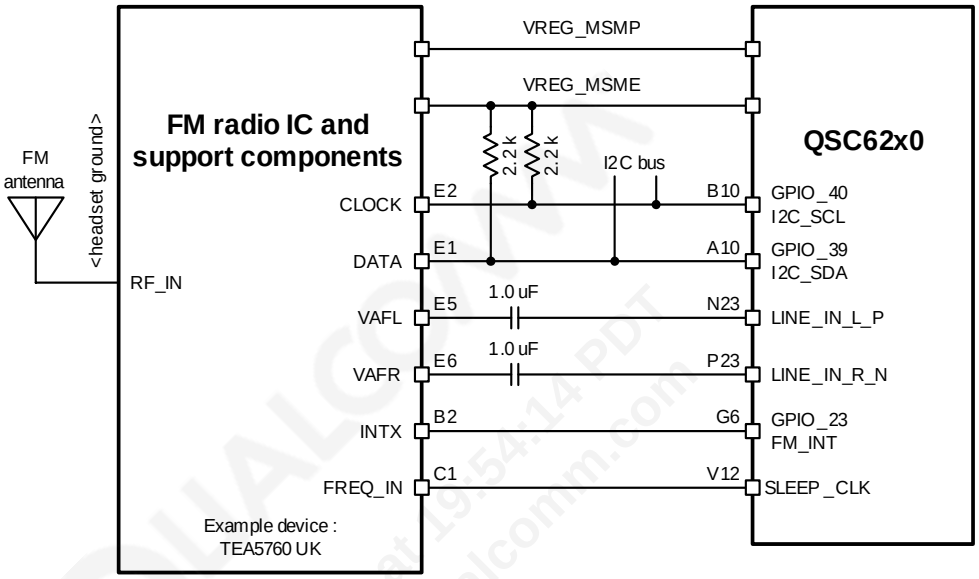


Figure 9-15 FM radio connections

Table 9-15 FM radio connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
FM_INT (GPIO_23)	G6	P2 (1.8/2.6 V)	DI	–	Allows the FM radio to interrupt the QSC
I2C_SDA (GPIO_39)	A10	P5 (1.8 V)	B	1-8 (1)	I^2C data for FM radio
I2C_SCL (GPIO_40)	B10	P5 (1.8 V)	B	1-8 (1)	I^2C clock for FM radio
LINE_IN_R_N	P23	–	AI	–	FM_AUDIO_M
LINE_IN_L_P	N23	–	AI	–	FM_AUDIO_P
SLEEP_CLK	V12	–	DO	–	Sleep clock to FM radio IC

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

9.12 WLAN

The QSC62x0 device supports WLAN applications using eight properly configured GPIO pins. An example QSC interface with a WLAN SoC is shown in Figure 9-16 and then the connections are listed in Table 9-16.

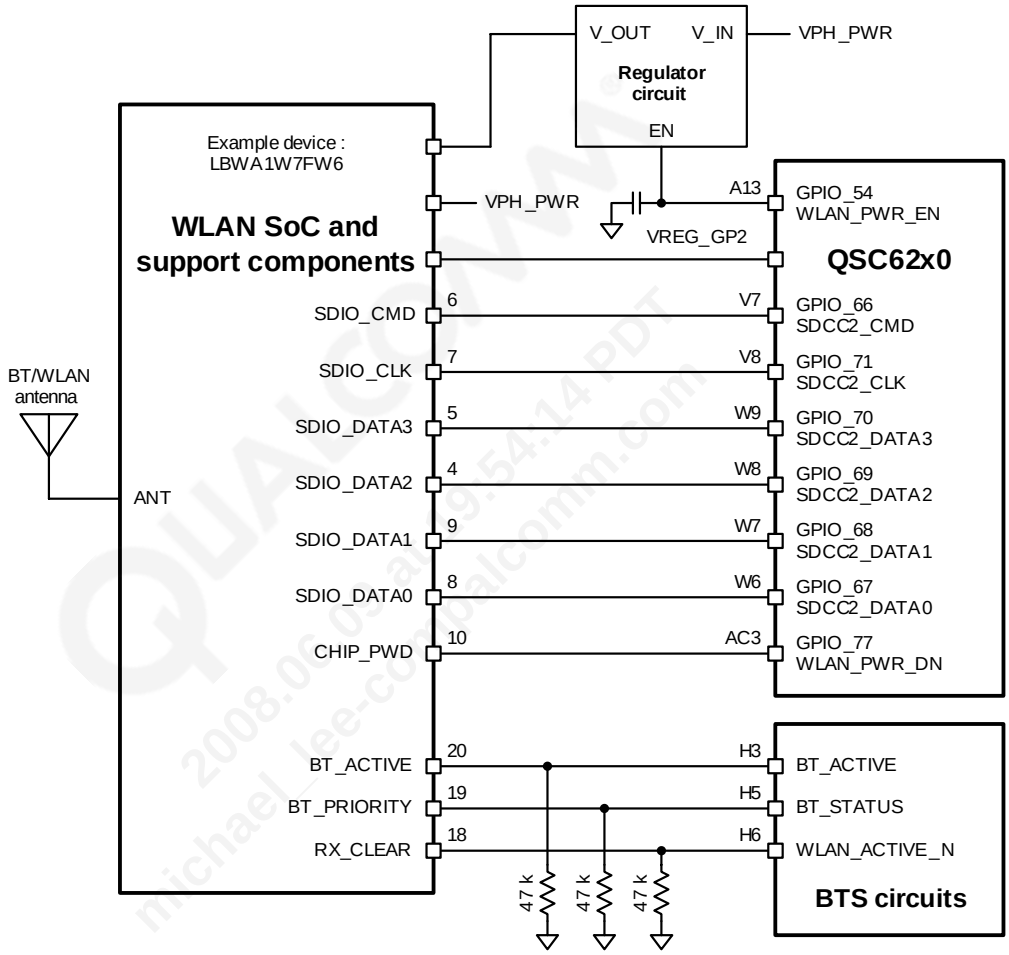


Figure 9-16 WLAN connections

Table 9-16 WLAN connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
WLAN_PWR_EN (GPIO_54)	A13	P5 (1.8 V)	DO	1-8 (1)	Enables power to the WLAN device
WLAN_PWR_DN (GPIO_77)	AC3	P7 (2.6 V)	DO	2-16 (2)	Powerdown signal to the WLAN device
SDCC2_CLK (GPIO_71)	V8	P7 (2.6 V)	DO	2-16 (8)	Output clock for WLAN device
SDCC2_DATA3 (GPIO_70)	W9	P7 (2.6 V)	B (HV)	2-16 (2)	WLAN data bit 3
SDCC2_DATA2 (GPIO_69)	W8	P7 (2.6 V)	B (HV)	2-16 (2)	WLAN data bit 2
SDCC2_DATA1 (GPIO_68)	W7	P7 (2.6 V)	B (HV)	2-16 (2)	WLAN data bit 1
SDCC2_DATA0 (GPIO_67)	W6	P7 (2.6 V)	B (HV)	2-16 (2)	WLAN data bit 0
SDCC2_CMD (GPIO_66)	V7	P7 (2.6 V)	B (HV)	2-16 (2)	WLAN command bit

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

10 General-purpose Input/Output Pins

The QSC62x0 device provides 78 GPIO pins that can be configured through software; all GPIO pins are listed in descending order in Table 10-1. Various GPIO pins allow different configuration combinations depending on their intended functions. This is shown in the Pad type column, which is decoded as follows:

- B: bidirectional; these can be configured as inputs, outputs, or bidirectional.
- K: indicates an internal weak keeper device (keepers cannot drive external buses).
- HV: a high-voltage tolerant digital input that allows input voltages as high as 3.0 V.
- S: Schmitt trigger
- nppdpuk: programmable pull resistor. The default pull direction is indicated by capital letters:
  - NPpdpuk = default no pull with other programmable options shown as lower case
  - npPDpuk = default pull down with other programmable options shown as lower case
  - nppdPUk = default pull up with other programmable options shown as lower case
  - nppdpuK = default keeper with other programmable options shown as lower case

Table 10-1 General purpose input/output (GPIO) pins

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
GPIO_77 WLAN_PWR_DN	AC3	P7 (2.6 V)	BS-nppdPUk DO	2-16 (2) (2)	Configurable I/O Powerdown signal for the WLAN device
GPIO_76 BT_PWR_ON	AB1	P1 (1.8 V)	BS-npPDpuk DO	1-8 (1) (1)	Configurable I/O Power-on signal for the Bluetooth device
GPIO_75 MSM_WAKES_BT	AB2	P1 (1.8 V)	BS-npPDpuk DO	1-8 (1) (1)	Configurable I/O Signal to wake up the Bluetooth device
GPIO_74 BT_WAKES_MSM	AA3	P1 (1.8 V)	BS-npPDpuk DI	1-8 (1) –	Configurable I/O BT signal to wake up the MSM (QSC)
GPIO_73 MUSIM_DP	AB8	P3 (1.8/2.85 V)	BS-npPDpuk DI, DO	1-8/2-16 (6/12) (6/12)	Configurable I/O USB-UICC data plus line
GPIO_72 MUSIM_DM	AA7	P3 (1.8/2.85 V)	BS-npPDpuk DI, DO	1-8/2-16 (6/12) (6/12)	Configurable I/O USB-UICC data minus line
GPIO_71 SDCC2_CLK	V8	P7 (2.6 V)	BS-npPDpuk DO	2-16 (8) (8)	Configurable I/O Output clock for SDIO2 device

Table 10-1 General purpose input/output (GPIO) pins (continued)

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
GPIO_70 SDCC2_DATA3	W9	P7 (2.6 V)	BS-npPDpuk DI, DO	2-16 (2) (2)	Configurable I/O SDIO2 device data bit 3
GPIO_69 SDCC2_DATA2	W8	P7 (2.6 V)	BS-npPDpuk DI, DO	2-16 (2) (2)	Configurable I/O SDIO2 device data bit 2
GPIO_68 SDCC2_DATA1	W7	P7 (2.6 V)	BS-npPDpuk DI, DO	2-16 (2) (2)	Configurable I/O SDIO2 device data bit 1
GPIO_67 SDCC2_DATA0	W6	P7 (2.6 V)	BS-npPDpuk DI, DO	2-16 (2) (2)	Configurable I/O SDIO2 device data bit 0
GPIO_66 SDCC2_CMD	V7	P7 (2.6 V)	BS-npPDpuk DI, DO	2-16 (2) (2)	Configurable I/O SDIO2 device command bit
GPIO_65 TRK_LO_ADJ GP_CLK GP_PDM_1	T14	P6 (1.8/2.6 V)	BS-npPDpuk DO DO DO	1-8/2-16 (1/2) (1/2) (1/2) (1/2)	Configurable I/O Backup VCTCXO PDM off pad group 6 Clock for FM radio, camera focus, etc. PDM for AFC, PA bias voltage, etc.
GPIO_64 PA_RANGE0	C17	P6 (1.8/2.6 V)	BS-npPDpuk DO	1-8/2-16 (1/2) (1/2)	Configurable I/O Controls UMTS PA output power range
GPIO_63 PA_RANGE1	C16	P6 (1.8/2.6 V)	BS-npPDpuk DO	1-8/2-16 (1/2) (1/2)	Configurable I/O Controls UMTS PA output power range
GPIO_62 ANT_SEL_0 BOOT_FROM_ROM	A16	P6 (1.8/2.6 V)	BS-npPDpuk DO DI	1-8/2-16 (1/2) (1/2) –	Configurable I/O Antenna switch control bit 0 Boot control; security fuse can override
GPIO_61 ANT_SEL_1 WDOG_DISABLE FAILED_BOOT	B16	P6 (1.8/2.6 V)	BS-npPDpuk DO DI DO	1-8/2-16 (1/2) (1/2) – (1/2)	Configurable I/O Antenna switch control bit 1 Disables watchdog timer Indicates boot failure type via toggle rate
GPIO_60 ANT_SEL_2	A17	P6 (1.8/2.6 V)	BS-npPDpuk DO	1-8/2-16 (1/2) (1/2)	Configurable I/O Antenna switch control bit 2
GPIO_59 ANT_SEL_3	B17	P6 (1.8/2.6 V)	BS-npPDpuk DO	1-8/2-16 (1/2) (1/2)	Configurable I/O Antenna switch control bit 3
GPIO_58 GPS_ADCQ	E15	P5 (1.8 V)	BS-npPDpuk DI	1-8 (1) –	Configurable I/O GPS Rx data Q bit from RGR6240 IC
GPIO_57 GPS_ADCCLK	B15	P5 (1.8 V)	BS-npPDpuk DO	1-8 (1) (1)	Configurable I/O GPS ADC sampling clock to RGR6240 IC
GPIO_56 GPS_SSBI	A14	P5 (1.8 V)	BS-npPDpuk DI, DO	1-8 (1) (1)	Configurable I/O SSBI for GPS status & control (RGR6240)
GPIO_55 GPS_ADCI	F14	P5 (1.8 V)	BS-npPDpuk DI	1-8 (1) –	Configurable I/O GPS Rx data I bit from RGR6240 IC
GPIO_54 WLAN_PWR_EN	A13	P5 (1.8 V)	BS-npPDpuk DO	1-8 (1) (1)	Configurable I/O Enables power to the WLAN device

Table 10-1 General purpose input/output (GPIO) pins (continued)

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
GPIO_53 SPI_CLK	E8	P2 (1.8/2.6 V)	BS-npPDpuk DI, DO	 (5.6/11.2)	Configurable I/O Serial peripheral interface clock
GPIO_52 SPI_CS_N	F8	P2 (1.8/2.6 V)	BS-nppdPUk DI, DO	 (5.6/11.2)	Configurable I/O Serial peripheral interface chip-select
GPIO_51 SPI_MOSI_DATA	F7	P2 (1.8/2.6 V)	BS-npPDpuk DI, DO	 (5.6/11.2)	Configurable I/O SPI master out / slave in data
GPIO_50 SPI_MISO_DATA	E7	P2 (1.8/2.6 V)	BS-npPDpuk DI, DO	 (5.6/11.2)	Configurable I/O SPI master in / slave out data
GPIO_49 EBI2_CS2_N	A7	P2 (1.8/2.6 V)	BS-nppdPUk DO	 (5.6/11.2)	Configurable I/O EBI2 chip-select bit 2
GPIO_48 EBI2_BUSY_CS3_N	B7	P2 (1.8/2.6 V)	BS-nppdPUk DI, DO	 (5.6/11.2)	Configurable I/O EBI2 chip-select bit 3 or busy for NAND
GPIO_47 USIM_DATA	AC5	P3 (1.8/2.85 V)	BS-npPDpuk DI, DO	1-8/2-16 (6/12) (6/12)	Configurable I/O USIM data
GPIO_46 USIM_CLK	AB6	P3 (1.8/2.85 V)	BS-npPDpuk DO	1-8/2-16 (6/12) (6/12)	Configurable I/O USIM clock
GPIO_45 USIM_RESET	AC4	P3 (1.8/2.85 V)	BS-npPDpuk DO	1-8/2-16 (1/2) (1/2)	Configurable I/O USIM reset
GPIO_44 ETM_MODE_CS_N	H10	P5 (1.8 V)	BS-npPDpuk Z	1-8 (1) –	Configurable I/O ETM chip-select
GPIO_43 NFC_IRQ ETM_PIPESTATA2	C9	P5 (1.8 V)	BS-nppdPUk DI DO	1-8 (1) – (1)	Configurable I/O Allows NFC to interrupt the QSC ETM pipe A status bit 2
GPIO_42 NFC_SHUTDOWN GP_CLK ETM_MODE_KYSNS_INT	B9	P5 (1.8 V)	BS-nppdPUk DO DO DI	1-8 (1) (1) (1) –	Configurable I/O Powerdown signal to NFC device General-purpose clock output ETM interrupt - alerts QSC of a key press
GPIO_41 HEADSET_DET_N GP_CLK ETM_TRACESYNC_A	H12	P5 (1.8 V)	BS-nppdPUk DI DO DO	1-8 (1) – (1) (1)	Configurable I/O Detects headset mechanical switch General-purpose clock output Trace packet A sync
GPIO_40 I2C_SCL	B10	P5 (1.8 V)	BS-nppdPUk DI, DO	1-8 (1) (1)	Configurable I/O I^2C clock for camera, BT, FM radio, etc.
GPIO_39 I2C_SDA	A10	P5 (1.8 V)	BS-nppdPUk DI, DO	1-8 (1) (1)	Configurable I/O I^2C data for camera, BT, FM radio, etc.
GPIO_38 CAMIF_MCLK	A11	P5 (1.8 V)	BS-npPDpuk DO	1-8 (1) (1)	Configurable I/O Master clock to the camera sensor
GPIO_37 CAMIF_DATA_9 ETM_TRACE_PKTB7	C14	P5 (1.8 V)	BS-npPDpuk DI DO	1-8 (1) – (1)	Configurable I/O Camera I/F pixel data bit 9 ETM trace packet B bit 7

Table 10-1 General purpose input/output (GPIO) pins (continued)

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
GPIO_36	B14	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_DATA_8		(1.8 V)	DI	–	Camera I/F pixel data bit 8
ETM_TRACE_PKTB6			DO	(1)	ETM trace packet B bit 6
GPIO_35	F13	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_DATA_7		(1.8 V)	DI	–	Camera I/F pixel data bit 7
ETM_TRACE_PKTB5			DO	(1)	ETM trace packet B bit 5
GPIO_34	B13	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_DATA_6		(1.8 V)	DI	–	Camera I/F pixel data bit 6
ETM_TRACE_PKTB4			DO	(1)	ETM trace packet B bit 4
GPIO_33	C13	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_DATA_5		(1.8 V)	DI	–	Camera I/F pixel data bit 5
ETM_TRACE_PKTB3			DO	(1)	ETM trace packet B bit 3
GPIO_32	F12	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_DATA_4		(1.8 V)	DI	–	Camera I/F pixel data bit 4
ETM_TRACE_PKTB2			DO	(1)	ETM trace packet B bit 2
GPIO_31	B12	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_DATA_3		(1.8 V)	DI	–	Camera I/F pixel data bit 3
ETM_TRACE_PKTB1			DO	(1)	ETM trace packet B bit 1
GPIO_30	C12	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_DATA_2		(1.8 V)	DI	–	Camera I/F pixel data bit 2
ETM_TRACE_PKTB0			DO	(1)	ETM trace packet B bit 0
GPIO_29	E13	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_DATA_1		(1.8 V)	DI	–	Camera I/F pixel data bit 1
ETM_PIPESTATB0			DO	(1)	ETM pipe B status bit 0
GPIO_28	B11	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_DATA_0		(1.8 V)	DI	–	Camera I/F pixel data bit 0
ETM_PIPESTATB1			DO	(1)	ETM pipe B status bit 1
GPIO_27	E12	P5	BS-nppdPUk	1-8 (1)	Configurable I/O
CAMIF_DISABLE		(1.8 V)	DO	(1)	Turns off the camera
ETM_MODE_INT			DI	–	Interrupts ETM mode
GPIO_26	E11	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_VSYNC		(1.8 V)	DI	–	Vertical sync from the camera
ETM_TRACESYNCB			DO	(1)	ETM trace B sync
GPIO_25	C11	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_HSYNC		(1.8 V)	DI	–	Horizontal sync from the camera
ETM_PIPESTATB2			DO	(1)	ETM pipe B status bit 2
GPIO_24	E14	P5	BS-npPDpuk	1-8 (1)	Configurable I/O
CAMIF_PCLK		(1.8 V)	DI	–	Pixel clock from camera
GPIO_23	G6	P2	BS-npPDpuk	1-8/2-16 (1/2)	Configurable I/O
MDP_VSYNC_P		(1.8/2.6 V)	DI	–	Vertical sync to/from MDP
FM_INT			DI	–	Allows the FM radio to interrupt the QSC

Table 10-1 General purpose input/output (GPIO) pins (continued)

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
GPIO_22 AUX_PCM_CLK	C8	P5 (1.8 V)	BS-npPDpuk DI, DO	1-8 (1) (1)	Configurable I/O Clock for the auxiliary PCM interface
GPIO_21 AUX_PCM_DOUT	E9	P5 (1.8 V)	BS-npPDpuk DI, DO	1-8 (1) (1)	Configurable I/O Auxiliary PCM I/F data out (if QSC master)
GPIO_20 AUX_PCM_DIN	B8	P5 (1.8 V)	BS-nppdPUk DI, DO	1-8 (1) (1)	Configurable I/O Auxiliary PCM I/F data in (if QSC master)
GPIO_19 AUX_PCM_SYNC	A8	P5 (1.8 V)	BS-npPDpuk DI, DO	1-8 (1) (1)	Configurable I/O Auxiliary PCM interface sync
GPIO_18 KEYSENSE0_N ETM_TRACE_PKTA7	W1	P1 (1.8 V)	BS-nppdPUk DI DO	1-8 (1) – (1)	Configurable I/O Bit 0 for sensing key press on pad matrix Trace packet A bit 7
GPIO_17 KEYSENSE1_N ETM_TRACE_PKTA6	V1	P1 (1.8 V)	BS-nppdPUk DI DO	1-8 (1) – (1)	Configurable I/O Bit 1 for sensing key press on pad matrix Trace packet A bit 6
GPIO_16 KEYSENSE2_N ETM_TRACE_PKTA5	V2	P1 (1.8 V)	BS-nppdPUk DI DO	1-8 (1) – (1)	Configurable I/O Bit 2 for sensing key press on pad matrix Trace packet A bit 5
GPIO_15 KEYSENSE3_N ETM_TRACE_PKTA4	W2	P1 (1.8 V)	BS-nppdPUk DI DO	1-8 (1) – (1)	Configurable I/O Bit 3 for sensing key press on pad matrix Trace packet A bit 4
GPIO_14 KEYSENSE4_N ETM_TRACE_PKTA3	W3	P1 (1.8 V)	BS-nppdPUk DI DO	1-8 (1) – (1)	Configurable I/O Bit 4 for sensing key press on pad matrix Trace packet A bit 3
GPIO_13 KEYPAD_0 ETM_TRACE_PKTA2	Y1	P1 (1.8 V)	BS-nppdPUk DO DO	1-8 (1) (1) (1)	Configurable I/O Bit 0 drive to the pad matrix Trace packet A bit 2
GPIO_12 KEYPAD_1 ETM_TRACE_PKTA1	Y2	P1 (1.8 V)	BS-nppdPUk DO DO	1-8 (1) (1) (1)	Configurable I/O Bit 1 drive to the pad matrix Trace packet A bit 1
GPIO_11 KEYPAD_2 ETM_TRACE_PKTA0	AA1	P1 (1.8 V)	BS-nppdPUk DO DO	1-8 (1) (1) (1)	Configurable I/O Bit 2 drive to the pad matrix Trace packet A bit 0
GPIO_10 KEYPAD_3 ETM_PIPESTATA0	Y3	P1 (1.8 V)	BS-nppdPUk DO DO	1-8 (1) (1) (1)	Configurable I/O Bit 3 drive to the pad matrix EBM pipe A status bit 0
GPIO_9 KEYPAD_4 ETM_PIPESTATA1	AA2	P1 (1.8 V)	BS-nppdPUk DO DO	1-8 (1) (1) (1)	Configurable I/O Bit 4 drive to the pad matrix EBM pipe A status bit 1
GPIO_8 UART1_RFR_N	F11	P5 (1.8 V)	BS-nppdPUk DO	1-8 (1) (1)	Configurable I/O UART1 ready for receive signal

Table 10-1 General purpose input/output (GPIO) pins (continued)

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
GPIO_7 UART1_CTS_N	F10	P5 (1.8 V)	BS-npPDpuk DI	1-8 (1) –	Configurable I/O UART1 clear to send signal
GPIO_6 SDCC1_CLK	AB4	P7 (2.6 V)	BS-npPDpuk DO	2-16 (8) (8)	Configurable I/O Output clock for SDIO1 device
GPIO_5 SDCC1_DATA3	AA6	P7 (2.6 V)	BS-npPDpuk (HV) DI, DO	2-16 (2) (2)	Configurable I/O SDIO1 data bit 3
GPIO_4 SDCC1_DATA2	AB5	P7 (2.6 V)	BS-npPDpuk (HV) DI, DO	2-16 (2) (2)	Configurable I/O SDIO1 data bit 2
GPIO_3 SDCC1_DATA1	AA5	P7 (2.6 V)	BS-npPDpuk (HV) DI, DO	2-16 (2) (2)	Configurable I/O SDIO1 data bit 1
GPIO_2 SDCC1_DATA0	AA4	P7 (2.6 V)	BS-npPDpuk (HV) DI, DO	2-16 (2) (2)	Configurable I/O SDIO1 data bit 0
GPIO_1 SDCC1_CMD	AB3	P7 (2.6 V)	BS-npPDpuk DI, DO	2-16 (2) (2)	Configurable I/O SDIO1 command bit
GPIO_0 GP_PDM_0 ETM_TRACECLK	C15	P6 (1.8/2.6 V)	BS-npPDpuk DI, DO DO	1-8/2-16 (1/2) (1/2) (1/2)	Configurable I/O “backlight” 12-bit PDM; XO/4 clock ETM trace clock

1 The parameters listed under the Pad type column are defined in the section preceding this table.

2 All GPIOs can be configured as digital outputs with programmable output drive current. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

The rest of this chapter provides the following details about QSC62x0 GPIO pins:

- Pad structure
  - A functional diagram and description of the GPIO pad structure
  - Groupings according to whether the pads are used for ETM mode or not
  - The state of each GPIO pad at powerup and cold resets; these states are maintained until AMSS software programs them differently through control registers
  - Programmable pad configurations, including input, output, and pull options
- Top-level mode multiplexer (TLMM) – A convenient way of programming functional groups of GPIO pads

10.1 Pad structure

Figure 10-1 provides a conceptual illustration of the GPIO pad structure. The figure includes three segments:

- The top segment shows the output driver circuits — See Section 10.1.3.1 for details.
- The middle segment shows the pull circuits — See Section 10.1.3.3 for details.
- The bottom segment shows the input circuits — See Section 10.1.3.2 for details.

All registers controlling pad configuration and control are asynchronously reset to assure immediate pad control on powerup without a clock dependency.

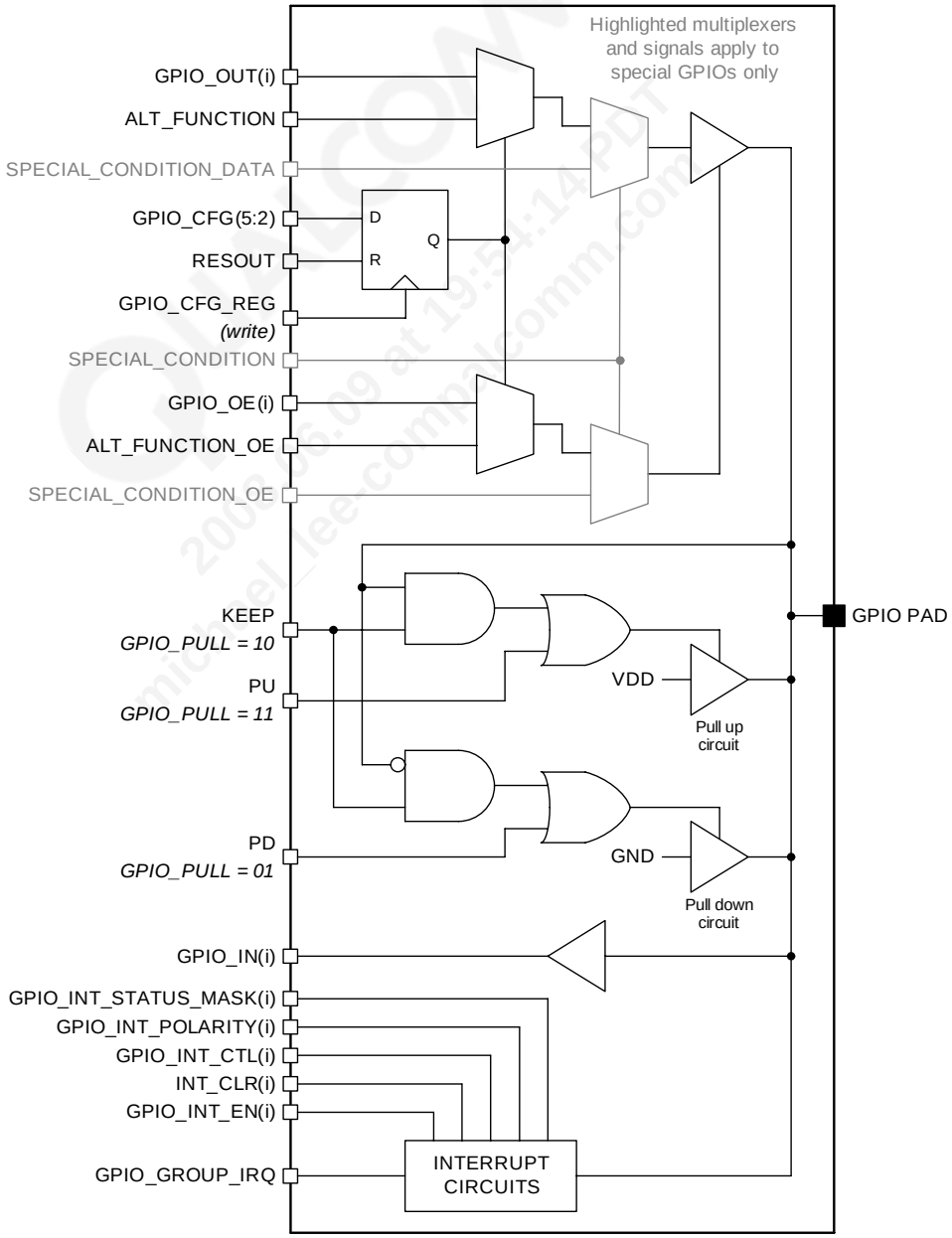


Figure 10-1 Conceptual GPIO pad structure

10.1.1 Pad groupings by ETM usage

Two controllers are used to support the enabling, masking, and status handling of GPIO interrupts for two separate groups of GPIOs:

- Controller 2 supports GPIOs used for ETM mode; this is GPIO group 2
- Controller 1 supports GPIOs not used for ETM mode; this is GPIO group 1

When in the ETM mode, GPIO group #2 functions are replicated off-chip by FPGAs. These GPIO functions are emulated using the EBI2 memory-mapped interface. Since this emulation occurs without software knowledge, the ETM-displaced GPIOs are grouped and distinguished from the other GPIOs.

10.1.2 Power-up states

Figure 10-2 provides a timeline for initializing the GPIO pads. A specific supply sequence is required when powering up the QSC62x0 device. GPIO powerup states require that the VDD_CORE supply turns on before any of the pad supplies (P1, P2, P3, P4, P5, P6, and/or P7) — ideally reaching 90% of its programmed value before any of the pad supplies are enabled. If this sequence and timing relationship is not achieved, the GPIO pads might come up in undefined states. The QSC device ensures the proper supply sequence and timing via its power management functions.

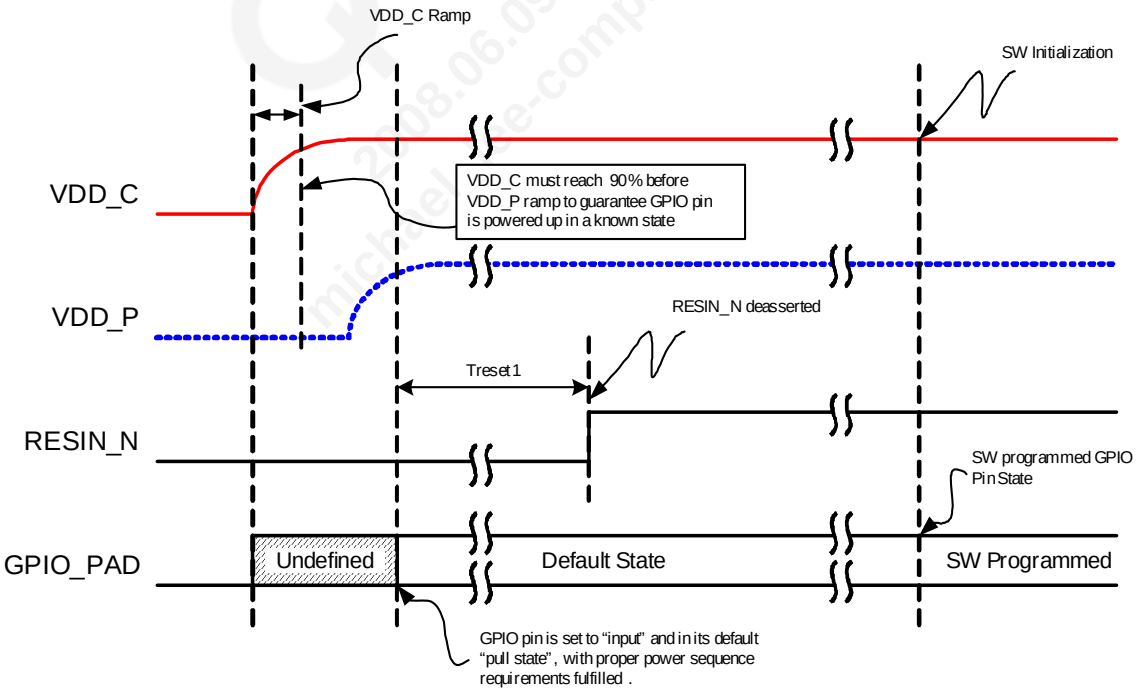


Figure 10-2 GPIO pad powerup state

As the supplies turn on, the GPIO pads go through an interval of undefined states. All GPIO pads are configured as inputs with their default pull states (default pull states were included within Table 10-1) when the RESIN_N signal is driven high (deasserted) by power management circuits, allowing the baseband circuits to initialize. After further delay, software initialization of the programmed GPIO values occurs. The GPIO pads are now properly configured as defined by AMSS software.

10.1.3 Programmable pad configurations

As discussed earlier, every GPIO pad is software configurable. These configuration options are summarized in Table 10-2 and further defined in the following subsections.

Table 10-2 Programmable GPIO configurations

Config type	Configuration description
Input	1. No pull 2. Pull-up 3. Pull-down 4. Keeper
Output	1. No pull 2. Pull-up 3. Pull-down 4. Keeper 5. Programmable drive current Value and range depend upon pad supply voltage 6. Tri-state
Bidirectional	All combinations of the inputs and outputs listed above

10.1.3.1 Output configurations

Three types of output signals are supported (as seen in the top segment of Figure 10-1):

- A normal GPIO output signal – uses GPIO_OUT(i) and GPIO_OE(i). To drive the output pad as a general-purpose output signal, configure the GPIO for nonalternate function, write the GPIO_OUT_X register with the desired value, then set the corresponding bit in the GPIO_OE_X register to enable the output path.
- An alternate GPIO output signal – uses ALT_FUNCTION and ALT_FUNCTION_OE. A procedure similar to the one just described is used, but the alternate functions are exercised rather than the normal functions.
- Special GPIO output signals – these correspond to the GPIO2 group (ETM pins). SPECIAL_CONDITION_DATA, SPECIAL_CONDITION, and SPECIAL_CONDITION_OE are the applicable signals used. These special signals and their associated multiplexers are highlighted in the figure. The special conditions are determined via the external mode control pins.

In addition to these types, each GPIO's output drive strength is programmable (Table 10-3). Pad drive strength is controlled by the appropriate bit fields in the GPIO_PAD_HDRIVE_MSEL_n registers.

Table 10-3 Programmable GPIO drive strength register settings example

GPIO_PAD_HDRIVE_MSELF_0 bits	Variable name	Current setting [1]
8:6	HDRIVE_GPIO_4	000: 2 mA 001: 4 mA 010: 6 mA 011: 8 mA 100: 10 mA 101: 12 mA 110: 14 mA 111: 16 mA

1 The stated 2 mA to 16 mA drive strength applies when the associated pad supply voltage is 2.6 V. GPIOs that are powered off different supply voltages will have somewhat lower drive strength (if lower voltage) or slightly higher drive strength (if higher voltage) than the values shown in Table 10-3. The impact of pad voltage to drive strength is apparent in the Drive current column of Table 10-1.

10.1.3.2 Input configurations

Two types of input configurations are supported (as seen in the bottom segment of Figure 10-1):

- Buffer – A standard CMOS input buffer; its output is GPIO_IN(i).
- Interrupt – An interrupt circuit allows the input signal's level or edge, with selectable polarity, to generate an interrupt. A more detailed diagram (Figure 10-3) and description of the interrupt circuit follows.

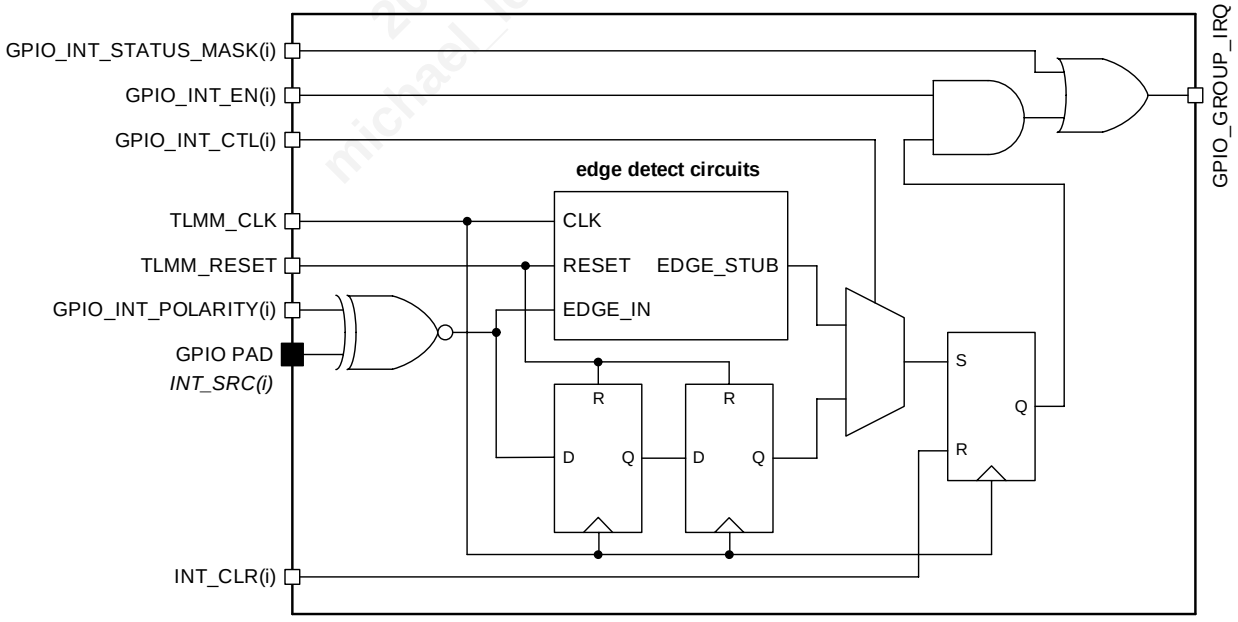


Figure 10-3 GPIO interrupt circuit

The input signal (interrupt source) to the interrupt circuit comes from the GPIO pad. A control signal (GPIO_INT_POLARITY) selects the desired polarity; then additional circuits allow the input signal to trigger an interrupt based on its edge or its level (as selected by the GPIO_INT_CTL signal). The level-based interrupt is delayed two clock cycles. The usual interrupt functions are also provided — clear (INT_CLR), mask (GPIO_INT_STATUS_MASK), and enable (GPIO_INT_EN).

The interrupt circuits depicted in Figure 10-3 are categorized into one of two groups as described in Section 10.1.1. So there are two GPIO interrupt outputs: GPIO_GROUP_IRQ (as shown in the figure) or GPIO_GROUP2_IRQ (not shown).

See the *QUALCOMM Single-Chip QSC240/QSC6270 Software Interface* (80-VF846-2) for further interrupt details.

10.1.3.3 Pull configurations

Three types of pulls are supported (as seen in the middle segment of Figure 10-1):

- Pull-up: a pull-up to the corresponding pad voltage (P1, P2, P3, P4, P5, P6, or P7) as defined within Table 10-1.
- Pull-down: a pull-down to ground.
- Keeper: an internal weak keeper device (keepers cannot drive external buses). This function maintains the pad's last valid logic level regardless of whether it was an input or an output signal.

The internal pulls are implemented using JFETs; their strengths will vary from device to device, but are not expected to be weaker than approximately 100 k.

10.1.3.4 GPIO multiplexing

A paging scheme is used to address individual GPIOs and to specify configuration of the alternate functions, pulls, and drive strengths. Before a particular GPIO pin and pad can be configured through the GPIOx_CFG register, its index must be written to the GPIOx_PAGE register. Non-selected interfaces will be gated in the GPIOx_CFG register, and thus have their inputs held low and quiet.

There are four types of registers in the GPIO blocks:

- Output registers: GPIO_OUT_0, GPIO_OUT_1, GPIO2_OUT_0, and GPIO2_OUT_1. These are read/write registers, where each bit of these registers corresponds to a specific GPIO.
- Output enable registers: GPIO_OE_0, GPIO_OE_1, GPIO2_OE_0, and GPIO2_OE_1. These are write registers, each bit of these registers corresponds to a specific GPIO.
- Input registers: GPIO_IN_0, GPIO_IN_1, GPIO2_IN_0, and GPIO2_IN_1. These are read registers, each bit of these registers corresponds to a specific GPIO's input value.

- Selection registers: GPIOx_PAGE and GPIOx_CFG, as follows:
  - GPIOx_PAGE is a 6-bit register. For GPIO1_PAGE, only non-ETM GPIOs are valid, and for GPIO2_PAGE, only ETM GPIOs are valid. The content of GPIOx_PAGE determines which GPIO is being programmed.
  - The GPIOx_CFG register contains the pin configuration — drive strength (bits 8:6), function (bits 5:2), and pull (bits 1:0).

Some GPIO pads have an additional layer of MUXing after the GPIO MUXing. These are discussed under Special condition MUXing within the next section and consist of the group 2 GPIOs.

For more information regarding registers discussed in this subsection (or entire chapter), refer to the *QSC6240/QSC6270 Software Interface* (80-VF846-2).

10.2 Top-level mode multiplexer (TLMM)

The top-level mode multiplexer (Figure 10-4) provides a convenient mechanism for sharing multiple internal functions onto the same sets of GPIO pads. The mode assignment for each set of GPIOs is specified using a combination of input pin settings (such as MODE[1:0]) and software-programmable register settings. Using the TLMM method for programming GPIO pads allows higher-level instructions, resulting in faster and easier GPIO assignments. Without the TLMM, each GPIO pad would require individual programming.

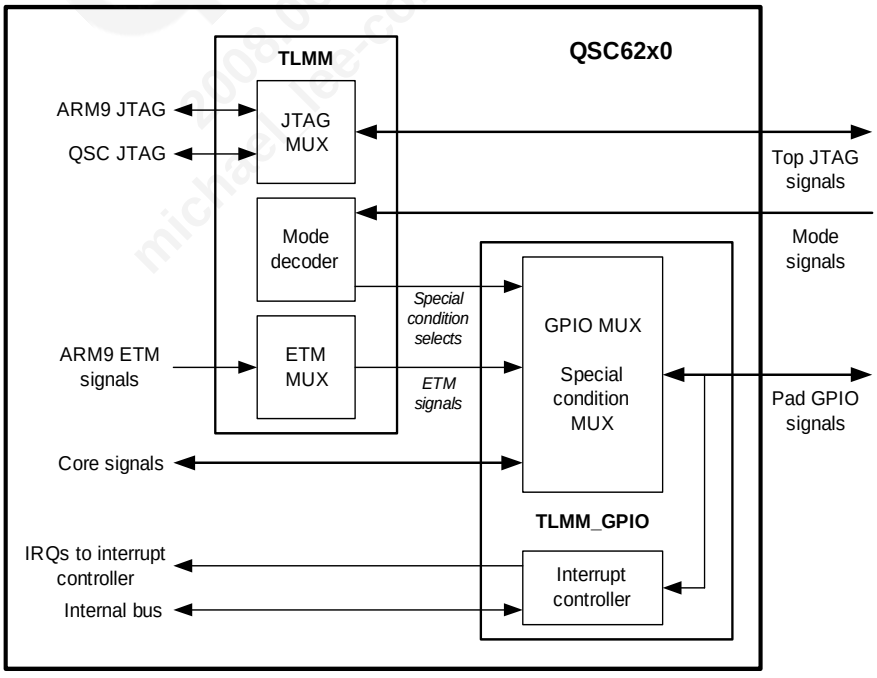


Figure 10-4 TLMM architecture

The TLMM provides software-controlled multiplexing of the QSC62x0 device and provides three multiplexing (and demultiplexing) modes:

- Standard: Most GPIOs fall into this category. They are configured as inputs on powerup, and then are set by software to their desired functionality. Some example uses include GPIOs supporting different feature sets, such as a phone manufacturer choosing to use the UIM interface in lieu of UART2.
- EBIs: These are the GPIO pads used for EBI1 and EBI2 functions. They assume their default EBI1 and/or EBI2 functions on powerup.
- Special condition: These are primarily GPIOs used in ETM modes. The GPIOs used in ETM modes are group 2 GPIOs; an external emulation FPGA is required to emulate GPIO2 functionality (if needed). Refer to Section 11.6 for more ETM details.

The TLMM module receives mode-select control from the mode pins and from software writable registers (used to control the GPIO configuration – drive strength, pull direction, and keeper). The GPIO pin values are readable directly as a register-mapped read.

All registers controlling pad configuration and control are asynchronously reset to ensure immediate pad control on powerup without clock dependency.

11 Internal Baseband Functions

Several baseband circuits within the QSC62x0 device provide functions that are necessary only to make the device operate properly — these functions are not generally used directly by other handset circuits and functions. These internal functions are discussed in this chapter:

- PLLs and clock generation
- Modes and resets
- Security
- Qfuse
- JTAG/ETM

11.1 PLLs and clock generation

The QSC62x0 clock block (Figure 11-1) is a single-phase clock generator that provides all the clocks required by internal cores and external interfaces.

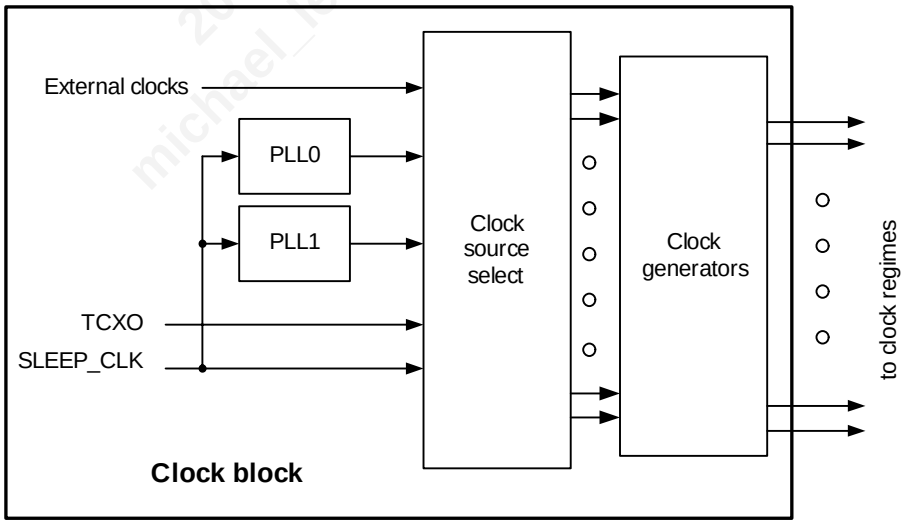


Figure 11-1 Clock block basic architecture

The clock block includes two PLLs, all phase-locked to the TCXO signal. These PLLs generate several different stable, low-jitter clock signals that are distributed throughout the QSC device and to external components as needed.

All the required WCDMA, GSM, GPS, ARM, QDSP, and most peripheral clocks are derived in some way from the TCXO (or XO) source for their operating modes, plus the 32.768 kHz oscillator for their sleep modes.

11.1.1 Clock block features

The clock block features (or its major responsibilities) include:

- Generate and/or distribute the necessary clocks to all cores.
- Support local clock gating to reduce power consumption of any inactive and unused portions of the hardware.
- Generate all the required frequencies for modem operation.
- Generate the required internal wideband codec frequencies.
- Support the external SDAC requirements.
- Support the externally generated PCM clock and sync.
- Provide low frequency, XO/4, and XO/2 clock options for peripheral devices such as the UART and/or UIM.
- Provide the target 184 MHz clock to the ARM.
- Provide the target 92 MHz HCLK for the system bus.
- Provide the target 115 MHz clock to the both aDSP and mDSP.

11.1.2 Clock regimes

The QSC62x0 device generates all clocks required for system performance (Table 11-1).

Table 11-1 Clock regime descriptions [1]

Clock regime	Default source	Source frequency	Output frequency
This information will be added in a future revision of this user guide.			

[1] These clock regimes and capabilities are provided for information and reference only. Handset designers should not deviate from the default system clock settings configured by AMSS — development and testing is optimized with these settings. Handset designers should not make changes to the clocking scheme without first consulting with QUALCOMM.

11.2 Modes and resets

The modes and resets functions of the QSC62x0 device are either self-explanatory or are addressed in detail elsewhere within this user guide. Although these details are not repeated here, pin assignments for all modes and resets are provided in the next subsection for the reader's convenience.

11.2.1 Modes and resets connections

Table 11-2 Clocks, resets, and mode controls

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
RESIN_N	–	P4 (1.8 V)	DI	–	Connected internally to PON_RESET_N
RESOUT_N	F9	P2 (1.8/2.6 V)	DO	(2/4)	Reset output generated by synchronized RESIN_N and WDOG_RESET
MODE_3	T8	P7 (2.6 V)	DI-PD	–	MODE[3:0] definitions: 0000 = Native with JTAG QSC 0001 = Native with JTAG ARM9 Others = Reserved
MODE_2	T9	P7 (2.6 V)	DI-PD	–	
MODE_1	T10	P7 (2.6 V)	DI-PD	–	
MODE_0	AC2	P7 (2.6 V)	DI-PD	–	
WDOG_DISABLE (GPIO_61)	B16	P6 (1.8/2.6 V)	DI	–	Disables watchdog timer when high
BOOT_FROM_ROM (GPIO_62)	A16	P6 (1.8/2.6 V)	DI	–	Determines boot mode if the security enable fuse is not blown: 0 = Invalid - only secure boot is supported 1 = Internal ROM boot (secure) If the security enable fuse is blown, internal ROM boot mode is enabled and this pin can be used as a GPIO.

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

11.3 Security and boot-up

Various security features are supported by QSC62x0 devices. While there are several software security features that may vary with the application, this section focuses on the dedicated hardware (HW) security features of the QSC62x0 device. For more information on the relevant software features, refer to the appropriate software documentation and release notes.

QSC62x0 device hardware security features are mostly controlled by the Qfuse block (see Section 11.4 for details). The main security features controlled by the Qfuse configuration are trusted (secure) boot, hardware key, OEM HW ID, and limited JTAG access under certain conditions.

11.3.1 Boot methodology

AMSS software and firmware images are stored in NAND flash memory. Both 8-bit and 16-bit NAND flash are supported. There are two external memory interfaces on-chip, EBI1 and EBI2. EBI1 supports SDRAM, while EBI2 supports NAND flash.

If the secure-boot eFuse is blown (as required for device operation), the boot code copies the AMSS image from the flash, hashes the image, and authenticates the image. Because the image has to be authenticated before use, the code is considered trusted, free of viruses and hacker code. Booting directly from flash compromises security because flash is subject to tampering.

If the secure-boot eFuse is not blown, the authentication process is skipped by software, and a secure platform cannot be provided, even if booting from on-chip ROM.

Since only trusted boot is available, the QSC62x0 device always boots from internal boot ROM (mapped at 0xFFFF0000 to 0xFFFFFFFF), then the primary boot loader (PBL) in ROM downloads the secondary boot loader (SBL) from NAND flash.

Given the trusted boot requirement, the QSC device must be configured to boot from ROM. This is accomplished one of two ways:

- Blowing the BOOT_FROM_ROM eFuse
- Setting the BOOT_FROM_ROM signal at GPIO_62 to logic high

If the eFuse is blown, any signaling at GPIO_62 is ignored if it is configured as BOOT_FROM_ROM. If the eFuse is blown, GPIO_62 is free to be assigned alternate functionality.

11.3.2 Boot-up procedure

This procedure describes a trusted boot-up using NAND flash. BOOT_ROM is mapped to 0xFFFF0000 and IMEM is mapped to its normal address space, 0x78000000.

1. The boot sequencer takes ARM out of reset and makes ARM start execution from 0xFFFF0000.
2. ARM executes the primary boot loader residing in BOOT ROM, starting at 0xFFFF0000.
3. The primary boot loader instructs the NAND controller to use:
   a. 8-bit NAND mode if BOOT_MODE2 = 0
   b. 16-bit NAND mode if BOOT_MODE2 = 1
4. The primary boot loader performs auto-detection of NAND page size (512 vs. 2k), and NAND type (NAND vs. SuperAND).
5. The primary boot loader copies configuration data from NAND to IMEM.
6. The primary boot loader hashes the secondary boot loader in ARM and copies it to SDRAM.
7. The primary boot loader performs three RSA calculations in ARM.
   a. Performs RSA decrypt on secondary signature and extracts hash.
   b. Compares extracted hash with original.
   c. Hashes attestation certificate and compares against RSA decrypt on the signature.
   d. Hashes intermediate certificate and compares against RSA decrypt on the signature.
8. The primary boot loader transfers control to the secondary boot loader.
9. The secondary boot loader hashes AMSS using SHA-1 accelerator (resides in Crypto Engine) and copies it to SDRAM/SRAM.
10. The secondary boot loader performs three RSA calculations in ARM.
    a. Performs RSA decrypt on secondary signature and extracts hash.
    b. Compares extracted hash with original.
    c. Hashes attestation certificate and compares against RSA decrypt on the signature.
    d. Hashes intermediate certificate and compares against RSA decrypt on the signature.
11. If they do not match, software concludes that the code might be compromised and should not be run. Software then indicates to the hardware that a failure has occurred, and the hardware causes the FAILED_BOOT pin (GPIO_61, pin B16) to toggle to indicate the detected failure to the external user.
12. If they match, boot-up code hands over control to AMSS.

11.3.3 Other boot-related requirements

Every time the power-on reset is asserted then de-asserted, the whole boot-up process is activated. The power-on reset can be caused by the RESIN_N pin or the watchdog.

- TCXO (19.2 MHz) is the only clock available to internal QSC circuits during the boot-up process, and it is used as the source of HCLK.
- Only NAND flash devices with a page size of 512 bytes and 2048 bytes are supported. NAND flash devices with a page size of 256 bytes are not supported.
- Any defects in the first 16 pages of data in NAND flash must be within the ECC's correction capability (1 and 4 bits per ECC codeword). In current NAND flash controller designs, each ECC codeword is 128 bytes long, so each page contains four codewords. The ECC correction capability is therefore 4 bit errors per page evenly distributed into the four codewords.
- NAND devices with I/O width of 8-bits or 16-bits are supported.
- The MMU is not turned on until the entire application software has been transferred to SDRAM.
- Two chip-selects are supported for NAND devices.

11.3.4 Related fuses and external pins

In order for the primary boot loader to authenticate the secondary boot loader, the SECURE_BOOT eFuse must be blown. Table 11-3 describes the secure boot and boot from ROM pins and eFuses. For debugging purposes, if the SECURE_BOOT eFuse is not blown, secondary boot loader authentication is bypassed.

Table 11-3 On-chip eFuses and external mode pins for QSC62x0 configuration

Name	Type	Description
BOOT_FROM_ROM	eFuse	If this fuse is blown, QSC62x0 continues to boot from internal ROM as desired.
BOOT_FROM_ROM	Input pin	If this pin is high, QSC62x0 continues to boot from internal ROM as desired.
SECURE_BOOT	eFuse	If this fuse is blown, the PBL in ROM will authenticate the SBL, and the SBL will authenticate the AMSS image.
HW_REVISION_NUMBER	eFuse	This holds the 4-bit revision number for QSC62x0.

Since the value of the BOOT_FROM_ROM pin is used to protect memory and to hide the 128-bit hardware key (see Section 11.3.8), its value should not be changed. To prevent boot from being performed in non-trusted mode, and then having the BOOT_FROM_ROM pin toggle in an attempt to allow JTAG access and reveal the hardware key and unprotected memory, an internal version of trusted boot is used. This internal signal gets its value from the pin (or samples the pin) at only two points in time — on the release of PON_RESET_N, and on the release of a reset due to a watchdog timeout. These two cases also cause the QSC device to re-authenticate the code stored in flash.

The samples of the BOOT_FROM_ROM pin are captured in registers that cannot be programmed or accessed. Therefore, the boot mode cannot be altered once the ARM starts the boot-up process.

Normally, the sampled version of BOOT_FROM_ROM is used as the trusted boot mode control. To further enhance security, an eFuse is used to override the BOOT_FROM_ROM pin's value. Once the fuse is blown, the QSC device is always forced to boot in trusted boot mode. Operating in the trusted boot mode affects other security features such as hardware key protection, memory protection, secure debugging. These features are discussed in later subsections.

11.3.5 Trusted boot connections

Three QSC62x0 pins (Table 11-4) are associated with the trusted boot function.

Table 11-4 Trusted boot connections

Pin #	Pin name	Pin type [1]	Functional description
A16	GPIO_62 BOOT_FROM_ROM	DI	Determines boot mode (unless overridden by the Qfuse bit): 0 = Nonsecure mode; **this option is not supported**. 1 = Secure mode (authenticated); forces the ARM9 processor to start executing code from location 0xFFFF 0000 (mapped to the on-chip boot ROM). This pin is only meaningful if the security enable fuse is not blown. Once this fuse is blown, pin A16 can be used as a GPIO.
V10	(RESIN_N) PON_RESET_N	DI (DO)	System reset input; externally driven by the PON_RESET_N output or by JTAG. When asserted (0) the baseband circuits are placed in their power-on reset (boot) mode. De-assertion (1) initiates a boot-up.

[1] Pin type is DI = digital input or DO = digital output.

11.3.6 Crypto engine, data mover, and reduced trusted boot time

The boot time for secure boot depends upon the type of flash (whether hashing is done in hardware or software) and the data transfer method (whether directly from flash to the hardware hashing engine or to intermediate memory). The time required to access a page of memory is often longer than the time to hash that page. Therefore, the total time is dominated by access; speeding up the hash will not speed up the boot time. For all supported types of flash, the total time is reduced when a page is read and immediately sent to the crypto engine for hashing while the next page is read.

The crypto engine is a co-processor that relies on the external bus master for its data, context, and configuration/control information for each operation. It is a slave on the SYS_AHB with a CRCI interface to the modem data mover (MDM), but it also provides maskable interrupts that enable its use without the MDM. AES128, DES/3DES, and C2 encryption and SHA1/SHA2 hashing algorithms are supported. The crypto engine supports CBC and EBC operations with encrypt only, hash only, encrypt-then-hash, and hash-then-encrypt modes.

The modem data mover is a configurable multi-channel DMA controller with two channels dedicated for inbound/outbound crypto engine data. The MDM has client (master) interfaces to both SYS_AHB and MDM_AHB buses, and to the mDSP DMA interface. The MDM can be programmed by both the ARM9 and the mDSP through their respective host interfaces.

Figure 11-2 shows the crypto engine and modem data mover connections.

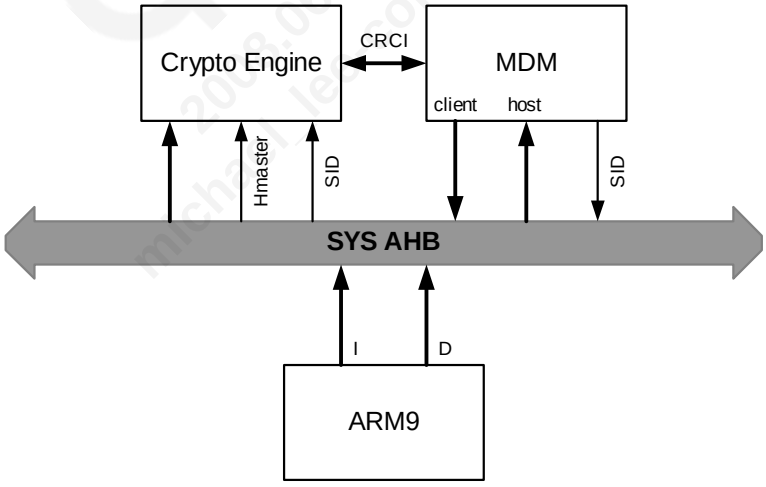


Figure 11-2 Crypto engine and modem data mover connections

11.3.7 aDSP memory protection

Since the aDSP handles sensitive data, its memory space from the ARM9 processor is protected. A small amount of on-chip RAM is attached to the aDSP as required to support security-related processes - keys, client side certificates, identity numbers, C2 cipher, etc. The protected memory space prevents exposure of secure data onto external pins and prevents the ARM from accessing secure information.

A total of 512 bytes of secure RAM is provided. It is protected by blocking the chip-select signal to the externally-addressable port on the aDSP DMA controller (ARM9, video, ADM). When not blocked, reads and writes to the memory space behave normally; when blocked, the data on a write is never written into the RAM and the data on read is erroneous.

An aDSP XMEMC register bit (BLOCK_MEM_SECURE) controls memory blocking. The XMEMC memory space is not accessible through the aDSP DMA interface, so the register is not directly accessible by the ARM9. The memory accesses to the secure RAM are blocked when in trusted boot with the register bit set, and the register bit resets to block memory access. The ADSP memory protection is always on when the SECURE_BOOT eFuse is blown. If secure boot is not used, the memory protection is controlled by the XMEMC aDSP register.

11.3.8 Hardware key

QSC62x0 devices support the use of a unique, random, 128-bit hardware key that is used for a secure file system. This key is programmed by a 128-bit random number, and it is programmed in a dedicated Qfuse chain by blowing the appropriate bits in the chain (see Section 11.4 for more information).

To prevent compromising the hardware key, the following measures are supported:

- The ARM9 processor is connected to aDSP only through the QDSP DMA interface that does not provide access to XMEMC space, so the ARM9 does not have access to the hardware key.
- A Qfuse bit is used to disable programming of the Qfuse in the hardware key Qfuse chain, making shifting of data in and out of the Qfuse chain impossible. After the desired hardware key is programmed, this Qfuse bit is blown and thus an intruder will no longer be able to alter the key or scan the key value out on the JTAG interface. During power-up, the reset circuit will hold the rest of the chip in reset until all fuses can be properly sensed, so there is no way to bypass this Qfuse bit.
- The hardware key is not part of the ATPG scan chain so the key's value cannot be scanned out directly during test.
- When JTAG debugging or TIC_MODE is enabled, a known constant replaces the hardware key.
- If the chip does not receive a reset or the chip boots in non-trusted mode, the hardware key is again replaced by a constant. In this case, the hardware key still cannot be scanned out since the programming/shifting of the Qfuse chains is disabled until reset is properly applied and removed.

11.3.9 OEM HW ID

In addition to the aforementioned features, there are 32 Qfuse bits that can be programmed by handset designers to their own provisioned security values to augment the hardware key. For example, the upper 16-bits of the OEM_HW_ID field can be used as the OEM identifier, while the lower 16-bits can be designated by the OEMs to further customize the QSC62x0 device.

11.3.10 Secure debugging and JTAG access

To ensure security, access to the ARM9 JTAG port is limited when booting in trusted boot mode. Trusted booting (out of the BOOT ROM) normally disables JTAG, but JTAG can be enabled or permanently disabled using a one-time writable register. This register is reset upon power-up and is powered by the Always On™ power domain, so it does not lose its state during a power collapse. Since the register is one-time writable, it is impossible to leave the JTAG state without reapplying hardware reset. Thus, an intruder cannot take over the ARM9 processor and then leave the JTAG state to bypass hardware key protection.

11.4 Qfuse

QSC62x0 devices have an integrated Qfuse block that contains three chains (two 128-bit chains and one 256-bit chain). Two of the chains are reserved for QUALCOMM internal use only. The third chain is EF_CONFIG and is a 128-bit chain; its bit breakdown is summarized in Table 11-5.

Table 11-5 EF_CONFIG_LSB descriptions [1]

Bits	Name	Description
31:4	OEM_HW_ID	OEM_HW_ID[27:0]
3	SW_FUSE_PGM_DSBL	0: enable 1: disable
2	HW_KEY_PGM_DSBL	0: enable 1: disable
1	MEM_REDUN_AND_CHIP_ID_DSBL	0: enable 1: disable
0	CONFIG_PGM_DSBL	0: enable 1: disable

[1] This chain is implemented with a 2:1 redundancy, so every logical bit represented in the chain is actually represented by 2 physical fuse bits. When programming the fuse chain, as will be explained in the next section, it is very important to understand this and to account for this in the programmed chain length and shift values. Thus, for every bit n in an x long chain, the physical fuse bits to be blown are 2n and 2n+1 in a chain that is 2x long.

Table 11-6 EF_CONFIG_MSB descriptions [1]

Bits	Name	Description
31	SPARE_FUSE_BIT	Reserved
30	OEM_HW_ID_AUTH_ENABLE	When blown, the OEM_HW_ID is used in the authentication of the secondary boot loader. This eFuse allows customers to use the OEM_HW_ID for their own purposes or for authentication, depending on what their needs are.
29:28	BOOT_ROM_CFG	Allows boot code to bypass certain risky but higher performance functions in favor of safer alternatives when the fuse is blown.
27:26	RESERVED	
25:22	V_SCALE	Dynamic voltage scaling constant.
21:20	RESERVED	
19	EBI1_NOR_BOOT_EN	Boot ROM code execution is immediately branched to external memory when the BOOT_FROM_ROM eFuse is not blown. The external memory the boot ROM code branches to is determined by the EBI1_NOR_BOOT_EN eFuse. 0: branch to EBI2 CS[0] (address 0x0 after EBI1/EBI2 address swap) 1: branch to EBI1 CS[0] (address 0x0)
18	C2_CIPHER_KEY_DISABLE	Reserved
17:14	ARM9_RAM_ACC_CTRL	Reserved
13:12	SOFTWARE_CRASH_DEBUG	
11	JTAG_EN_N	0: always enable ARM9 JTAG 1: normal functional mode
10	SECURE_BOOT	0: do not force SBL authentication 1: force code authentication
9	BOOT_FROM_ROM	0: boot from external memory 1: use Boot ROM boot loader
8:4	ROOT_KEY_SELECT	Select which certificate to use for authentication
3:0	OEM_HW_ID	OEM_HW_ID[31:28]

[1] This chain is implemented with a 2:1 redundancy, so every logical bit represented in the chain is actually represented by two physical fuse bits. When programming the fuse chain, as will be explained in the next section, it is very important to understand this and to account for this in the programmed chain length and shift values. Thus, for every bit ‘n’ in an ‘x’ long chain, the physical fuse bits to be blown are 2n and 2n+1 in a chain that is 2x long.

A flowchart for the Qfuse blowing procedure using JTAG is shown in Figure 11-3.

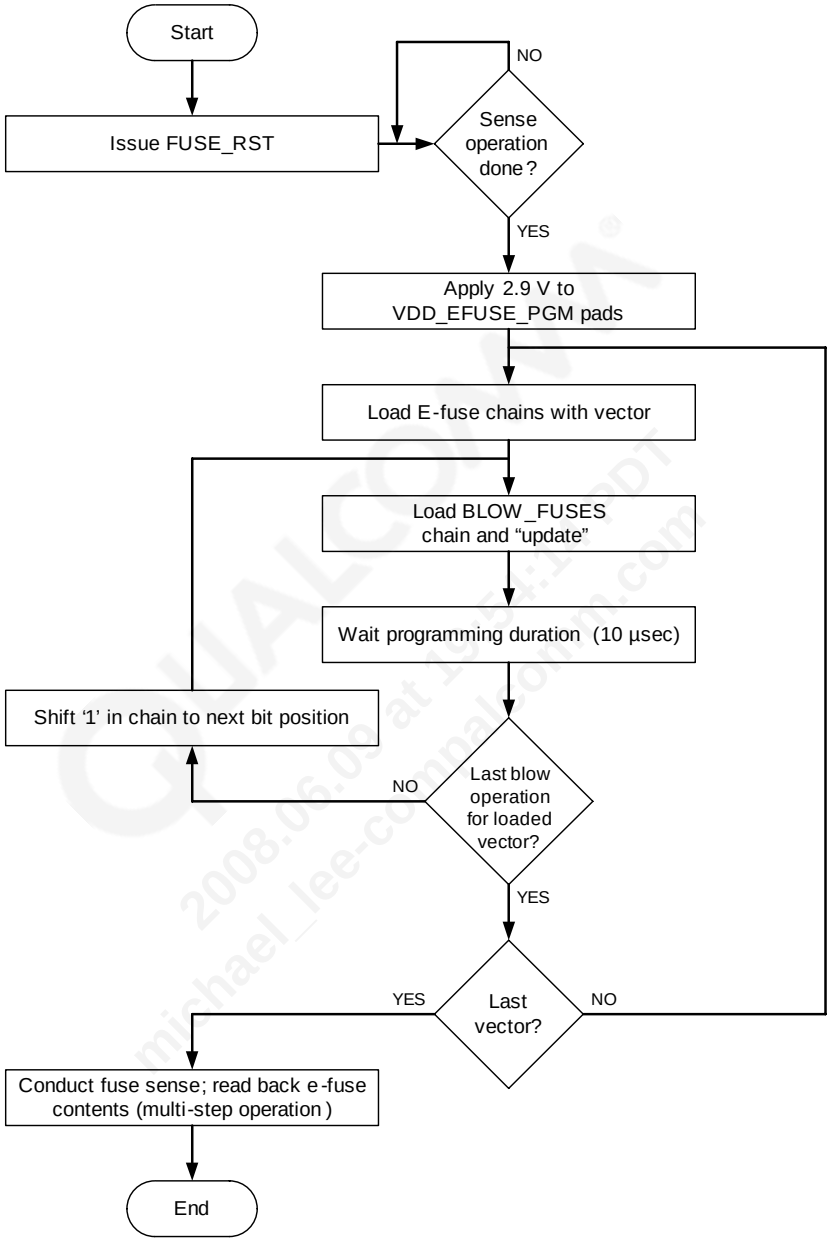


Figure 11-3 Qfuse blowing procedure

NOTE Since it is imperative that only one physical fuse bit should be blown at a time, whenever a fuse-blowing process starts, handset designers should set the EF_BLOW_VALUE to 0x0 with EF_SHIFT_VALUE set to the chain length. This step ensures that any 1 in the chain will be cleared out.

The following instruction sequence explains the Qfuse programming procedure:

1. Set mode pins to JTAG program eFuse mode.
2. Wait for eFuse sense operation to finish.
3. Load the program value via JTAG to the intended serial chain. Since each fuse being blown requires a current between 10 mA and about 17 mA, just one eFuse can be programmed at a time - the loaded pattern must not exceed one bit set per QFUSE_N cell. Step 5 describes how the BLOW_FUSES chain is used to select a single QFUSE_N cell.
4. Apply 2.9 V (nominal) to pin V13 (VDD_EFUSE).
5. Load the appropriate values to program the EF_CHAIN_SEL chain. Load the EF_BLOW_TIMER register with the number of cycles needed to blow the fuses (10 microseconds). Writing to the STATUS_START bit in the EF_STATUS register starts a state machine that actually blows the fuse.
6. Once each QFUSE_N cell has been programmed for a loaded vector, load the chain with all zeros (using EF_BLOW_VALUE and EF_SHIFT_VALUE) to disable input ports for all QFUSE_N cells.
7. Go to step 2 to load the next vector. Repeat until all eFuse bits are programmed.

Finally, after all the desired fuse programming is complete, the disable bit (CONFIG_PGM_DSBL bit of EF_CONFIG_LSB) can be blown to eliminate access and assure no more tampering to those Qfuse chains.

11.4.1 Important fuse considerations

In order to assure proper fuse blowing, customers should pay extra attention to the following considerations:

- The maximum clock frequency is 20 MHz, running off TCK when JTAG is used or TCXO when the software interface is used.
- A 2.9 ± 0.1 V fuse blowing voltage is needed. This needs to be switched between 2.9 V (when programming fuses) and ground (during sensing and normal operation).
- Only one fuse can be blown at a time. This restriction is due to the high-current requirements of fuse blowing. Therefore, customers should pay extra attention to the above instructions and the clearing of the chain(s) when fuse-blowing begins.
- The CONFIG chain is implemented with 2:1 redundancy for reliability reasons.

Therefore, for every logical bit (described in Table 11-5), there are actually two fuse bits that should be blown. These two fuse bits are effectively logical OR'd to give the value of the logic bit. This means that the chain length of CONFIG is actually 128.

In order to take advantage of the redundancy and increase reliability, the two physical fuse bits need to be blown for every logical bit. Thus, to blow logical bit i, physical bits 2i and 2i+1 need to be blown. For example, if we wanted to blow the SECURE_BOOT fuse bit (logical bit 10+32 = 42 in the CONFIG chain – see Table 11-5), we would have to program the chain length to 128, and shift the 1 to positions 84 and 85 in the chain.

11.5 Joint Test Action Group

This section describes and explains the Joint Test Action Group (JTAG) interface on QSC62x0 devices: supported features, its adherence to the IEEE standard 1149.1-1993, and usage information.

11.5.1 JTAG standard overview

The QSC62x0 JTAG interface conforms to the IEEE 1149.1A-1993 standard specifying components that accept test instruction and data inputs, then provide the respective results as outputs in a serial format. The standard requires a test access port and a boundary-scan architecture in fulfilling the above requirements.

This test circuitry is used for board-level testing, and achieves the following objectives:

- Confirms that each component on the board performs its function correctly
- Confirms that all components are interconnected in the correct manner
- Confirms that the entire design behaves as intended

These confirmations are achieved using a boundary-scan architecture technique that includes a shift register stage (or cell) adjacent to each component pin so that signals at the component's boundaries can be tested, controlled, and observed. The boundary-scan cells will be connected in a serial manner as a long chain and will behave as an overall shift register. More information about the boundary-scan cells is provided in Section 11.5.6.3.

For more detailed JTAG information, refer to the *IEEE Standard 1149.1A-1993* and the *JTAG/ETM Interface for ARM9-based MSM Devices Application Note* (80-V7838-1).

11.5.2 JTAG connections - the test access port (TAP)

QSC62x0 pins supporting the JTAG interface are listed in Table 11-7.

Table 11-7 JTAG connections

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
TRST_N	V6	P1 (1.8 V)	DI-PD	–	JTAG reset
TCK	U6	P1 (1.8 V)	DI-PU	–	JTAG clock input
TMS	W7	P1 (1.8 V)	DI-PU	–	JTAG test mode select
TDI	V5	P1 (1.8 V)	DI-PU	–	JTAG test data input
TDO	V3	P1 (1.8 V)	Z	1-8 (8)	JTAG test data output
RTCK	U5	P1 (1.8 V)	DO	1-8 (8)	JTAG return clock
(RESIN_N) PON_RESET_N	V10	P4 (1.8 V)	DI	–	Reset input, connected to QSC PON_RESET_N output. JTAG drives this pin and overrides the PON_RESET_N.
PS_HOLD	V11	---	DO	---	Connected internally to PM circuitry to initiate/control powerdown and powerup sequences. In JTAG mode, JTAG overrides the internal drive and ensures it is always asserted (1), probably within the adapter.

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

The following sections provide more information regarding the JTAG interface signals, and the basic connections required from the external ARM JTAG device (ICD) to the QSC62x0 device.

11.5.3 JTAG operation

The QSC62x0 JTAG interface aids in mobile station board-level testing and debugging.

Referring to the IEEE 1149.1A-1993 manual: *IEEE Standard Test Access Port and Boundary Scan Architecture*, the compliance clause defines how compliance with this standard is switched on or switched off.

In QSC devices, the test access port (TAP) selection is included in the MODE selection for the device. For normal operation, and to select the ARM JTAG TAP, the MODE[2:0] pins are left unconnected (they are pulled low internally). To disable the watchdog, the WDOG_DISABLE pin must be pulled high externally; the WDOG_DISABLE pin is pulled low internally to enable the watchdog for normal operation. To select the QSC device scan chain TAP, or BSDL, and also for Qfuse blow2 operation, the MODE [2:0] pins must be pulled high externally. Also, for this latter mode (BSDL), WDOG_DISABLE must be asserted.

NOTE This implementation does not conform to 1149.1-90 but does meet the intent of IEEE 1149.1A 93.

A TAP controller that is not selected is held in the TEST_LOGIC_RESET state. Since the JTAG port is shared, the TMS and TCK pins are not seen by both controllers. TRST_N is pulled low internally on the QSC device.

As defined by the standard, the QSC62x0 JTAG interface allows test instructions and data to be shifted into the QSC device, and allows the test results to be read out in a serial format.

The JTAG interface on QSC62x0 devices consists of four main functional elements: a TAP, a TAP controller, an instruction register, and three test data registers. Figure 11-4 shows the QSC62x0 device-level JTAG interface; Figure 11-5 shows the functional diagram for each internal interface block.

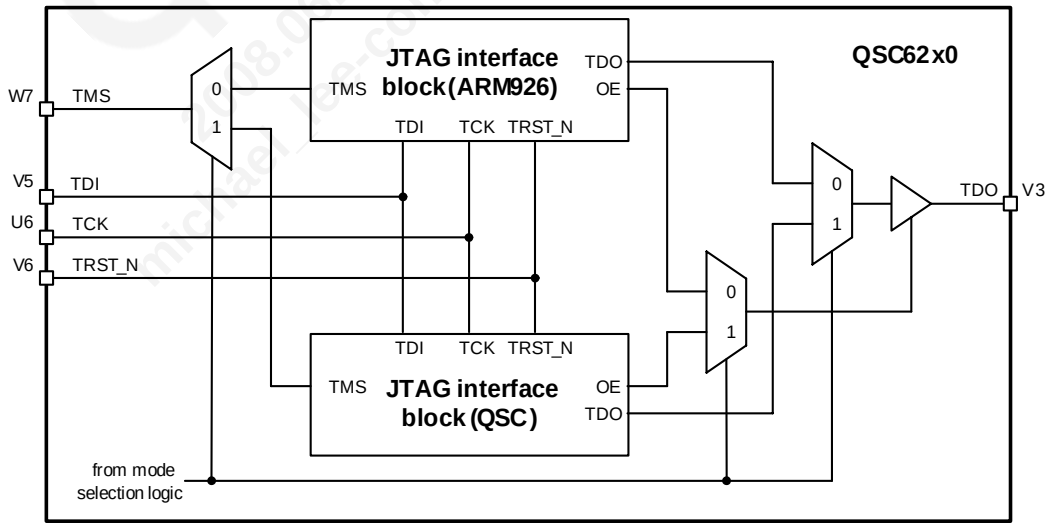


Figure 11-4 QSC device-level JTAG interface

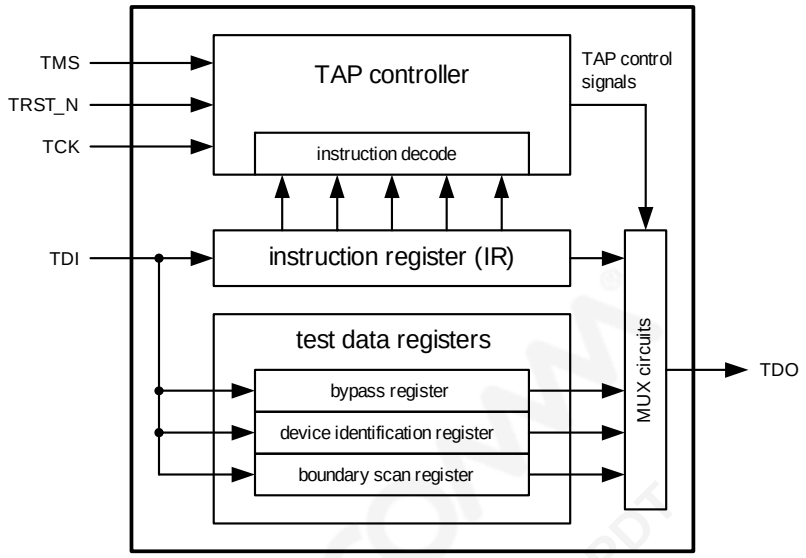


Figure 11-5 Functional diagram for each internal JTAG interface block

11.5.4 Test access port (TAP)

The JTAG interface is accessed through the TAP. The TAP includes the following input/output (I/O) pins:

- TCK – Test clock input
- TMS – Test mode select input
- TDI – Test data input
- TDO – Test data output
- TRST_N – Test reset input

The TAP dedicates its I/O pins to the JTAG interface. The nature of these pins and any internal pulls they may have are described in the following subsections; these pins have no other use on the QSC device.

Test clock (TCK) input

TCK is a user-defined clock input for the JTAG interface. It is independent of the system clock, and it supplies a clock input to the serial test path between TDI and TDO. TCK also permits shifting of test data concurrently with QSC62x0 device operation. The fact that it is independent ensures that the shifting of test data in and out of the TAP does not change the state of the QSC device. If the TCK signal is stopped at zero, stored-state devices contained in the test logic retain their state indefinitely.

Test mode select (TMS) input

The test mode select input is sampled on the rising edge of TCK, and is decoded by the TAP controller to set up all operations. The TMS input is pulled-high internally; if unconnected, it sees a logic high input thus ensuring that the QSC62x0 device continues operating unhindered by the test logic. Therefore, a logic zero must be applied externally to enter test modes.

Test data input (TDI)

TDI is the TAP input for test instruction or test input (serial format). It is sampled on the rising edge of TCK. Data is pushed into the TAP through TDI and propagates to TDO without inversion. Furthermore, it is pulled high internally, again seeing a logic high if left unconnected. This ensures that open-circuit faults at the board-level and in the serial data path force a defined logic state into the TAP.

Test data output (TDO)

TDO outputs the TAP serial data. Changes in the state of the signal driven through TDO occur only on the falling edge of TCK. This ensures race-free operation (since changes on TMS and TDI occur only on the rising edge of TCK). The TDO driver will be inactive at all times (tri-stated) unless data scanning is in progress.

Test reset (TRST_N) input

When this TRST_N signal is driven to a logic low, it asynchronously resets the TAP controller and drives it to the test-logic-reset state, and asynchronously initializes all other test logic (as is required by the test-logic-reset state). This input is pulled down internally to make sure it does not interfere with normal device operation, even if unconnected.

When in JTAG mode, it should be ensured that TRST_N is asserted for at least 5 tck cycles before doing anything.

11.5.5 TAP controller

The TAP controller is a synchronous finite state machine that responds to changes in the TMS and TCK signals by controlling a sequence of operations. This state machine has 16 states, allowing both data and instructions to be shifted. There are states for capturing, shifting, and updating data and instructions; a state for running tests; plus a reset state. For a detailed description of the different states and a state diagram, refer to *IEEE Standard 1149.1a-1993*.

The actions of test logic (both instruction and data registers) will occur on either the rising or falling edge of TCK in each respective controller state. More information on this is provided in *IEEE Standard 1149.1a-1993*.

The TAP controller initializes when it is in the test-logic-reset state. The two ways to achieve this state are:

- Applying a logic low to TRST_N brings the TAP controller asynchronously into the test-logic-reset state.
- Holding the TMS high while allowing at least five rising edges of TCK to occur brings the TAP controller synchronously into the test-logic-reset state (worst-case time to reach it from any other state).

Once the test-logic-reset state is reached, all test logic disables and normal QSC device operations resume.

For board-level designs where JTAG testing is not used, TRST_N must be a logic low. Before using a QSC62x0 device in a mobile station application, TRST_N must be pulled low at power-up, or the TAP controller must be in the test-logic-reset state.

11.5.6 Data registers

Three registers within each QSC62x0 device JTAG interface block are data registers: device identification register, bypass register, and boundary-scan register.

11.5.6.1 Device identification register

The device identification register provides a way for the manufacturer, part number, and version of a component to be determined through the TAP. One likely use: distinguishing the manufacturers of components on an assembled board when more than one source is used.

The device identification register used on the JTAG interface of the QSC device is a 32-bit register holding three pieces of information:

1. The device's manufacturer identity code, bits[11:1]: This code is administered by JEDEC; QUALCOMM's code is 0x070.
2. Part number, bits[27:12]: This is a QUALCOMM-generated number.
3. Version data, bits[31:28]: This is also a QUALCOMM-generated number.

The least significant bit (bit 0) is the device identification register's start bit, and is set to a logic 1. The device identification register contents for both QSC62x0 device types are listed in Table 11-8.

Table 11-8 QSC62x0 device identification register contents

Device	Sample type	Value
QSC6240	Engineering sample I	TBD
QSC6270	Engineering sample I	TBD

Selecting the device identification register loads its value into the shift register on the rising edge of TCK in the capture-DR state. It has a parallel input, but it does not have a parallel output. Figure 11-6 shows the structure of the device ID register.

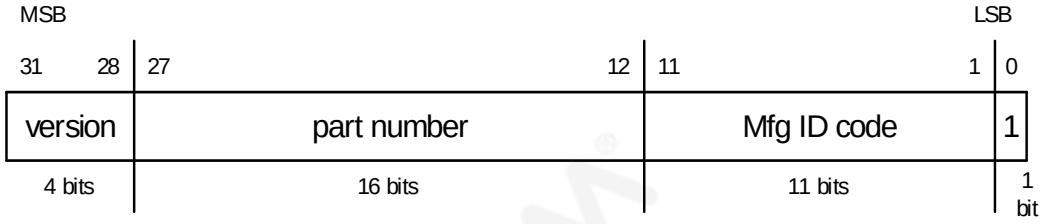


Figure 11-6 Device identification register data structure

11.5.6.2 Bypass register

The bypass register is a single-stage shift register that is located between the TDI and the TDO. It is used to route the scan data directly from TDI to TDO without going through the entire boundary-scan register. Bypassing the boundary-scan register allows quicker movement of data to and from other board components. The operation of the bypass register will have no effect on the operation of the on-chip system logic. It does not have a parallel output.

Selecting the bypass register loads the shift register with a logic low on the rising edge of TCK, following entry into the capture-DR state.

11.5.6.3 Boundary-scan register

The boundary-scan register allows board interconnection testing in order to detect typical defects, like opens and shorts. The boundary-scan register connects, via boundary-scan cells (Figure 11-7), between each digital I/O pin and the QSC-internal baseband circuits. This gives the tester the capability to access and/or sample system I/O signals when testing system logic. The boundary-scan register also handles parallel inputs and outputs.

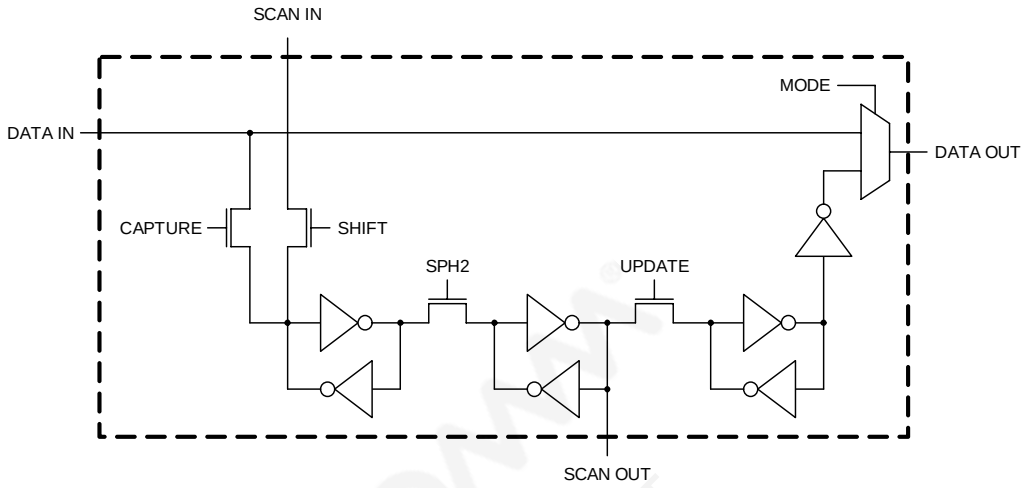


Figure 11-7 Typical structure of a boundary-scan cell

The boundary-scan register is comprised of boundary-scan cells that access digital signals within the QSC62x0 device. Each cell has a single-shift register stage, has single serial input and output terminals, and is connected to the cell before it and the cell after it within the boundary-scan register. In addition, each cell has a parallel input/output terminal through which data is sampled and/or latched (this is how the entire register handles parallel input/output).

11.5.7 Instruction register

The instruction register (IR) is a 7-bit shift register having a serial input and parallel latched output. It holds JTAG instructions that were shifted into the JTAG interface, and allows instructions to be serially loaded into the test logic during an IR scan cycle. A JTAG instruction shifts through TDI (LSB first) until it is in the IR. Once in the IR, the instruction is decoded and the test to be performed and the data register to be used are selected. The parallel output holds the current instruction, and is updated on the falling edge of TCK (when in the update-IR state), or asynchronously on entry into the test-logic-reset state. The basic functions used and supported on the QSC JTAG interface are BYPASS, SAMPLE/PRELOAD, EXTEST, and IDCODE.

11.5.7.1 IDCODE consideration

The following design consideration should be implemented for handling an IDCODE instruction while in BSDL (or QSC62x0 JTAG) mode:

Before issuing the IDCODE instruction, or blowing fuses, customers should be sure to wait at least 45 tck cycles.

NOTE For more general information on JTAG instruction registers, refer to the *JTAG/ETM Interface for ARM9-based MSM Devices Application Note* (80-V7838-1).

11.5.8 JTAG selection

Since both the embedded ARM microprocessor and the QSC62x0 device have JTAG capability, the active JTAG interface is selectable via the MODE[2:0] pins. Restating previous instructions for emphasis:

- To select the ARM JTAG, these MODE pins should be left unconnected (they are pulled-low internally). See Figure 11-8.
- To select QSC JTAG (or BSDL), the MODE_2 pin should be pulled high externally. See Figure 11-9.
- Realizing that the watchdog timer could expire during JTAG operation (either ARM or QSC), it is important to externally pull up its WDOG_DISABLE pin to disable it and allow JTAG operation.

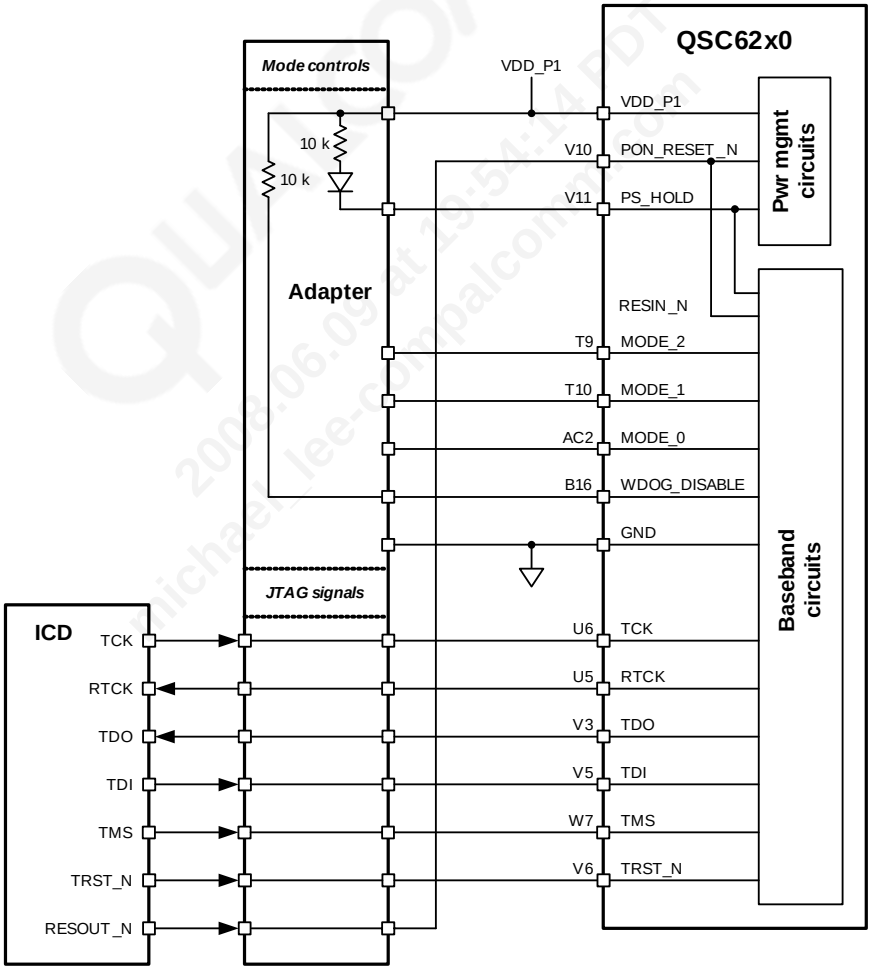


Figure 11-8 Pin connections for selecting ARM JTAG, WDOG disabled (no ETM)

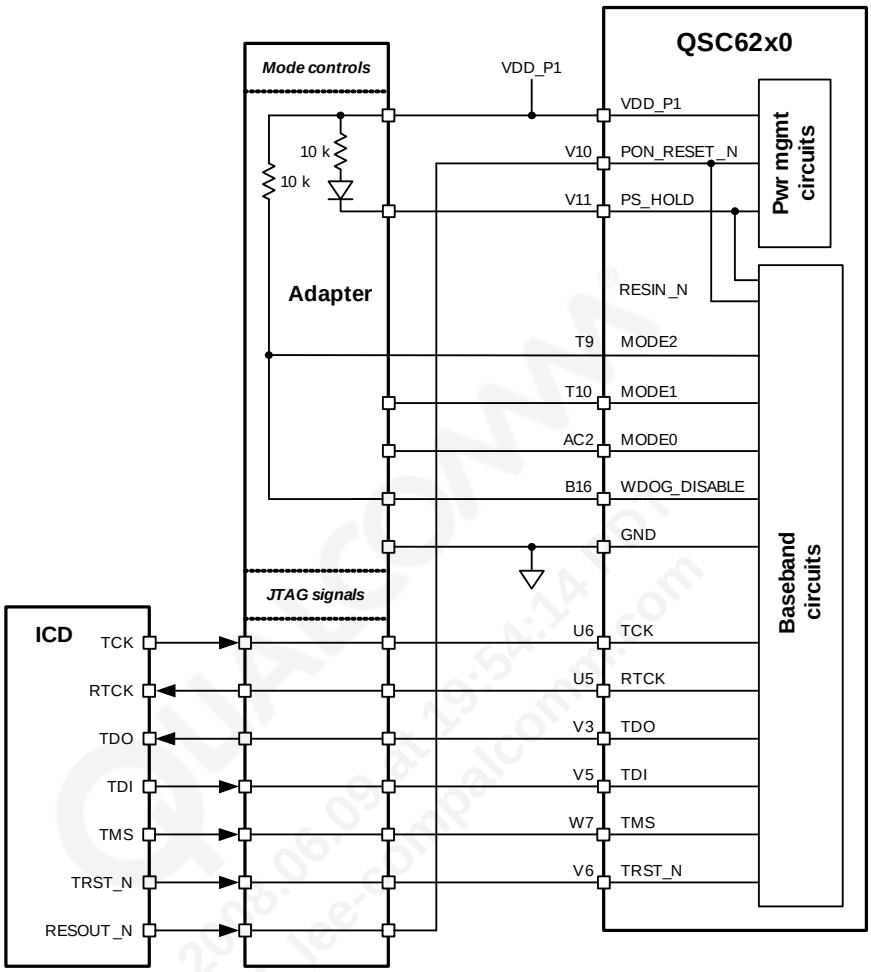


Figure 11-9 Pin connections for selecting QSC JTAG (BSDL), WDOG disabled

11.6 Embedded trace macrocell

This section describes the operation of the embedded trace macrocell (ETM) in the ARM family of processors, and how it is implemented and used within QSC62x0 devices. Since the QSC device has a target ARM speed of 184 MHz, the deMUX ETM mode should be used for target-speed emulation (the clock pin may not be able to toggle that fast in the normal mode).

The ETM is a block that provides instruction and data trace for the ARM family of microprocessors; its use within a system allows real-time debugging. QSC62x0 devices use ETM9 — the block used by ARM9 microprocessors.

The ETM consists of two parts:

- Trace port: broadcasts trace information (either instruction trace or data trace)
- Triggering facilities: controls the ETM to filter and control trace operations

ETM is designed to be connected directly to the ARM core it is tracing. The trace port connects to a software debugger that configures the ETM and processes information provided by the ETM.

NOTE For more information regarding the ETM, refer to ARM ETM9 manuals and documentation at www.arm.com or the *JTAG/ETM Interface for ARM9-based MSM Devices Application Note* (80-V7838-1).

An example debugging environment is shown in Figure 11-10.

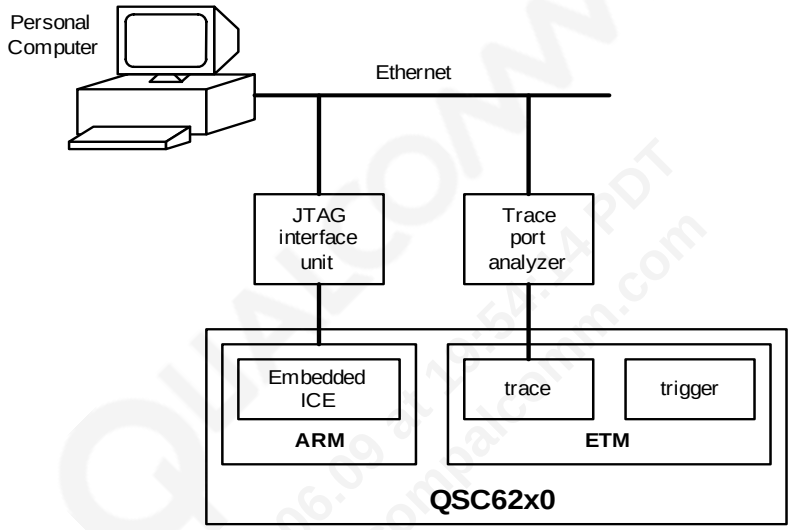


Figure 11-10 Example debugging environment using ETM

More detailed information on the ETM and its modes is provided throughout the rest of this section.

11.6.1 ETM architecture

The basic architecture of the ETM is shown in Figure 11-11.

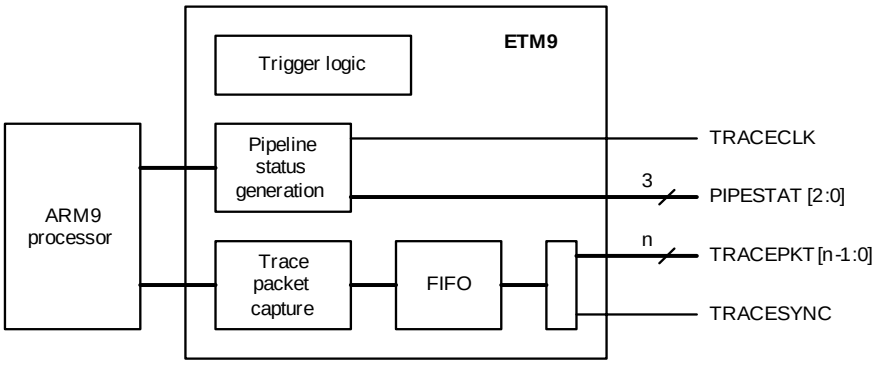


Figure 11-11 Basic ETM architecture

The trace packet can be 4, 8, or 16 bits, with bigger packets allowing higher trace data bandwidth.

The trace port is composed of four signals:

- PIPESTAT (2:0) – Three pipeline status pins
- TRACEPKT – An n-pin trace packet port, where n can be 4, 8, or 16 pins
- TRACESYNC – A trace packet synchronization pin
- TRACECLK – A clock signal with the same frequency as the processor clock

The GPIO pins used for each of the above signals are described in Section 11.6.3.

In addition to the ETM modes supported in previous MSM devices that use ETM7, the QSC62x0 ETM9 implementation includes a wide (de-multiplexed) trace port. This is achieved by clocking the trace port at half the processor operating frequency and routing trace port outputs to pairs of output pins (see Section 11.6.4.2 for more information).

11.6.2 ETM features

The ETM configuration on QSC62x0 devices supports several different options. However, as mentioned earlier, the device's target speed of 184 MHz might make full speed emulation unrealizable due to limitations introduced by the pads' maximum toggling frequency. Therefore, the 8-bit deMUX mode is the best choice for target speed QSC emulation needs.

11.6.3 ETM connections

QSC62x0 devices are capable of implementing the maximum trace port size (16 bits) and the maximum FIFO size.

For ETM debugging, QSC devices are set to a dedicated debug mode called ETM mode. This mode is set through the MODE pins: MODE[0] is pulled high externally, and MODE[1] is left unconnected (pulled-down internally). In ETM mode, the QSC device operates in its native mode, but with selected GPIO pins dedicated to the ETM trace port.

The trace data and clock for various modes are implemented differently. In deMUX mode, the trace data (TSYNC, PSTAT (2:0), TDATA_(15:0)) are deMUXed to the A and B path. A programmable delay cell in the clock path allows delay adjustment. This ensures timing for the trace port analyzer (TPA). In normal mode, the A and B paths are the same.

The ETM interface uses GPIO pins. A total of 28 GPIO pins (including the ETM off-chip FPGA emulation related signals) are needed to support the different ETM modes. The ETM configuration register's PORTMODE (1:0) is used to select normal or de-multiplexed configuration. The PORTSIZE[2:0] bits are used to select between 8- or 16-bit modes. For more information about the configuration of the ETM, refer to the appropriate ARM documentation.

Table 11-9 and Table 11-10 lists and describes the relevant GPIO pins and the different signal assignments for each ETM mode.

Table 11-9 ETM connections

Pin name/function	Pin #	Pad group [1]	Pad type [2]	Drive current [3]	Functional description
ETM - normal 8-bit and demux 8-bit					
ETM_TRACE_PKTA7 (GPIO_18)	W1	P1 (1.8 V)	DO	1-8 (1)	Trace packet A bit 7
ETM_TRACE_PKTA6 (GPIO_17)	V1	P1 (1.8 V)	DO	1-8 (1)	Trace packet A bit 6
ETM_TRACE_PKTA5 (GPIO_16)	V2	P1 (1.8 V)	DO	1-8 (1)	Trace packet A bit 5
ETM_TRACE_PKTA4 (GPIO_15)	W2	P1 (1.8 V)	DO	1-8 (1)	Trace packet A bit 4
ETM_TRACE_PKTA3 (GPIO_14)	W3	P1 (1.8 V)	DO	1-8 (1)	Trace packet A bit 3
ETM_TRACE_PKTA2 (GPIO_13)	Y1	P1 (1.8 V)	DO	1-8 (1)	Trace packet A bit 2
ETM_TRACE_PKTA1 (GPIO_12)	Y2	P1 (1.8 V)	DO	1-8 (1)	Trace packet A bit 1
ETM_TRACE_PKTA0 (GPIO_11)	AA1	P1 (1.8 V)	DO	1-8 (1)	Trace packet A bit 0
ETM_PIPESTATA2 (GPIO_43)	C9	P5 (1.8 V)	DO	1-8 (1)	ETM pipe A status bit 2
ETM_PIPESTATA1 (GPIO_9)	AA2	P1 (1.8 V)	DO	1-8 (1)	ETM pipe A status bit 1
ETM_PIPESTATA0 (GPIO_10)	Y3	P1 (1.8 V)	DO	1-8 (1)	ETM pipe A status bit 0
ETM_TRACESYNC_A (GPIO_41)	H12	P5 (1.8 V)	DO	1-8 (1)	Trace packet A sync
ETM_TRACECLK (GPIO_0)	C15	P6 (1.8/2.6 V)	DO	1-8/2-16 (1/2)	ETM trace clock
ETM_MODE_CS_N (GPIO_44)	H10	P5 (1.8 V)	Z	1-8 (1)	ETM chip-select
ETM_MODE_KYSNS_INT (GPIO_42)	B9	P5 (1.8 V)	DI	–	ETM interrupt - alerts QSC of a key press
ETM_MODE_INT (GPIO_27)	E12	P5 (1.8 V)	DI	–	Interrupts ETM mode

Table 11-9 ETM connections (continued)

Pin name/function	Pin #	Pad group [1]	Pad type [2]	Drive current [3]	Functional description
ETM - demux 8-bit only					
ETM_TRACE_PKTB7 (GPIO_37)	C14	P5 (1.8 V)	DO	1-8 (1)	Trace packet B bit 7
ETM_TRACE_PKTB6 (GPIO_36)	B14	P5 (1.8 V)	DO	1-8 (1)	Trace packet B bit 6
ETM_TRACE_PKTB5 (GPIO_35)	F13	P5 (1.8 V)	DO	1-8 (1)	Trace packet B bit 5
ETM_TRACE_PKTB4 (GPIO_34)	B13	P5 (1.8 V)	DO	1-8 (1)	Trace packet B bit 4
ETM_TRACE_PKTB3 (GPIO_33)	C13	P5 (1.8 V)	DO	1-8 (1)	Trace packet B bit 3
ETM_TRACE_PKTB2 (GPIO_32)	F12	P5 (1.8 V)	DO	1-8 (1)	Trace packet B bit 2
ETM_TRACE_PKTB1 (GPIO_31)	B12	P5 (1.8 V)	DO	1-8 (1)	Trace packet B bit 1
ETM_TRACE_PKTB0 (GPIO_30)	C12	P5 (1.8 V)	DO	1-8 (1)	Trace packet B bit 0
ETM_PIPESTATB2 (GPIO_25)	C11	P5 (1.8 V)	DO	1-8 (1)	ETM pipe B status bit 2
ETM_PIPESTATB1 (GPIO_28)	B11	P5 (1.8 V)	DO	1-8 (1)	ETM pipe B status bit 1
ETM_PIPESTATB0 (GPIO_29)	E13	P5 (1.8 V)	DO	1-8 (1)	ETM pipe B status bit 0
ETM_TRACESYNC_B (GPIO_26)	E11	P5 (1.8 V)	DO	1-8 (1)	Trace packet B sync

1 External voltage translations may be required for ETM mode.

2 The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

3 Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

Table 11-10 Pin multiplexing for different ETM modes

Pin description	8-bit deMUX mode pin function	8-bit normal mode pin function
Trace clock	Trace clock	Trace clock
Trace sync	Trace sync #1	Trace sync
Pipe stat [2:0]	Pipe stat [2:0] #1	Pipe stat [2:0]
Trace packet [7:0]	Trace packet [7:0] #1	Trace packet [7:0]
Trace packet [15:8]	Trace packet [7:0] #2	N/A
Trace sync #2	Trace sync #2	N/A
Pipe stat [2:0] #2	Pipe stat [2:0] #2	N/A

11.6.4 ETM modes

The recommended 8-bit deMUX mode is discussed in Section 11.6.4.1.

11.6.4.1 8-bit deMUX mode

When operating the microprocessor at the target speed, it is difficult to toggle ETM pins at such a high speed. The ETM needs to be configured in the 8-bit deMUX mode to achieve tracing at the target speed of 184 MHz. Therefore, 8-bit deMUX mode should be selected for QSC emulation at target speed. In this mode, the trace port analyzer captures the data at half the ARM's clock rate.

The 8-bit deMUX mode configuration is shown in Figure 11-12.

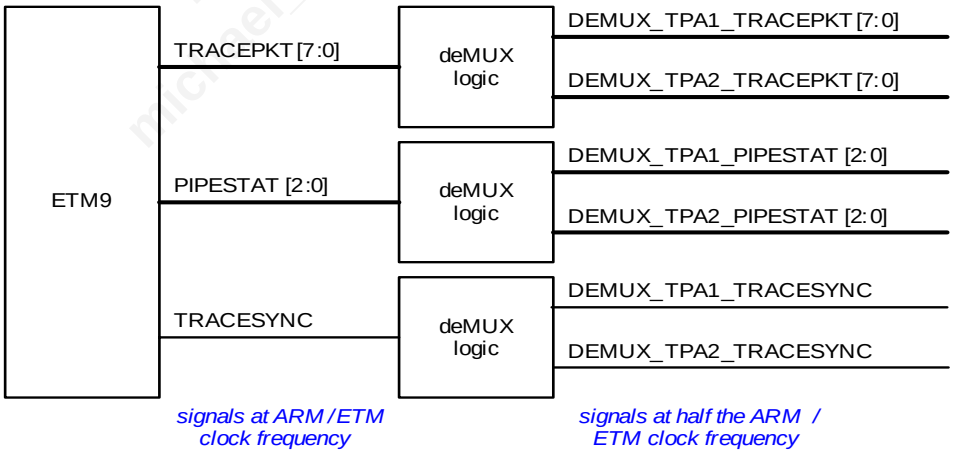


Figure 11-12 ETM 8-bit deMUX mode

The cost of this mode is an additional trace packet analyzer (TPA), four more QSC pins, and less trace (not full).

11.6.4.2 Half-rate clocking mode

The QSC62x0 ETM also supports half-rate clocking mode. When in this mode, the TRACECLK to the TPA is one-half the rate of the ARM clock in normal mode, and one-quarter the rate of the ARM clock in deMUX mode. The primary purpose for the half-rate clocking mode is to reduce the signal transition rate on the TRACECLK pin. When this mode is selected, the TPA samples the trace data signals on both edges of the clock, instead of only the rising edge in full-rate clocking mode.

11.6.5 ETM design considerations

11.6.5.1 ETM interface voltage considerations

As seen in Table 11-9, ETM pins use three different (P1, P5, and P6). These pad groups may be powered off different voltage supplies (P1 is fixed at 1.8 V pad, but P5 and P6 could be 1.8 V or 2.6 V). Therefore, designs that enable and use the QSC ETM interface with an external TPA must use a special connector to level-translate those ETM signals to and from one consistent and appropriate voltage level at the TPA.

11.6.5.2 Additional ETM usage considerations

Because the ETM signals are all grouped as GPIO2, the following steps must be taken to program GPIO2 group properties via the GPIO2 registers when in ETM:

1. First bring up the device in its normal (native) mode.
2. Configure the desired GPIO2 registers for the desired settings (GPIO functionality, drive strength, etc.).
3. Without resetting the device, set the MODE[0] pin appropriately (1) to enable the ETM mode.

The above steps are required since all regular access to the GPIO2 group configuration registers is blocked once ETM mode is enabled. ETM is a special condition — see Chapter 10 for more GPIO information. Therefore, configuration of properties such as drive strength or GPIO configuration will not be possible once ETM is enabled.

11.7 General-purpose PDM (GP_PDM)

The QSC62x0 device includes two general-purpose pulse density modulated (GP_PDM) outputs that can be used for any purpose (such as a backlight control or an analog control voltage after lowpass filtering). GP_PDMs are user-configurable 8-bit or 12-bit PDM signals clocked off the TCXO signal divided by 4 (4.8 MHz).

11.7.1 GP_PDM connections

The two GPIOs that can be configured as GP_PDM outputs are listed in Table 11-11.

Table 11-11 GP_PDM connections

Pin #	Pin name	Pad type [1]	Functional description
T14	GPIO_65 GP_PDM_1	DO, Z	Second of two PDM outputs; 8-bit PDM value, clocked at TCXO/4
C15	GPIO_0 GP_PDM_0	DO, Z	First of two PDM outputs; 12-bit PDM value, clocked at TCXO/4

1 The GP_PDM pad type is DO, Z = digital output with tristate capability.

11.7.2 GP_PDM implementation

The GP_PDM functions are enabled via the TCXO_PDM_CTL register; the PDM output polarity is also configured in this register. After enabling the PDM(s) and choosing the required polarity, the pulse density is programmed using the PDMn_CTL register (where n is 0, 1, or 2).

Table 11-12 gives example PDMn_CTL values and their resulting PDM output formats (assuming noninverted polarity setting).

Table 11-12 Example PDMn_CTL settings and resulting outputs

PDMn_CTL register value	Output signal format
0x80	Low for 256/256 TCXO/4 pulses
0x00	Low for 128/256 TCXO/4 pulses
0x7F	Low for 1/256 TCXO/4 pulses

The register values are in the two's complement format. Therefore the most negative value (0x80 = -128 decimal) produces the least number of ones, the middle value (0x00 = 0 decimal) produces the median number of ones, and the most positive number (0x7F = 127 decimal) produces the greatest number of ones.

11.8 General-purpose clock (GP_CLK)

The QSC62x0 device provides up to three general-purpose clock outputs that can be used for various applications and allows greater flexibility for implementing handset designs. The following general purpose clocks (GP_CLK) are available:

- On pad group 5: pin B9 (GPIO_42) and pin H12 (GPIO_41)
- On pad group 6: pin T14 (GPIO_65)

The GP_CLK output frequency is selected by programming the GP_CLK_NS and GP_CLK_MD registers. The clock source is always PLL0/2 or 184 MHz. The M, N, and D values programmed in the above registers set the divide ratio for the M/N counter and the desired duty cycle. For a 50% duty cycle, set D = N/2. Also, N should not be less than 2xM.

The GP_CLK can be used as a clock source for peripherals such as a camera sensor or FM receiver chip. Unless the frequency is the same, it cannot be used to clock more than one device at a time.

NOTE The M/N:D counter used by the GP_CLK cannot arbitrarily generate any frequency that is a fraction of the original. When noninteger divisions are attempted, jitter is created that can vary from negligible to very severe. Each configured clock output must be tested to verify that any pertinent jitter requirements are not violated, and that a high-quality signal is generated.

NOTE For guaranteed minimal jitter, even integer divisors should be used.

For more information regarding the registers discussed in the above sections, refer to the *QUALCOMM Single-Chip QSC6240/QSC6270 Software Interface Document* (80-VF846-2).

12 Baseband-Analog/RF-PM Interfaces

Many signals that were routed between ICs in previous-generation chipset implementations are now routed completely internally to the QSC62x0 device. Examples include the Rx baseband and Tx baseband signals, status and control buses, and many dedicated status or control signals.

A few signal types previously routed between ICs are brought out of the QSC device to allow external connections and/or components. Examples of these signal types include the power-supply distribution and power-sequence triggering signals. Interfaces such as these are discussed in other chapters within this document.

And finally, some signals are brought off-chip, allowing QSC baseband circuits to:

- Support the GPS receiver and accept its real-time data
- Control external components or functions

These interfaces are not fully addressed elsewhere in this document, so they are addressed below.

12.1 GPS interface

The QSC62x0 devices support GPS position location with the addition of the RGR6240 IC. The interface between these two components is shown in Figure 12-1, and consists of just five lines:

- Real-time data is transferred from the RGR to the QSC device using serial links.
  - GPIO_55 (pin F14) is configured as the in-phase data input GPS_ADCI.
  - GPIO_58 (pin E15) is configured as the quadrature-phase data input GPS_ADCQ.
- The QSC provides the ADC clock for the RGR device: GPIO_57 is configured as GPS_ADCCLK.
- Status and control signaling is implemented via the single-wire serial bus interface (SSBI): GPIO_56 is configured as GPS_SSBI.
- One of the QSC device's general-purpose, buffered TCXO outputs provides the RGR device's frequency reference for its LO synthesizer: pin V18 (XO_OUT_GP2.

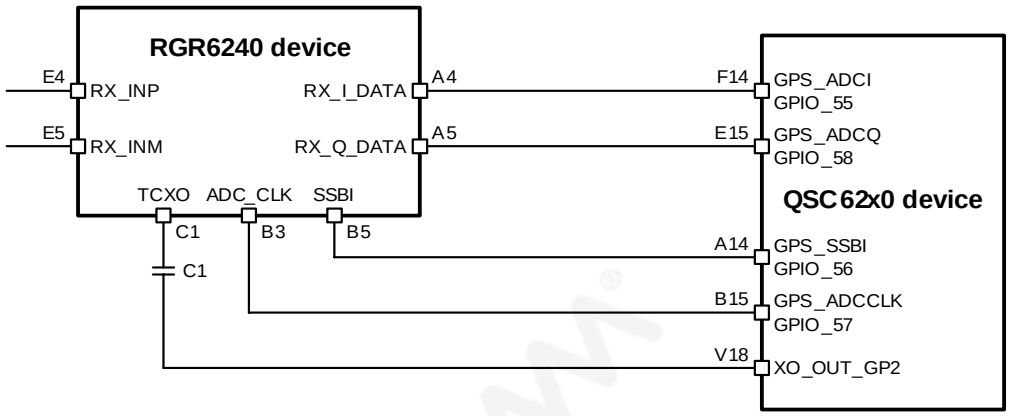


Figure 12-1 GPS connections to/from the RGR6240 IC

12.2 Control of external functions

The QSC62x0 device interfaces with several external functions not fully addressed elsewhere; these interfaces are shown in Figure 12-2 (based on a QCT reference design).

The external control interfaces are:

- UMTS PA range control
  - PA_R0 and PA_R1 (pin C17 and C16) – two GPIOs are configured to control the output power range of the active UMTS PA.
  - The expected logic is:
    - PA_R0 = H and PA_R1 = H -> low-power mode
    - PA_R0 = L and PA_R1 = H -> mid-power mode
    - PA_R0 = H and PA_R1 = L -> invalid setting
    - PA_R0 = L and PA_R1 = L -> high-power mode
  - Both QSC outputs are connected to each UMTS PA, with low-value capacitors located near each PA input pin for control-line filtering.
- Antenna-switch channel selection
  - ANT_SEL_0, ANT_SEL_1, ANT_SEL_2, and ANT_SEL_3 – four GPIOs are configured to select which transceiver path is connected to the antenna.
  - Logic depends upon the number of bands supported by a given application and the switch module being used
  - All control signals should be direct connections, with low-value capacitors located near the switch input pins for control-line filtering.

- If a VCTCXO module is used, it is tuned via pin T14, a GPIO configured as TRK_LO_ADJ. A two-pole RC filter must be included between the QSC control output and the VCTCXO input. In addition, a digital buffer or D flip-flop might be required at the QSC output. Conditions requiring this added circuitry are discussed in *radioOne Solutions for GPS Position Location Application Note*(80-V1647-1).

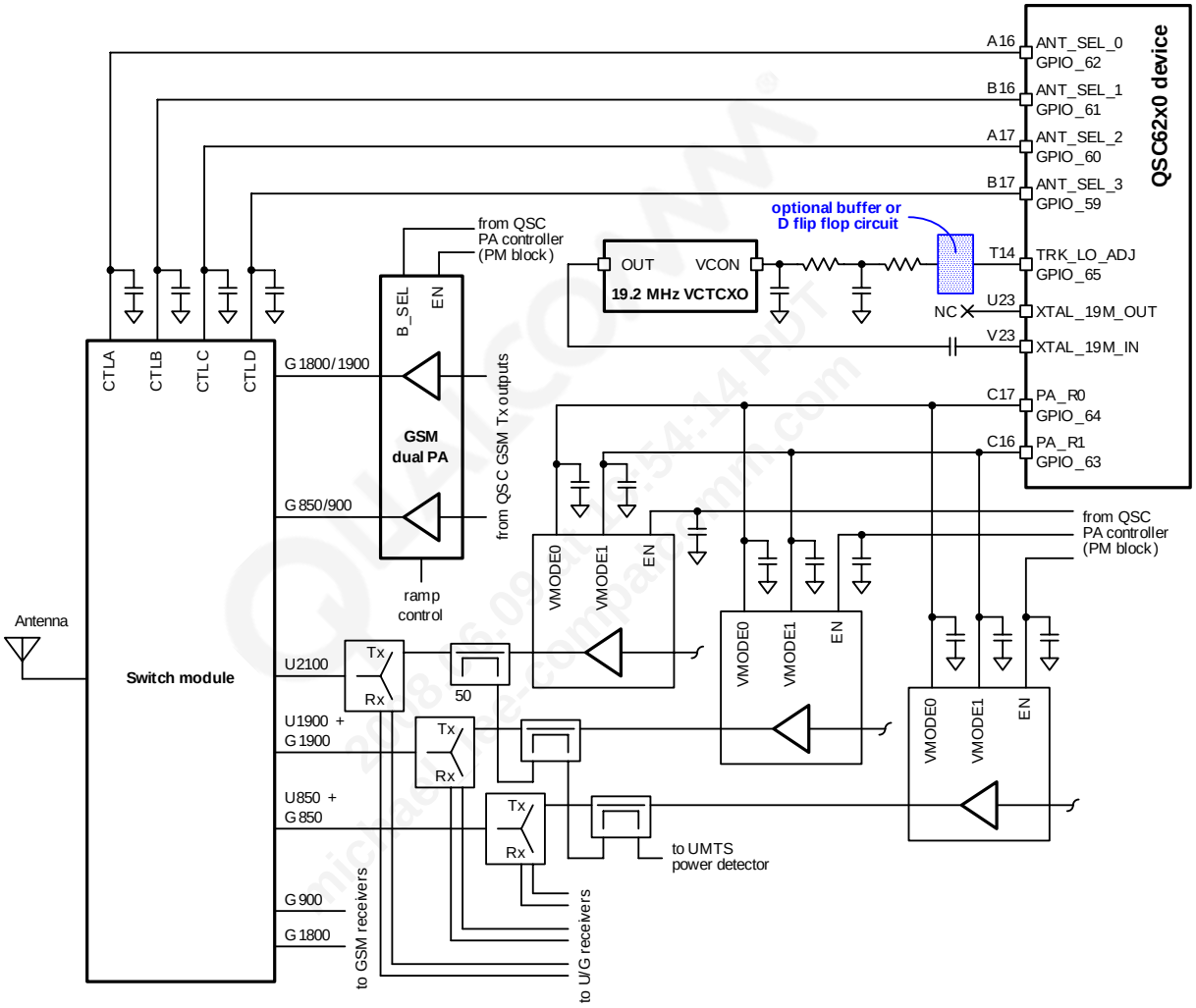


Figure 12-2 External control interfaces

NOTE The VCTCXO module might not be required. See the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* (80-VF846-5) for details regarding the 19.2 MHz reference implementation.

As noted in Figure 12-2, additional UMTS and GSM power amplifier controls are provided by the QSC power management circuits; see Section 16.5 for details.

12.3 External function connections

The QSC62x0 connections to external functions discussed above are summarized in Table 12-1.

Table 12-1 Connections to external functions

Pin name/function	Pin #	Pad group	Pad type [1]	Drive current [2]	Functional description
GPS receiver interface					
GPS_ADCI (GPIO_55)	F14	P5 (1.8 V)	DI	–	GPS Rx data I bit from RGR6240 IC
GPS_ADCQ (GPIO_58)	E15	P5 (1.8 V)	DI	–	GPS Rx data Q bit from RGR6240 IC
GPS_ADCCLK (GPIO_57)	B15	P5 (1.8 V)	DO	1-8 (1)	GPS ADC sampling clock to RGR6240 IC
GPS_SSBI (GPIO_56)	A14	P5 (1.8 V)	B	1-8 (1)	SSBI for GPS status and control (RGR6240)
XO_OUT_GP2	V18	–	DO	–	TCXO clock to RGR6240 IC
Power amplifier controls					
PA_RANGE0 (GPIO_64)	C17	P6 (1.8/2.6 V)	DO	1-8/2-16 (1/2)	Controls UMTS PA output power range; R0 = 1, R1 = 1 sets low-power mode R0 = 0, R1 = 1 sets medium-power mode R0 = 0, R1 = 0 sets high-power mode
PA_RANGE1 (GPIO_63)	C16	P6 (1.8/2.6 V)	DO	1-8/2-16 (1/2)	
VCTCXO control					
TRK_LO_ADJ (GPIO_65)	T14	P6 (1.8/2.6 V)	DO	1-8/2-16 (1/2)	PDM output from frequency tracking loop; for VCTCXO control if needed
Antenna switch controls					
ANT_SEL_0 (GPIO_62)	A16	P6 (1.8/2.6 V)	DO	1-8/2-16 (1/2)	Antenna band select bit 0
ANT_SEL_1 (GPIO_61)	B16	P6 (1.8/2.6 V)	DO	1-8/2-16 (1/2)	Antenna band select bit 1
ANT_SEL_2 (GPIO_60)	A17	P6 (1.8/2.6 V)	DO	1-8/2-16 (1/2)	Antenna band select bit 2
ANT_SEL_3 (GPIO_59)	B17	P6 (1.8/2.6 V)	DO	1-8/2-16 (1/2)	Antenna band select bit 3

[1] The parameters listed under the Pad type column are defined in the QSC62x0 device specification (80-VF846-1).

[2] Some pins that can be configured as digital outputs allow their output drive current to be programmed. The programmable range is listed in this column, with the recommended value in parentheses. If the pin operates off two voltages, two ranges and recommended values are listed. For example, 1-8/2-16 (1/2) means the programmable range is 1 to 8 mA at the lower supply voltage, with a recommended value of 1 mA; and the programmable range is 2 to 16 mA at the higher supply voltage, with a recommended value of 2 mA. As implied, the actual current realized is dependent upon supply voltage.

13 Input Power Management

The QSC62x0 input power-management circuits (Figure 13-1) accept power-supply inputs from two handset-external power sources:

- A wall charger at pin AA14 and AB14 (VCHG)
- A USB supply at pin AB7 (USB_VBUS)

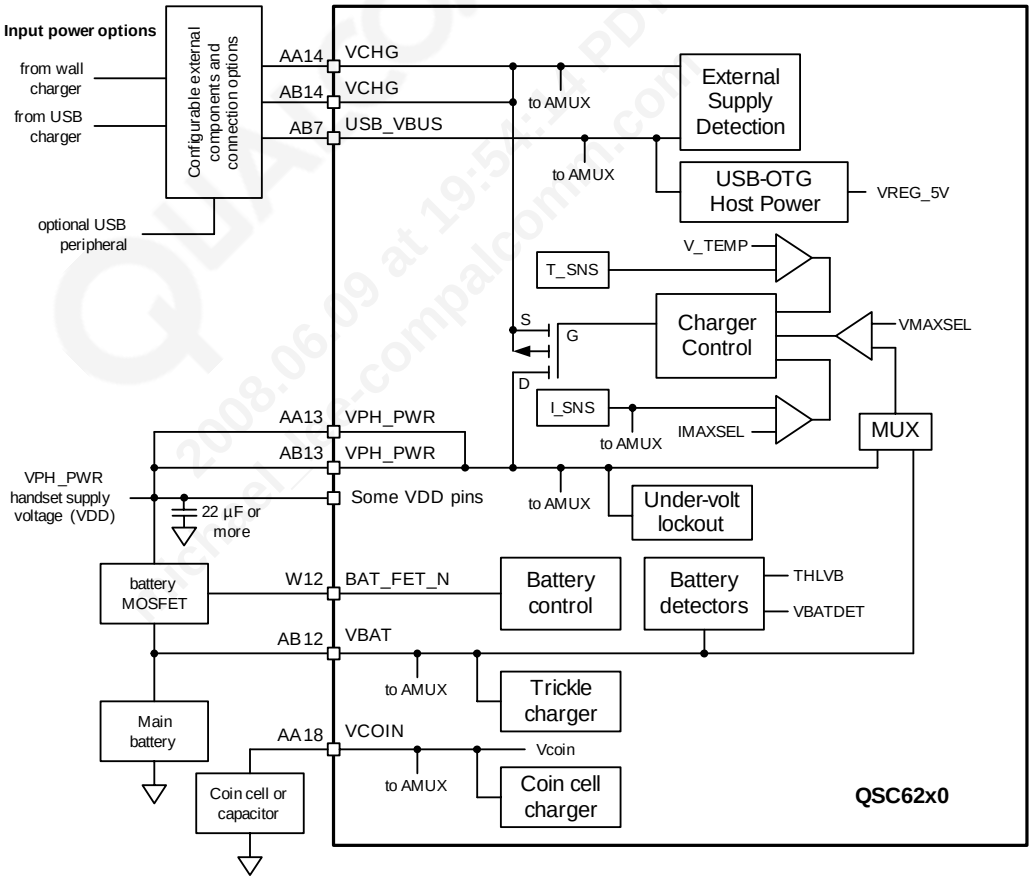


Figure 13-1 Input power-management functional block diagram

Depending upon external circuitry and connections, the QSC device can accept primary power from either one of these sources, or both could be attached simultaneously (with the charger being selected as the primary power source). When a wall charger is used as the primary power source, an external USB device can still be used to power the handset's USB transceiver circuits. Voltages at VCHG, USB_VBUS, and VPH_PWR nodes are continually monitored to automatically detect valid power source connections and disconnections.

In addition to the external supply, handset electronics can be powered by the main battery. A coin cell is supported as a keep-alive power source for essential QSC circuits when an external source is not connected and the battery is depleted.

The QSC's input power-management functions and handset software coordinate these sources — detecting which sources are applied, verifying that they are within acceptable operational limits, and orchestrating battery and coin-cell recharging while maintaining the handset electronics supply voltages.

When an external supply, charger-type or USB-type, is connected to the handset, its voltage (at VCHG) is measured and qualified by on-chip detector circuits. Transistors controlled by QSC drivers open and close connections between an external supply and the handset operating voltage (V_{DD}), and between V_{DD} and the main battery. With appropriate connections, the supported external supply can power the handset electronics only or simultaneously charge the main battery, and the main battery can be charged by an external supply or provide power to handset electronics.

If desired, USB charging can be enabled by making appropriate connections between the VCHG and USB_VBUS pins. If a wall charger is never used, USB_VBUS and VCHG pins can be shorted together to accomplish battery charging from a USB supply. If options for both a wall charger and a USB supply are supported, a detection mechanism must be implemented. The mechanism should detect the presence of a USB supply (no wall charger) and connect USB_VBUS and VCHG pins through an external FET, or detect the presence of a wall charger (no USB supply) and bypass the FET, connecting directly to the VCHG pin. If both a wall charger and a USB supply are connected, the charger should be the main power source and the USB supply should power only the handset's USB transceiver circuits. For details, see *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* (80-VF846-5).

Power management circuits switch between the power supply sources automatically without software intervention. If more than one valid source is available, the circuits select the external supply (charger or USB) first, then switch to the main battery if the external supply is removed.

This chapter presents the input power-management features and connections, then discusses the following topics:

- Input circuits in general
- External supply detection
- Sourcing power for USB peripherals
- USB charging (optional)
- Charging pass transistor controls and thermal limiting
- Voltage regulation at VDD and VBAT nodes
- Current regulation, monitoring, and overcurrent protection
- Main battery charging, including autonomous mode
- Coin-cell charging
- Battery voltage detection and alarms, UVLO, and SMPL
- V_{DD} collapse protection

13.1 Input power-management features

- Valid external supply attachment and removal detection
- Unregulated (closed-loop) external charger supply as input power source
- Integrated PFET charging pass transistor; eliminated sense resistor
- Support for lithium-ion and lithium-ion polymer main batteries; nickel-based batteries are not supported
- Trickle, constant current, constant voltage, and pulse charging of the main battery
- Autonomous charging option – driven by an on-chip state machine without software intervention
- Software-controlled charging option – backwards-compatible with previous QSC and PM products
- Coin-cell battery (including charging)
- Battery-voltage detectors with programmable thresholds
- V_{DD} collapse protection
- Charger-current regulation and real-time monitoring for overcurrent protection
- Charger-transistor protection by thermal control
- Control drivers for the internal charging PFET and external battery PFET
- Voltage, current, and thermal control loops
- Automated recovery from sudden momentary power loss (requires external 32.768-kHz crystal)

13.2 Input power-management connections

All QSC62x0 pins associated with its input power-management functions are listed in Table 13-1.

Table 13-1 Input power-management connections

Pin name/function	Pin #	Pin type [1]	Functional description
VCHG	AA14, AB14	AI	External supply voltage; connect both pins directly to the external power supply (such as a wall charger). Connect the immediate 2.2-µF capacitor and 47-k resistor to ground. **NOTE**: This is a high current input since the charging pass transistor is internal.
USB_VBUS	AB7	AI, AO	This pin is configured as an analog input or an analog output depending upon the type of peripheral device connected. Connect the immediate 2.2-µF capacitor and 47-k resistor to ground.
VBAT	AB12	AI, AO	This pin is used as an analog input to sense the main battery voltage or as an analog output that sources trickle-charging current for the battery.
VCOIN	AA18	AI, AO	Used as an analog input from the 3-V coin cell for SMPL, RTC, and crystal oscillator keep-alive power; a capacitor (rather than a coin cell) can be used if only SMPL is supported (not RTC or XTAL). Used as an analog output for coin cell or capacitor charging.
BAT_FET_N	W12	AO	Battery MOSFET control output – connect directly to the gate. Pulls up to the higher of V_{DD} or V_{BAT}.
VPH_PWR	AA13, AB13	AO	Primary phone power supply voltage node; connects internally to the charging pass transistor and to the sense point for V_{DD} voltage regulation. **NOTE**: This is a high current output since the charging pass transistor is internal.

[1] Pin type is AI = analog input or AO = analog output.

13.3 Input circuits overview

New QSC62x0 features simplify the external circuit requirements: an internal P-channel MOSFET eliminates the external charging pass transistor, and a current sensor eliminates the sense resistor. A typical (baseline) input power management application is shown in Figure 13-2 as an example. Important characteristics of the external components are discussed in the following paragraphs. Other input power source options are discussed in the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* document (80-VF846-5).

Shunt resistors and capacitors

The charger input should always include a shunt RC network to ground. The capacitor (2.2 µF) provides filtering, improved stability, and assures electromagnetic compatibility. The resistor (47 k) pulls the VCHG node to ground when an external supply is not connected, providing faster detection of connects and disconnects. These values also fulfill USB requirements (in the case of a single USB source).

If USB functions are supported separate from the charger input, the USB_VBUS input should have its own 47 k, 2.2 µF network to ground (as shown in the example).

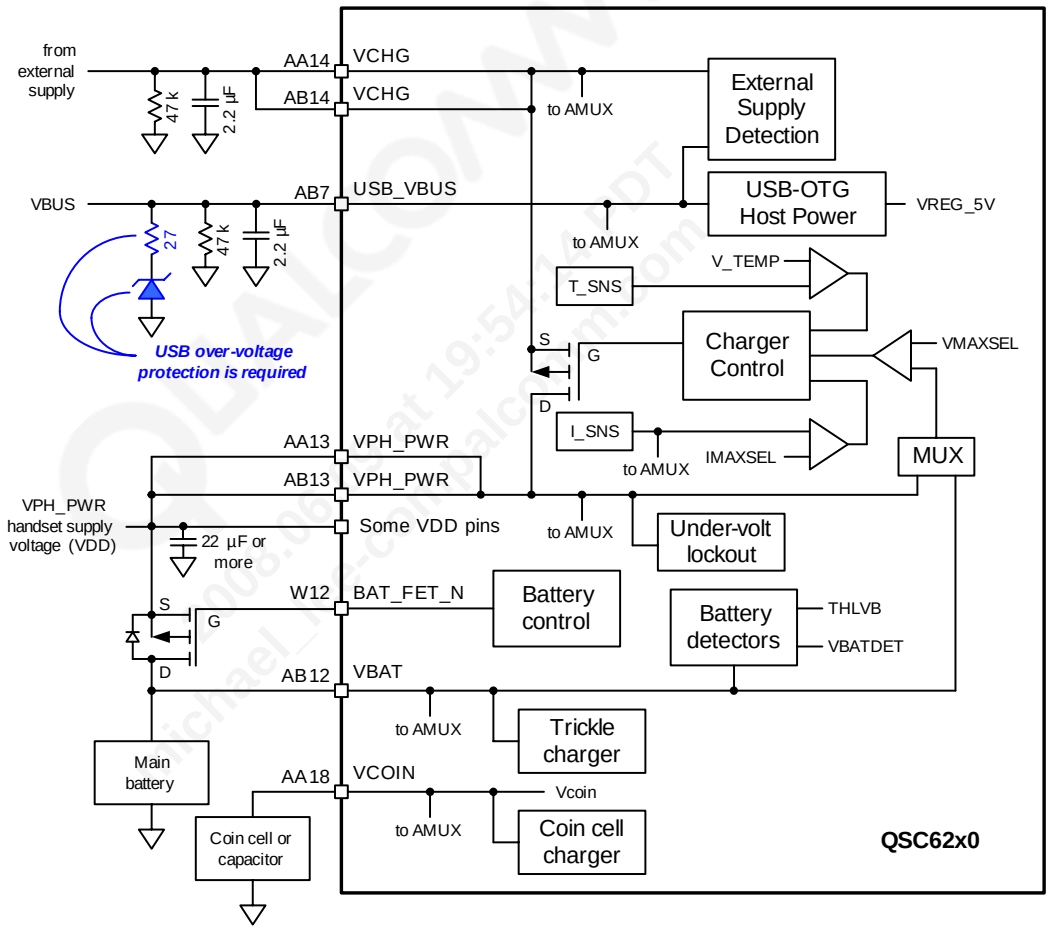


Figure 13-2 Input circuits schematic diagram

Bulk capacitor

A high-value bulk capacitor is required at the VPH_PWR node (or VDD node). This capacitor must be at least 22 µF to ensure V_{DD} regulation stability and must be located near pins AA13 and AB13 (VPH_PWR) — near the internal charging pass transistor and in the high current path.

VBUS voltage clamp

Overvoltage protection is required at pin AB7 (USB_VBUS) to limit its input voltage to 5.25 V. This is critical since USB standards allow 5.6 V, which is beyond the QSC device's rating at this pin.

Battery transistor (required for trickle charging)

The battery transistor must be a P-channel MOSFET — a bipolar device cannot be used. This transistor provides a second switching function (beyond the charging pass transistor) and is controlled by the BAT_FET_N signal (pin W12) to connect and disconnect the main battery to and from V_{DD}. When the transistor is off (BAT_FET_N approaches V_{DD}, $V_{GS} \sim 0$ V, and the transistor becomes a nearly open circuit), the battery is disconnected from the handset electronics and the external supply, so the battery is not providing handset power or being charged.

NOTE V_{DD} is at a significant voltage even when the QSC device is turned off due to forward biasing of the battery MOSFET body diode. Therefore, when the device is off, all internal loads must be disconnected or powered down — the design cannot rely on V_{DD} going to 0 V.

With BAT_FET_N near 0 V, $V_{GS} \sim -V_{DD}$, the transistor becomes a nearly shorted circuit, and the battery is connected to the VDD node. This condition can serve different functions, depending on other circuit elements and software controls:

- Main battery charging occurs if a valid supply is connected and the charging pass transistor is on. If the battery transistor is on when a valid supply is detected, it remains on until directed otherwise via software or optionally turned off by software programming.
- If the charging pass transistor is off (whether a valid external supply is present or not), the main battery provides power to the handset electronics.

Regardless of the battery MOSFET's state, the main battery voltage (V_{BAT}) is always connected to battery detector circuits. In addition, V_{BAT} is sometimes connected to the housekeeping ADC through the PM circuit's analog multiplexer. The battery detector provides real-time and precise measurements of the battery voltage, but provides coarse resolution. The HKADC is used for higher resolution measurements but requires software interaction (and the resulting delay).

Whenever the battery MOSFET is open, a differential circuit senses the voltage between VPH_PWR (V_{DD}) and VBAT and automatically turns the battery MOSFET on if V_{DD} dips 10 mV below V_{BAT}. This is true only if V_{BAT} is greater than the UVLO threshold (2.55 V). See Section 13.17 for details.

13.4 External supply detection

The QSC62x0 device monitors external supply voltages that are connected directly to VCHG (pins AA14 and AB14) and USB_VBUS (pin AB7), and the handset supply voltage at VPH_PWR (pins AA13 and AB13). See Figure 13-2 for the external schematic diagram. Internal detector circuits measure these voltages to recognize when external supplies are connected or removed, and verify they are within their valid ranges when connected. Hysteresis prevents undesired switching near the thresholds, and interrupts notify the software of input voltage conditions. The nominal thresholds are:

- V_{CHG} detection — 3.3 to 14.5 V
- V_{USB} detection — 3.3 V minimum

Detecting an external supply removal considers the voltage drop across the charging pass transistor (V_{CHG} - V_{DD} or V_{USB} - V_{DD}). If this voltage drop is less than 30 mV or so the charging pass transistor is turned off regardless of the external supply voltage. This prevents reverse operation of the charging pass transistor that could leave sufficient voltage at VCHG or USB_VBUS to prevent the device from realizing the external supply was disconnected.

The QSC62x0 device responds automatically to external power supply changes while reporting its conditions to software using interrupts. The software monitors the V_{CHG} and V_{USB} voltages through the analog multiplexer and controls power management functions accordingly, such as proceeding with battery charging.

13.5 USB_VBUS power source

A variety of peripheral devices can be connected to the phone's USB port. QSC62x0 circuits not only detect when a peripheral device is connected, but they determine its type as well (see Section 13.7 for details). If an OTG type-B device is connected, the QSC device provides a +5 V power source at the USB_VBUS pin capable of driving 200 mA, and the USB charger function cannot be used.

13.6 Option: USB charging

If desired, USB charging can be enabled on QSC62x0 devices by externally connecting the USB_VBUS (AB7) and VCHG (AA14, AB14) pins. Special consideration for maximum charger voltage and IMAXSEL will need to be taken in this configuration. For details, see the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* (80-VF846-5).

13.7 USB host/charger detection and single-path charging

The QSC62x0 device includes integrated switches and resistors (Figure 13-3) that allow proper logic levels to be established and detected when a USB host or a charger is connected. Both connections to the QSC's USB_DP and USB_DM pins are shown:

- The USB host connections are shown at the upper left in pink.
- The charger connections are shown at the lower left in light blue.

When the QSC device detects a valid voltage on its USB_VBUS pin, it closes on-chip switches that make the following internal connections:

- At its USB_DP pin: enables a pull-up resistor (R_DP_UP, 125 k) to VREG_USB.
- At its USB_DM pin: enables a pull-down resistor (R_DM_DN, 375 k) to ground.

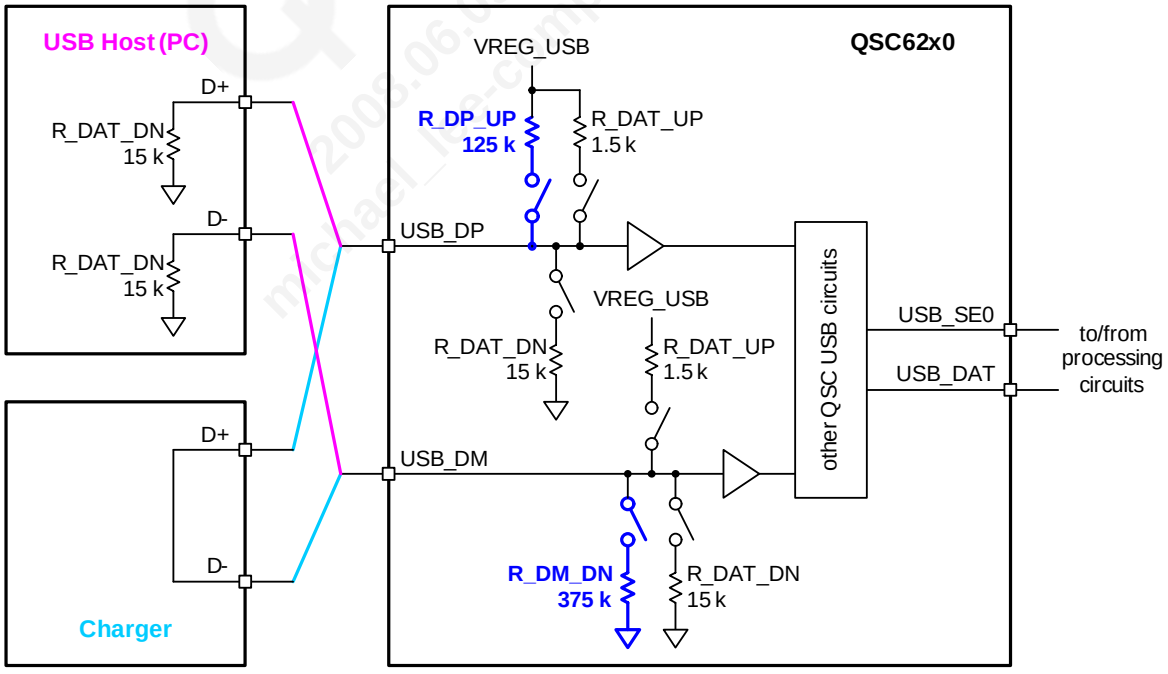


Figure 13-3 Integrated circuits for detecting USB host or charger

After a short delay that ensures the plug is fully inserted — serving a debounce function — software checks the logic levels detected at the two pins. The readings determine whether a host or charger is attached as follows:

- If a USB host is attached:
  - The USB host's R_DAT_DN pull-down resistor (15 k) at D+ is strong enough to force a logic low at the USB_DP pin, despite the internal 125 k pull-up resistor.
  - The USB host's R_DAT_DN pull-down resistor (15 k) at D- reinforces the 375 k pull-down resistor at the USB_DM pin, yielding a logic low.
  - The two logic low conditions is decoded by software to mean that a USB host is attached.
  - USB_DP = low and USB_DM = low --> USB host is attached.
- If a USB charger is attached:
  - The USB charger provides a short circuit between its D+ and D- pins that shorts the QSC's USB_DP and USB_DM pins together.
  - A voltage divider is created that presents the following voltage at both QSC pins:

    V_IN = VREG_USB x 375k / (125k + 375k)
  - This results in a logic high at both QSC pins (USB_DP and USB_DM).
  - The two logic high conditions are decoded by software to mean that a USB charger is attached.
  - USB_DP = high and USB_DM = high --> USB charger is attached.
- If nothing is attached, a logic high is detected at USB_DP and a logic low is detected at USB_DM. Software decodes this condition to mean either nothing or an unknown device is attached.

Once software identifies the USB device type, it enables single-path charging as follows:

- If a USB host is recognized: the phone can draw current from the USB host, but cannot initially exceed 100 mA (as specified in the USB standard).
- If a USB charger is recognized: the phone can draw as much current as the charger can provide (the USB standard's initial limit of 100 mA does not apply).

13.8 Charging pass transistor controls and thermal limiting

A control driver for the charging pass transistor is included within the QSC device; the driver output is applied internally to an on-chip P-channel MOSFET. The QSC62x0 device uses closed-loop control of the charging pass transistor to regulate the V_{DD} voltage during normal operation, the detected current (I_{DET}) during fast (constant current) charging if needed, or the battery voltage (V_{BAT}) during the final stages of charging. The resistance of the charging pass transistor can also be increased for overcurrent protection.

See Section 13.9 for more details about V_{DD} and V_{BAT} regulation, and Section 13.10 for current regulation, monitoring, and overcurrent protection.

Controlling the charging pass transistor also allows for overheating protection: the QSC device monitors the charging pass transistor temperature and reduces its through current if the temperature exceeds an acceptable limit.

Handset designers should consider the following points to help reduce charging pass transistor power dissipation and heating:

- Minimize the voltage across the charging pass transistor by using an external supply voltage that is only slightly above the top-off voltage of the Li-ion battery.
- Design the charger voltage compliance so its output voltage collapses during fast charging, thereby reducing the voltage across the charging pass transistor.

If thermal limiting occurs and the V_{DD} voltage collapses momentarily below V_{BAT}, the QSC620x device will close the battery MOSFET to prevent a phone shutdown. See Section 13.17 for details.

13.9 Voltage regulation (V_{DD} or V_{BAT})

The QSC62x0 device provides closed-loop control of the charging pass transistor to regulate either the handset supply voltage (V_{DD}) when not charging or the main battery final voltage (V_{BAT}) when charging. When fast charging is disabled, the battery MOSFET is opened and the voltage regulation point is the VPH_PWR pin (V_{DD}). When fast charging is enabled, the VBAT pin is the voltage regulation point, thereby improving the voltage precision of the fully charged battery (Li-ion battery manufacturers specify 1% accuracy).

The V_{DD} control loop that runs off the charger external supply is highlighted in Figure 13-4. The V_{DD} control loop input is the programmable value VMAXSEL, the target or desired voltage; valid settings include 4.000 to 4.375 V in 25-mV steps. The error amplifier generates an error signal by magnifying the difference between the target signal (VMAXSEL) and the actual output signal (V_{DD}).

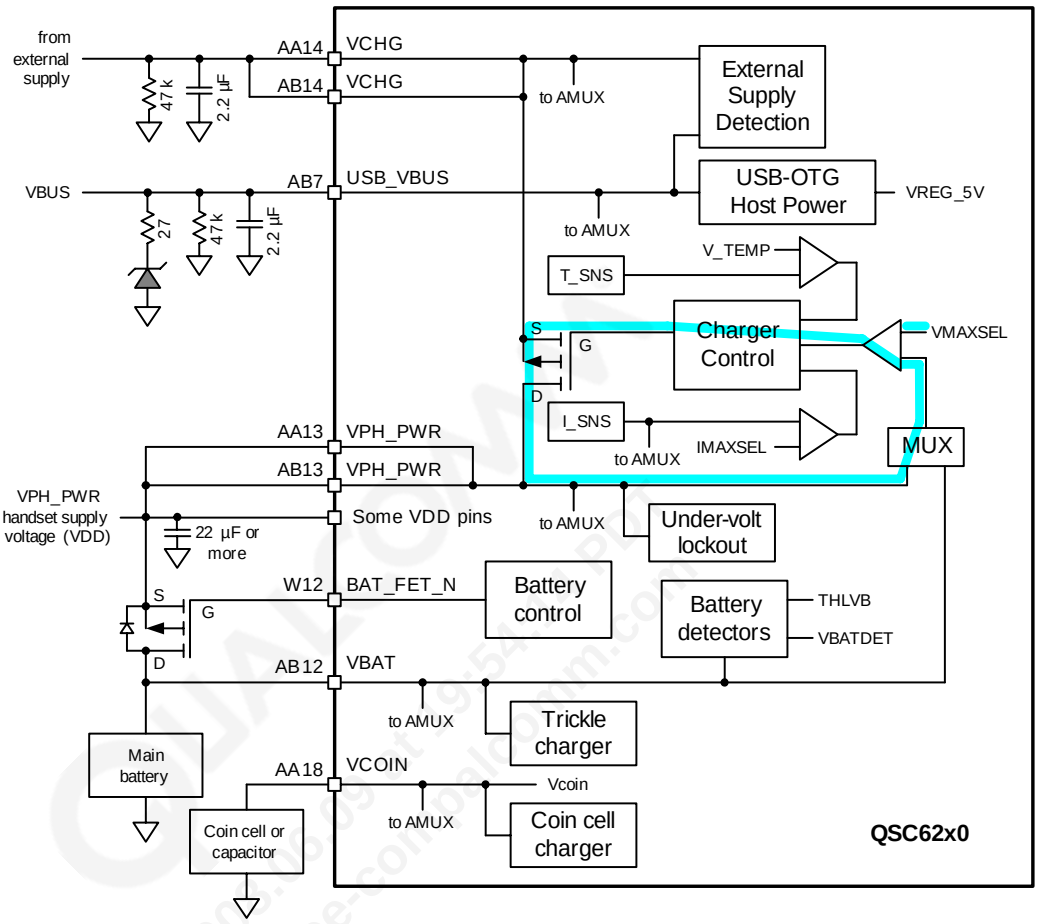


Figure 13-4 V_{DD} regulation loop

The error signal is conditioned within the charger control block and applied to the charging pass transistor. This charging pass transistor is the controlled element within the loop — its gain (on-resistance) varies with the applied control signal. The control loop automatically adjusts the charging pass transistor until the error approaches zero, resulting in an output equaling the input (V_{DD} = VMAXSEL). The current sensor is within the loop, can be lumped with the varying charging pass transistor, and therefore does not impact the regulated V_{DD} accuracy.

During closed-loop V_{DD} regulation, the control signal is varied continuously as needed to force V_{DD} to equal VMAXSEL.

The V_{BAT} regulation loop is identical to the V_{DD} loop except that the multiplexer selects the voltage at the VBAT pin. During V_{BAT} regulation, the programmable value VMAXSEL settings are 4.000 V to 4.375 V in 25-mV steps. See Section 13.12.3 for more information.

13.10 Current regulation, monitoring, and protection

The QSC62x0 device constantly monitors the total handset electronics plus charging current using an on-chip sensor. If the programmed current threshold is exceeded, the charging pass transistor is forced to a higher resistance, disrupting V_{DD} or V_{BAT} regulation but protecting against excess current. The current monitor defaults to the maximum current setting on powerup. This minimizes the risk of a large in-rush load collapsing V_{DD}, but the phone designer must ensure that the initial USB current specification of 100 mA is met.

The same circuits are used to regulate the total handset electronics plus charging current (if needed) during the main battery's constant current charging mode. Figure 13-5 highlights the current-control loop running off the charger external supply.

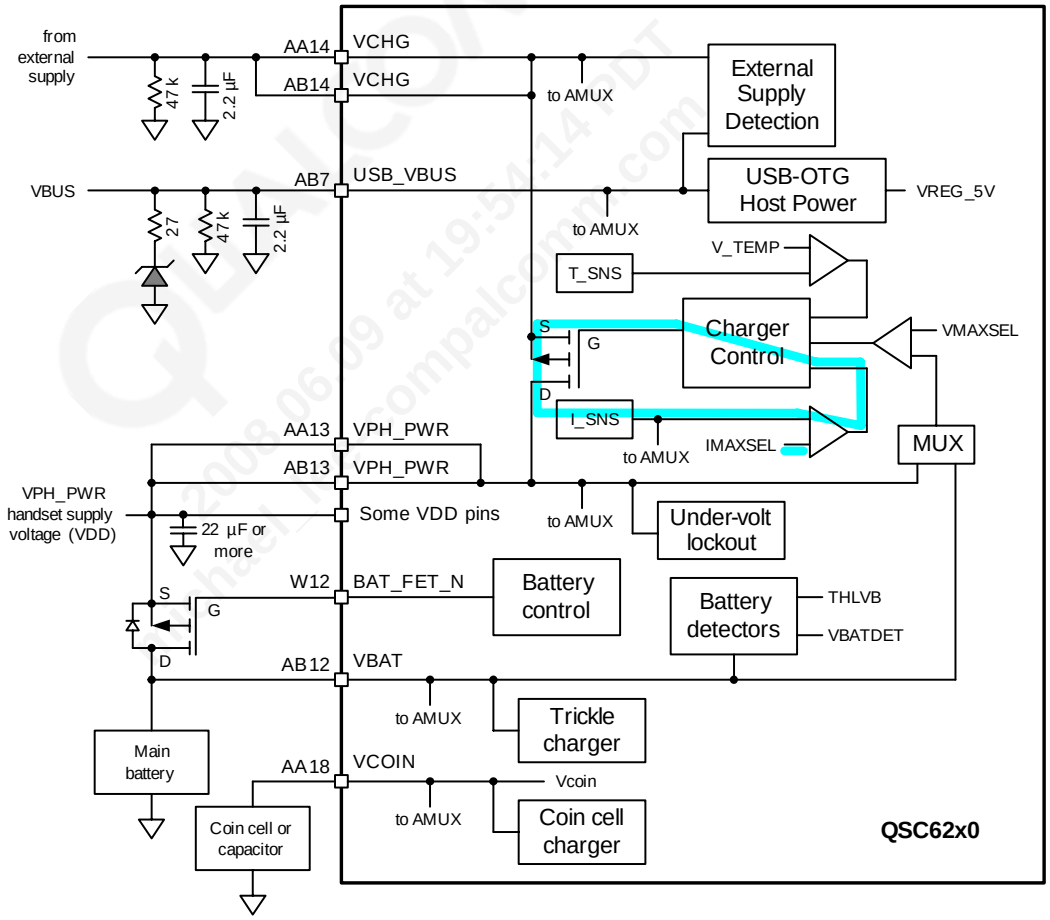


Figure 13-5 Current regulation loop

This control loop input is IMAXSEL, the programmable target voltage. The controlled output is current sensed by the charging pass transistor's current sensor. The error amplifier generates an error signal by magnifying the difference between the target signal (IMAXSEL) and the actual output signal (I_{DET}). Valid IMAXSEL settings range from 100 mA to 2.0 A in 100-mA steps.

Similar to voltage regulation, the error signal is conditioned within the charger control block and applied to the internal charging pass transistor. This charging pass transistor is the controlled element within the loop — its gain (on-resistance) varies with the applied control signal. The control loop automatically adjusts the charging pass transistor until the error approaches zero, resulting in an output equaling the input (IDET = IMAXSEL). The current sensor generates the controlled parameter (I_DET) and impacts the regulation accuracy directly.

13.11 Thermal regulation

Thermal regulation of the charging loop is a new feature. Specifications and operational details are TBD. A general summary is presented in this section.

The charger pass device temperature control features two different temperature regulation techniques: a continuous control and an on/off type control. The first technique is suitable for noncollapsing external chargers (such as USB chargers) and continuously adjusts the pass device current so that the temperature does not exceed a given value TMAX. The second method turns the pass device off when a value TMAX,HI is reached and then turns it back on when the temperature has fallen to a lower value TMAX,LOW. This technique is advised when collapsing external chargers are used. With this type of charger, lowering the pass device current to limit the temperature would cause the external charger voltage to increase and the power dissipated in the pass device (and hence the temperature) would not necessarily decrease. In the worst case, oscillations of the charging voltage and current could occur.

13.12 Main battery charging

The QSC62x0 device provides two methods for lithium-ion and lithium-ion polymer battery charging:

1. A software-controlled method that is identical to previous generations of QSC and PM products
2. An autonomous method - where the PM circuits conduct charging under the direction of a state machine without software intervention

As mentioned earlier, nickel-based batteries are not supported.

When using the software-controlled method, the QSC62x0 device provides as many as four charging techniques: trickle, constant current, constant voltage, and pulse. Battery voltage, external supply voltage, and total detected current measurements are available to software through the analog multiplexer. This allows software to monitor charging parameters, make decisions, and control the charging process.

Autonomous charging

In addition to software control of battery charging, the QSC62x0 device includes a new feature that allow its power management functions to control charging without software intervention. This feature is called autonomous charging and is discussed in

Section 13.12.5. The descriptions within Section 13.12.1 through Section 13.12.4 use the software control method as examples.

Software-controlled charging

Charging of a severely depleted battery (Figure 13-6) begins with trickle charging, a mode that limits the current and avoids pulling V_{DD} down. Once a minimum battery voltage is established using trickle charging, constant current charging is enabled to charge the battery quickly — this mode is sometimes called fast charging. Once the Li-ion battery approaches its target voltage (through constant current charging), the charge is completed using either constant voltage or pulse charging. Further discussions of all charging modes are provided in the sections identified within Figure 13-6.

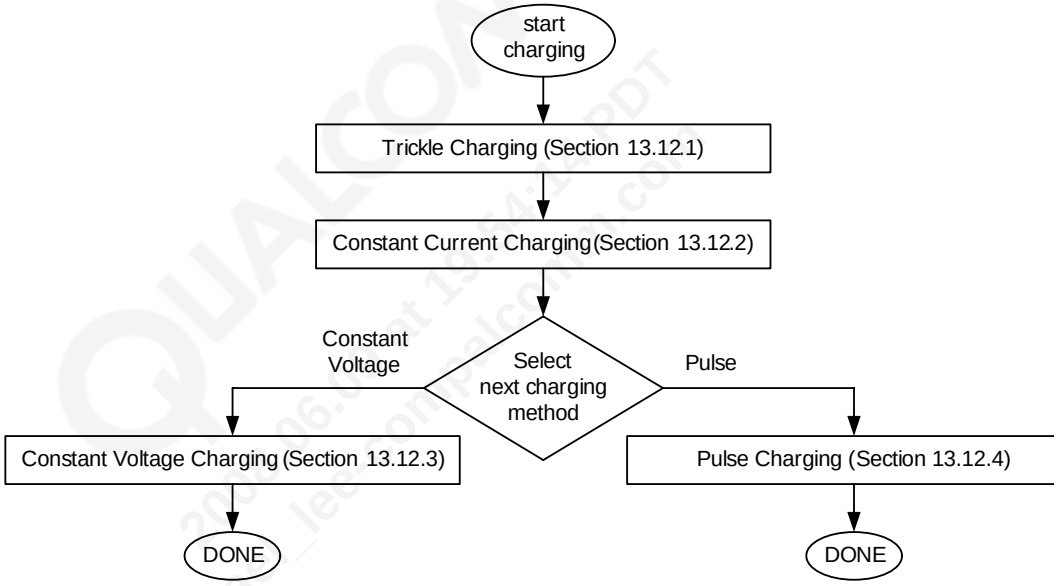


Figure 13-6 Charging sequence for a severely depleted battery (SW-control)

13.12.1 Trickle charging

Trickle charging of the main battery, powered from V_{DD}, is provided by the QSC62x0 device. This mode is used to raise a severely depleted battery's voltage to a level sufficient to begin fast charging.

Fast charging with a high-current supply should not be attempted on a deeply discharged battery — the battery would draw excessive current, pull the V_{DD} voltage down, and possibly cause a handset malfunction or shutdown due to an undervoltage lockout condition. If the QSC device implements current limiting, the excessive current would combine with the potentially large voltage drop across the charging pass transistor to generate unwanted heat within the handset as well. To avoid these problems the QSC provides a constant, low-current charging mode — trickle charging.

The trickle charger is an on-chip programmable current source that supplies current from V_{DD} to pin AB12 (V_{BAT}). Trickle charging can be enabled through software and should be used until the main battery reaches its desired threshold, usually about 3.0 V for Li-ion

batteries. The threshold varies with battery type and application, so there is no predefined value implemented in the detection circuits. Software must terminate trickle charging based on battery voltage measurements at the HKADC (routed through the PM circuit's analog multiplexers) and the battery type — there is no preset termination threshold.

The charging current path and voltage monitoring path are discussed in Figure 13-7.

Software sets the desired trickle charging current. Valid settings are from 10 to 160 mA, in 10 mA increments.

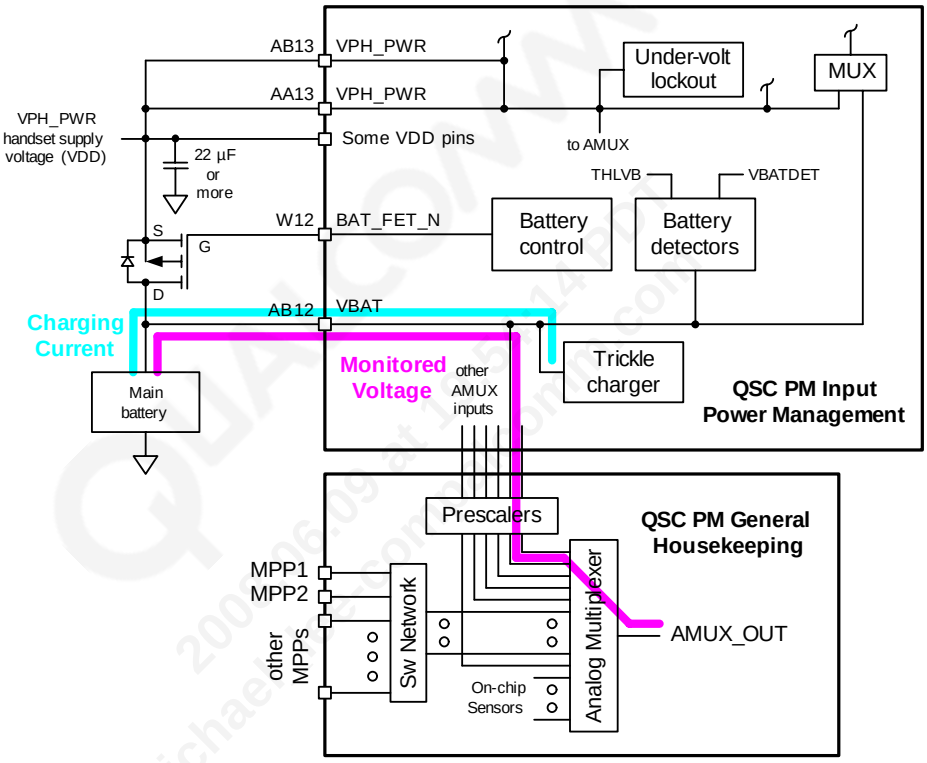


Figure 13-7 Trickle charging

13.12.2 Constant current charging

The QSC62x0 device supports constant current charging of the main battery by closing the battery MOSFET (connecting the battery to V_{DD}), and closed-loop controlling the charging pass transistor. If current limiting is not implemented by the external supply, the closed-loop charging pass transistor control regulates the total current (handset electronics (plus charging current) to match the programmed value (IMAXSEL). Software monitors the charging process as described earlier and continues the constant current mode until the battery reaches its target voltage. Charging of Li-ion batteries require further charging using constant voltage or pulse techniques for top-off and maintenance.

Constant current charging operation is shown in Figure 13-8 with both the charging current and the monitored voltage highlighted. The charging pass transistor is *on* — continuously closed-loop controlled to regulate the total current. The battery MOSFET is fully on, thereby connecting the battery to the VDD node.

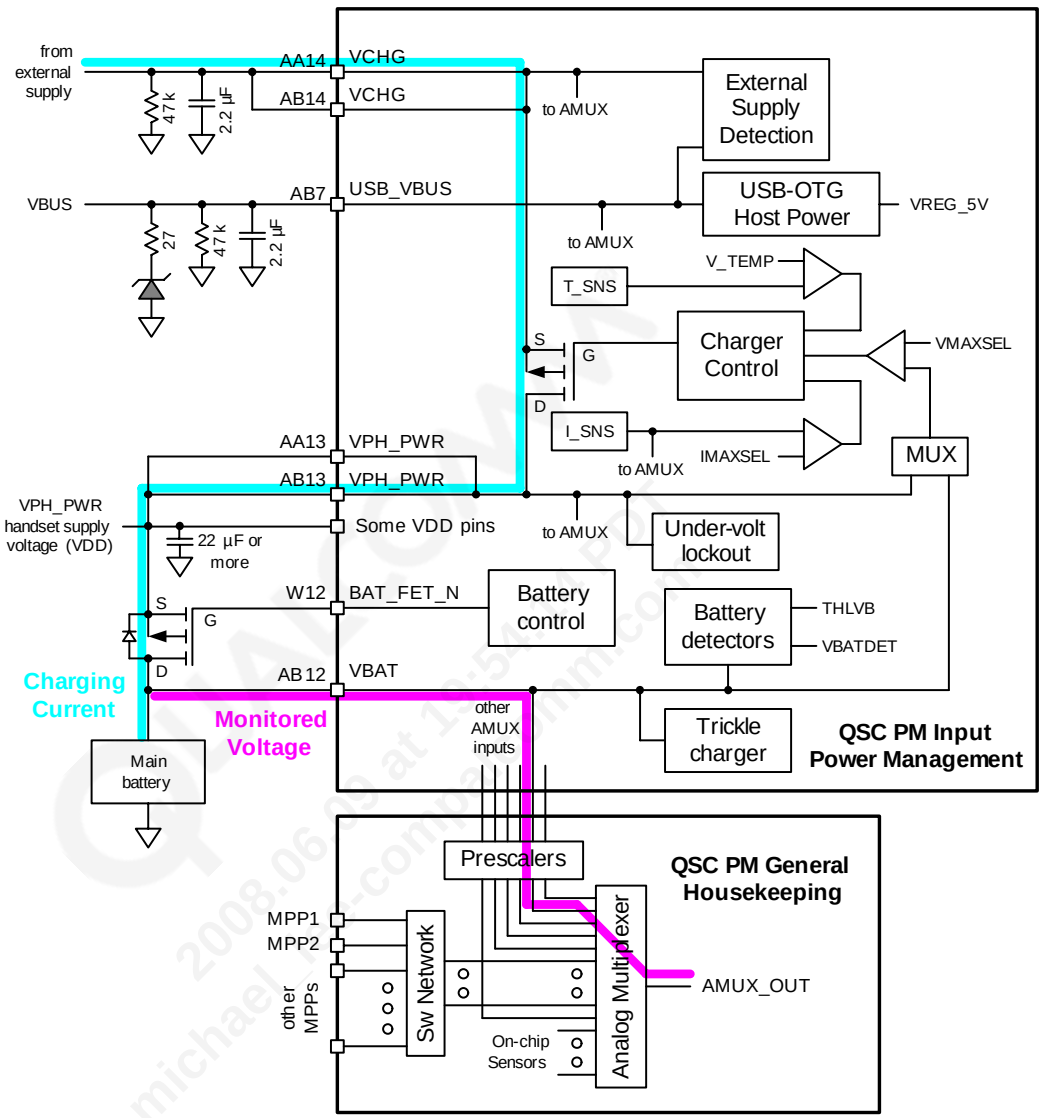


Figure 13-8 Constant current charging

During constant current charging the battery is charged with a constant current level that is set by the current regulation loop (see Section 13.10 for details) or the current limited external supply. As the battery voltage rises and approaches its desired value, the charging current begins to decrease. This is the end of constant current charging and the beginning of residual charging (Figure 13-9). The target value, VMAXSEL, is set by software to a value higher than the desired final voltage (VBATDET) to overcome the battery's internal ESR and achieve faster charging.

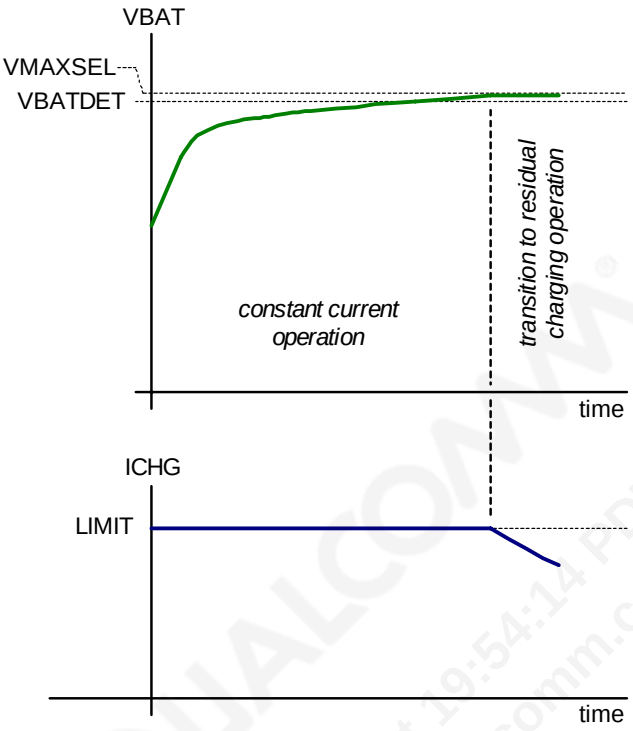


Figure 13-9 V, I curves near the end of constant current charging

The charging battery voltage is monitored by an on-chip, dedicated voltage detection circuit or by the housekeeping ADC via the PM analog multiplexers. The HKADC path is highlighted in Figure 13-8. If the battery detector method is used, its real-time status is reported automatically, and the interrupt manager could also be used. If the HKADC is used, the proper analog multiplexer input is selected and its output signal delivers the battery voltage to the HKADC.

Whichever technique is used to detect the battery reaching its threshold, software monitors the voltage and takes the appropriate action to terminate the constant current charging mode. Charging continues with residual charging (either constant voltage or pulse).

13.12.3 Constant voltage charging

Once constant current charging of a Li-ion battery is finished, the charging continues using either constant voltage or pulse techniques. Discussion of constant voltage charging is presented here; pulse charging is covered in the next subsection.

The QSC62x0 device and its software determines if and when it is appropriate to begin the constant voltage mode within the charging process. Usually the decision to stop the constant current mode is based on the battery voltage reaching a programmed threshold slightly above the intended final voltage.

QSC support of main battery constant voltage charging is very similar to its constant current mode: the battery MOSFET is closed and the charging pass transistor is closed-loop controlled. But in this case, the closed-loop control regulates the voltage at VBAT (pin AB12) to match the programmed value (VMAXSEL). This ensures the most accurate final battery voltage — Li-ion battery manufacturers recommend a voltage accuracy of 1% or better at the end of charge. The battery voltage is constant (or nearly so) while the charging current decreases exponentially for the remaining charge process (Figure 13-10).

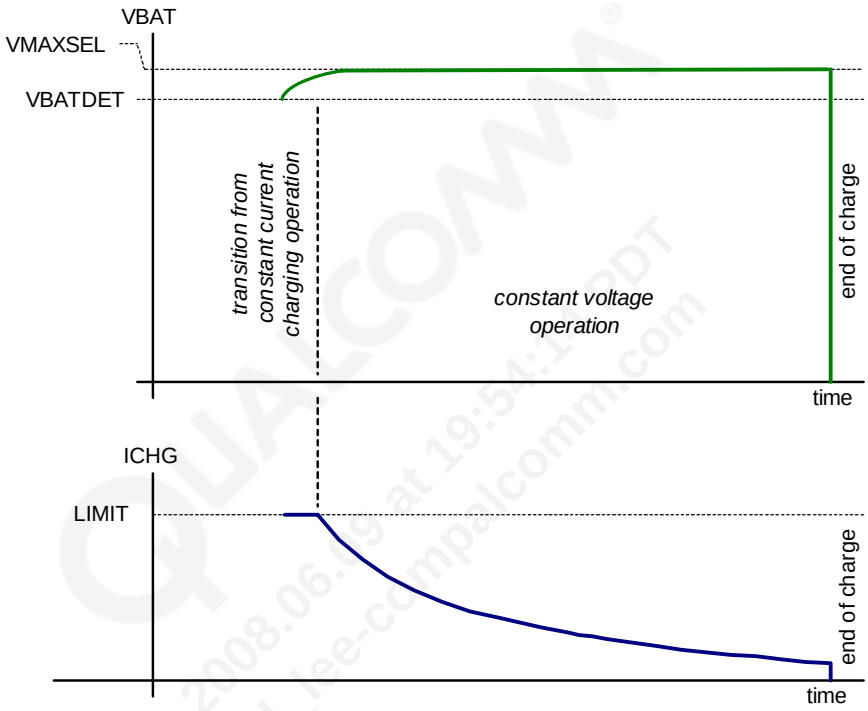


Figure 13-10 V, I curves for constant voltage charging

To improve the charged battery's final voltage accuracy, V_{BAT} is regulated rather than V_{DD} thereby eliminating the voltage drop across the battery transistor. The feedback input is V_{BAT}, selected automatically via the on-chip analog multiplexer (Figure 13-11). VMAXSEL must be programmed to the desired battery voltage.

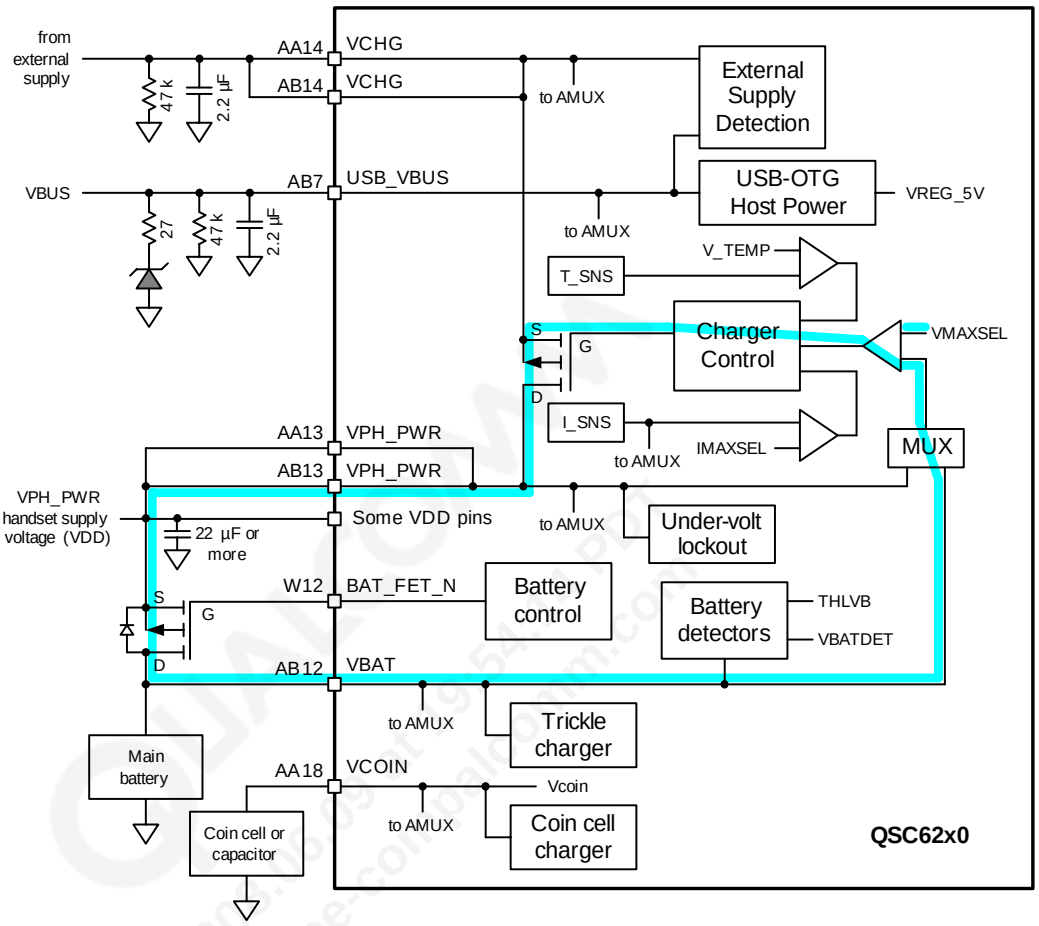


Figure 13-11 V_{BAT} regulation during constant voltage charging

The end of constant voltage charging is typically detected in one of two ways:

1. Monitor the charging current using the HKADC and terminate charging when it decreases to a predetermined threshold (such as 10% of the full charging current).
2. Allow constant voltage operation for a predetermined duration beyond crossing the VBATDET threshold (on the order of one-half to two hours).

In either case, do not allow charging to continue indefinitely — charging too long will damage the battery. Consult battery manufacturers for specific recommendations.

If option 1 is implemented, the charging current is measured during the constant-current charging mode, then is monitored regularly during constant voltage charging until the terminating condition is realized. The HKADC takes two readings for each charging current measurement:

1. With the battery MOSFET closed (handset plus charging current)
2. With the battery MOSFET opened (handset current only)

The difference between these two current measurements is the charging current. Regardless of the technique used to detect the end of charge, software must take the appropriate action to terminate the charging operation.

13.12.4 Pulse charging

After constant current charging of a Li-ion battery is finished, the charging continues using either constant voltage or pulse techniques. Pulse charging is discussed in this subsection; constant voltage charging was covered in the previous subsection.

NOTE Pulse charging is only available when using the software-controlled charging method. It is not available when using the autonomous charging method.

There are two options for implementing pulse charging:

1. By switching the battery MOSFET on and off until the battery's open circuit voltage stays above the VMAXSEL setting.
2. By switching the charging pass transistor on and off. This option is the preferred method for all designs and makes pulse charging possible even if the battery MOSFET is omitted. In this case, phone electronics should draw minimal current so that the battery's open circuit voltage can be measured accurately during the off interval.

The pulse-charging algorithm is illustrated in Figure 13-12. This illustration assumes charging pass transistor switching with the battery transistor closed or omitted.

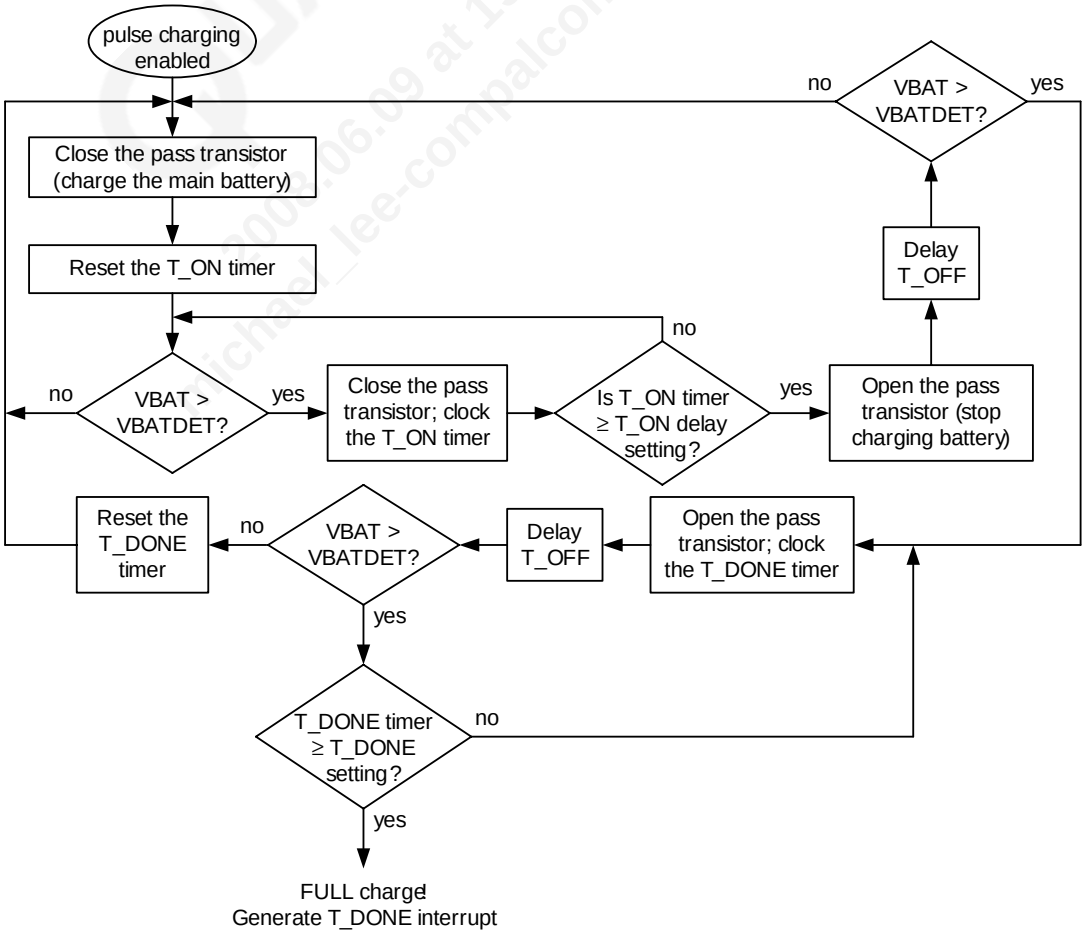


Figure 13-12 Pulse charging algorithm

Pulse charging, compared to constant voltage, provides better battery voltage accuracy, reaches full charge more quickly, and dissipates less transistor power when switching from constant current charging. Pulse charging is enabled through software and uses the same hardware as constant current or constant voltage charging, but repetitiously opens and closes the charging pass transistor (or battery transistor) to deliver current pulses to the battery.

First, consider option 1 for pulsed charging — switching the battery transistor. Figure 13-12 still applies, but the battery transistor is switched on and off rather than the charging pass transistor. Charging begins with the battery transistor closed, the battery charging, and the QSC device comparing the battery voltage (V_{BAT}) against the programmed threshold VBATDET. The charging pass transistor is closed-loop controlled to regulate V_{DD} to VMAXSEL. The VMAXSEL value is programmed higher than the desired final battery voltage (VBATDET) to yield higher charging current for the battery and to overcome its internal ESR (as the lithium battery approaches full charge, its ESR increases).

As the battery charges, it reaches the VBATDET threshold, causing the internal counter T_ON to start. The battery transistor stays closed (and the battery continues charging) until the T_ON counter expires. Then the battery transistor is opened, charging stops, and another internal counter (T_OFF) is enabled. Without continued charging, the battery voltage may drop — if it drops too far, additional charging is needed. If it holds its voltage, it is fully charged and the charging process is terminated. This is one purpose of pulsed operation — to check and recheck the battery's open circuit voltage, confirming a full charge before terminating the process.

The battery voltage is allowed to drop throughout the T_OFF interval. When T_OFF times out, the battery voltage is again compared to VBATDET, and:

If V_{BAT} < VBATDET	Battery charging is resumed until V_{BAT} > VBATDET, then the T_ON counter is restarted. The T_ON and T_OFF cycles are continued and the battery voltage is monitored as described above until V_{BAT} > VBATDET at the end of a T_OFF cycle.
If V_{BAT} > VBATDET	The battery transistor is kept open and a third internal counter (T_DONE) is incremented. The T_OFF counter recycles and continues counting as long as V_{BAT} > VBATDET at the end of each off cycle.

At the end of each T_OFF cycle V_{BAT} is checked again, leaving the transistor open:

If V_{BAT} < VBATDET	The battery voltage has dropped below the threshold, so charging is resumed. The T_DONE counter is cleared and the battery transistor is closed, resuming battery charging until V_{BAT} > VBATDET, then the T_ON counter is started and the entire pulse charging sequence is restarted.
If V_{BAT} > VBATDET	The T_DONE counter is checked. As long as V_{BAT} stays above VBATDET, the T_DONE continues counting T_OFF cycles until it expires. This signals that the battery is fully charged and an interrupt is sent to the baseband circuits. A SW override is used to force the battery transistor to stay open, ensuring the charging process is terminated. If T_DONE has not expired when checked, the final loop of the algorithm is continued until it does expire (as long as V_{BAT} > VBATDET).

When pulse charging is finished (T_DONE reaches terminal count and the interrupt is generated), the battery is fully charged and has proven so by holding that full charge for an extended period.

The charging algorithm is the same for option 2, only the charging pass transistor is switched on and off rather than the battery transistor. This is the preferred option, especially if weak external chargers might be used.

Timing intervals for pulsed current charging are programmable:

- T_ON is the on time (the current pulse duration) — the charging current source is connected to the battery. Valid settings are 62.5, 125, 250, and 500 msec.
- T_OFF is the off time – the charging current source is disconnected from the battery. Valid settings are 62.5, 125, 250, and 500 msec.
- T_DONE is the number of consecutive T_OFF intervals required before the QSC device will end the charging — the battery is fully charged, so charging is done. The PMIC reports that charging is finished using an interrupt. T_DONE values represent multiples of the programmed T_OFF value; valid settings are 16, 32, 64, and 128.

Sample voltage and current waveforms are shown in Figure 13-13. VBATDET is programmed via software.

Each time the charging pass transistor (or battery transistor) is closed, the battery voltage jumps up, then continues rising in an exponential fashion. Opening the transistor causes an immediate drop, followed by an exponential decay. Early in pulse charging operation, the battery voltage drops below the VBATDET threshold before T_OFF times out (as shown in detail A of Figure 13-13). Possibly thousands of pulses later, the battery voltage stays above VBATDET for multiple T_OFF intervals, but still drops below that threshold before T_DONE expires (see detail B of Figure 13-13).

The battery continues charging with each pulse of current, though, and eventually T_DONE expires before the battery voltage drops below VBATDET. This indicates the battery is fully charged and end-of-charge is signaled (detail C of Figure 13-13). The sample waveform shows very few charge pulses where thousands are actually required — the diagrams in Figure 13-13 are for illustrative purpose only. For example, detail C

illustrates a case where T_DONE = 4 x T_OFF, but the minimum programmable value for T_DONE is actually 16 x T_OFF.

Pulse charging uses the same hardware as the other techniques, but requires proper software control. On-chip counters are programmed to define the pulse timing (T_ON, T_OFF, and T_DONE). These circuits count the on-chip RC oscillator or off-chip crystal oscillator cycles — the crystal is used except at initial startup. These circuits count the internal time base and the time values of these counters are based on the nominal crystal frequency of 32.768 kHz.

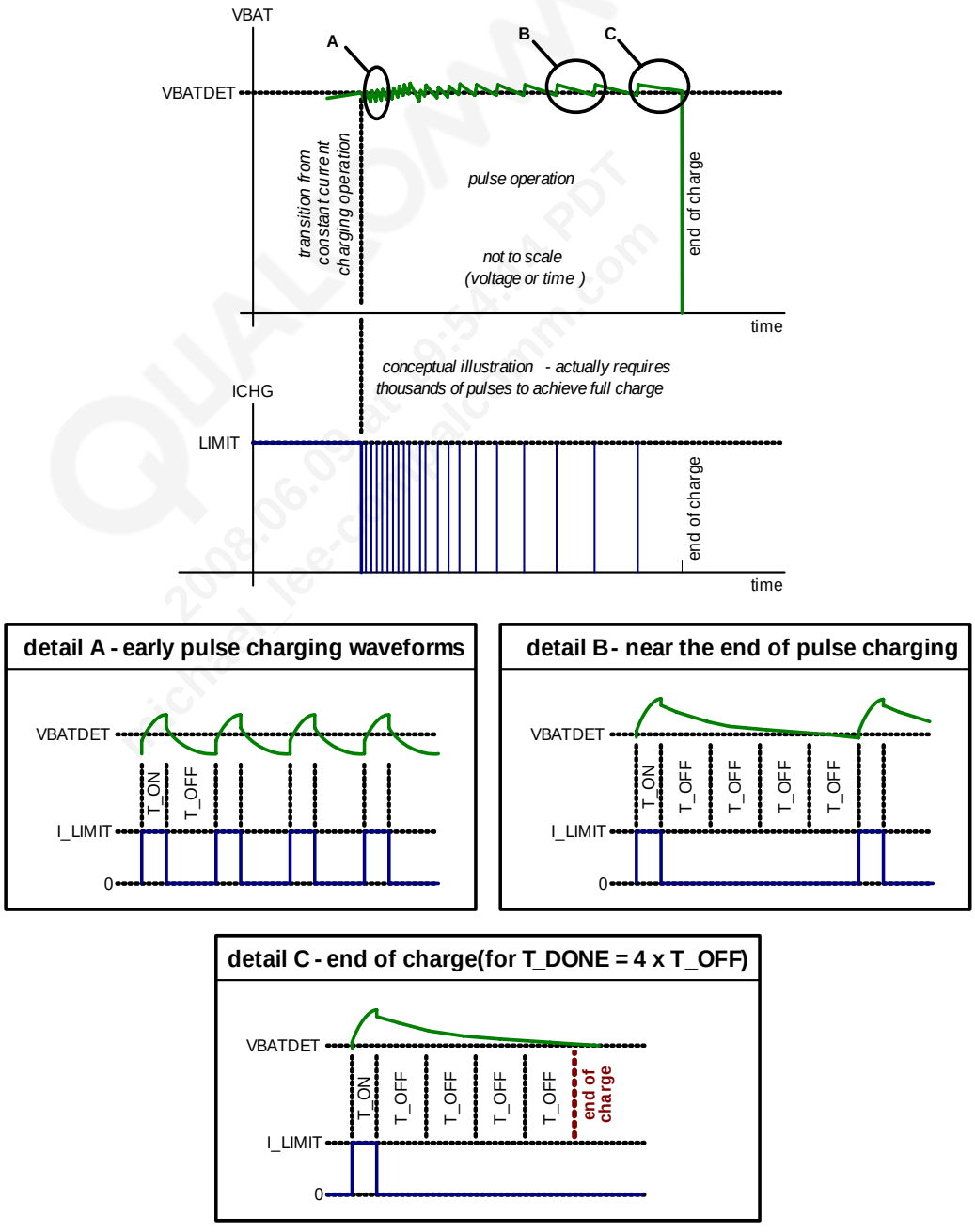


Figure 13-13 V, I curves for pulse charging

13.12.5 Autonomous charging

The QSC62x0 device includes a new feature called autonomous charging that lets the power management circuits conduct battery charging without intervention from the handset software. The autonomous charging algorithm is implemented in a state machine; once it begins, it executes a sequence of predetermined states until charging is completed.

13.12.5.1 Autonomous-charging state machine

The state machine is started during powerup, and coordinates the establishment of the handset supply voltage (V_{DD}) during the earliest stages of the power-on sequence. In the course of establishing V_{DD}, the state machine recognizes and responds to several conditions — which supplies are available, whether an intelligent USB charger is attached or not, whether the battery requires trickle charging, etc. After executing the proper sequence, the startup is complete and a stable V_{DD} supply is established.

After a successful startup, the state machine recognizes if the battery needs to be charged and takes the appropriate steps. It conducts battery charging until its end-of-charge requirements are met, and then it stops charging and allows the external supply to power the handset until the freshly charged battery is needed.

Other features of the autonomous charging state machine include:

- Trickle charging algorithm (not supported in the first engineering sample)
- Fast charging algorithm

  NOTE Pulse charging is not available during state machine operation; it is only available when the handset is using its software-controlled charging mode.

- Support for switching from one power source to the other (as needed) to ensure a continuous supply to the handset system
- Guaranteed USB compliance upon power-on
  - When an intelligent USB device is connected the state machine will power up from the battery (if the battery is sufficiently charged).
  - If the battery is severely depleted, the state machine revives the battery and then continues the powerup sequence. This procedure is called auto trickle charging (not supported in the first engineering sample).
- Support for determining the end-of-charge condition based upon measured parameters
- Support for partially disabling the state machine's functions by means of software

13.12.5.2 State machine operation

At startup, the power-on sequencer turns over control to the charger state machine in an effort to establish V_{DD} (VPH_PWR). The state machine reads some initial conditions and then selects the most appropriate of three courses of action:

- Close the battery MOSFET to allow the battery to power the VDD node.
- Turn on the charging pass transistor and allow the input power management circuits to regulate the V_{DD} voltage using the external charger as the power source.
- Initiate auto-trickle charging to precharge the battery. Once the battery is sufficiently charged, the battery MOSFET is closed and the charger regulator is turned on with a 90 mA current limit.

The most appropriate course of action is decided using the following criteria:

- Battery power (first) is selected if:
  - A charger is not present, or
  - An invalid charger is present (V_{CHG} is below 3.3 V or above 7 V), or
  - A valid charger is present but it is an intelligent USB device and the battery is not severely depleted
- External charger power (second) is selected if:
  - A regular charger (not an intelligent USB device) is present and valid or
  - An intelligent USB charger is present and valid, but the battery is severely depleted and the LOW_CURR_BOOT register bit is set to 1 (meaning that the QSC device can perform a low current boot)
- Auto-trickle charging (third) is selected if:
  - An intelligent USB charging device is present (no more than 100 mA draw is allowed until negotiations are complete, otherwise a regular start-up sequence might not be sustained) and
  - The battery is severely depleted (cannot support startup, as determined by a programmable voltage threshold) and
  - The LOW_CURR_BOOT register bit is set to 0 (meaning that the QSC device cannot perform a low current startup)

The third course, auto-trickle charging, is explained in the next section.

13.12.5.3 Auto trickle charging

NOTE Auto trickle charging is not supported in the first engineering sample.

As suggested in the previous section, auto trickle-charging can only be used if an intelligent USB device is present. If this condition and the other two conditions are met, auto trickle charging is conducted in two phases:

- Phase 1: the battery is charged with a programmed constant current (such as 80 mA) until the battery voltage rises above a programmed threshold (such as 3.2 V).
- Phase 2: once the threshold is matched, the battery continues charging with the same current for a fixed time (also programmable).

At the conclusion of auto trickle charging, the state machine continues to detect that an intelligent USB device is attached and that the battery is revived and can continue the startup sequence. So the algorithm continues:

- The battery MOSFET is closed (connecting the battery to the VDD node).
- To supplement the battery during the startup process, the charger regulator is also turned on.
- The charger regulator current is limited to about 90 mA (ensuring that the USB specified maximum current of 100 mA is not exceeded).
- The auto-trickle charger signals the power-on sequencer that it can proceed.

13.12.5.4 Conclusion of a successful startup

Successful completion of startup proceeds differently depending upon which one of the three courses of action was selected (from Section 13.12.5.2):

1. If battery power was selected: the battery was deemed capable of maintaining a stable V_{DD} throughout the power-on sequence.
2. If the external supply was selected: the external supply was deemed capable of maintaining a stable VDD throughout the power-on sequence.
3. If auto-trickle charging was selected and executed: the battery (possibly supplemented by the external supply) is now capable of maintaining a stable V_{DD} throughout the power-on sequence. An intelligent USB device is connected, requiring a few more state machine steps:
   - Upon completion of the startup sequence, software negotiates (USB enumeration) with the USB device for a higher current and then sets the ENUM_DONE register bit to 1.
   - When ENUM_DONE goes high, the charger state machine sets the charger regulator current limit to its maximum value (IMAXSEL).
     - If autocharging (Section 13.12.5.5) is enabled, the battery MOSFET remains closed.
     - If autocharging is not enabled, the battery MOSFET is opened.

As seen above, a successful startup ends with a stable power source for the VDD node (regardless of the path taken).

After a successful startup, the power-on sequencer communicates with the autonomous charging state machine to maintain the V_{DD} voltage (from either the external supply or the battery), or to disconnect the VDD node from both power sources (to turn off the system). If the power-on sequencer holds V_{DD} on, the state machine determines whether to initiate autocharging or not. Auto-charging is discussed in the next section.

If the power-on sequencer is holding V_{DD} on via the battery, the state machine will switch over to a valid charger only if:

- The charger is not an intelligent USB device, or
- The charger is intelligent and the USB enumeration has already happened

13.12.5.5 Autocharging

After a successful startup, while V_{DD} is being held on, the state machine will automatically begin charging the battery if:

- Autocharging is enabled by software through its register bit, and
- The battery temperature is within range, and
- The charger is not an intelligent USB device or USB enumeration has happened

If all three of these conditions are met, autocharging is started. But the method of charging depends upon the battery voltage:

- If V_{BAT} is between 3 V and 4.1 V --> fast charging is enabled.
- If V_{BAT} is below 3 V --> trickle charging is enabled, eventually followed by fast charging.

13.12.5.6 End of charge

The charging process is terminated when:

- The maximum charging time has expired or
- The termination current has been reached and the charging pass transistor temperature is within its tolerable range

All the key voltage, current, and timing thresholds involved in determining the end of charge are software programmable.

If current flow is the variable used to terminate charging, be aware that the total current (phone plus battery) is measured rather than just the battery current. If the phone is being used during autocharging, one of the following situations may occur:

- The maximum charging timer expires and the charging process is terminated before the battery is fully charged. This is not critical since a new charging cycle is restarted immediately by the state machine and an interrupt is sent to inform software.
- The battery reaches a fully charged condition but the state machine remains in the charging state. Again, this is not critical because the end of charge is just postponed until the user stops using the phone (the battery MOSFET remains closed after the termination current is reached, but this should not cause any harm to the battery).

13.12.5.7 Autonomous charging flow diagram

A high-level flow diagram for autonomous charging is shown in Figure 13-14.

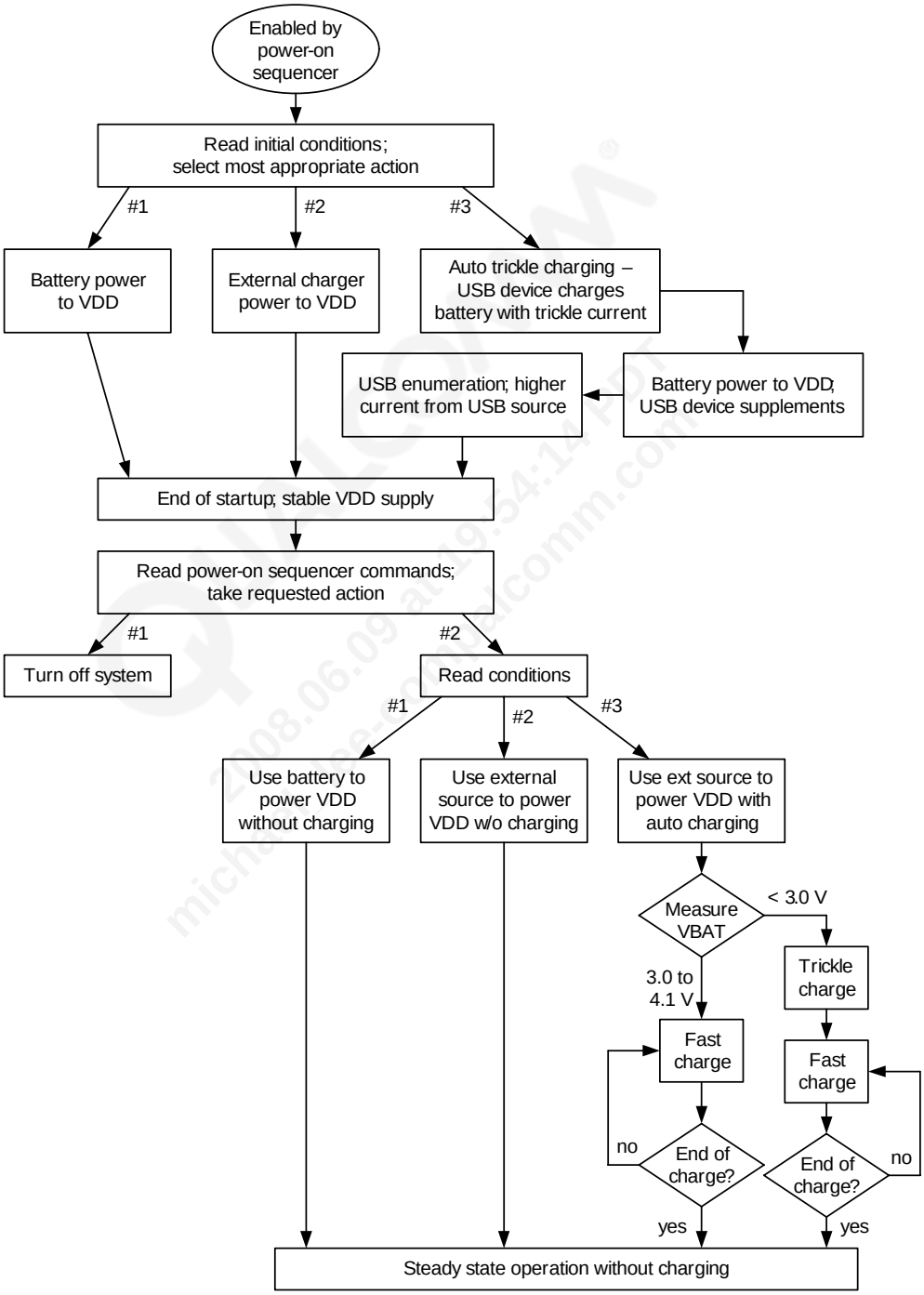


Figure 13-14 Autonomous charging flow diagram

13.13 Coin-cell charging and keep-alive functions

Coin-cell charging is enabled through software control; its supply current is sourced from the main battery through the VBAT pin (Figure 13-15). The on-chip coin-cell charger is implemented using a voltage regulator and series resistor, both programmable. The regulator output can be set to 3.0, 3.1, or 3.2 V and the resistor value can be set to 800, 1200, 1700, or 2100 Ω. All coin-cell charging activity is controlled through software, allowing continuous or periodic charging of the coin cell and management of the main battery current. The HKADC reads the coin-cell voltage through the analog multiplexer to monitor charging. The charging current path and voltage monitoring path are both highlighted in the figure.

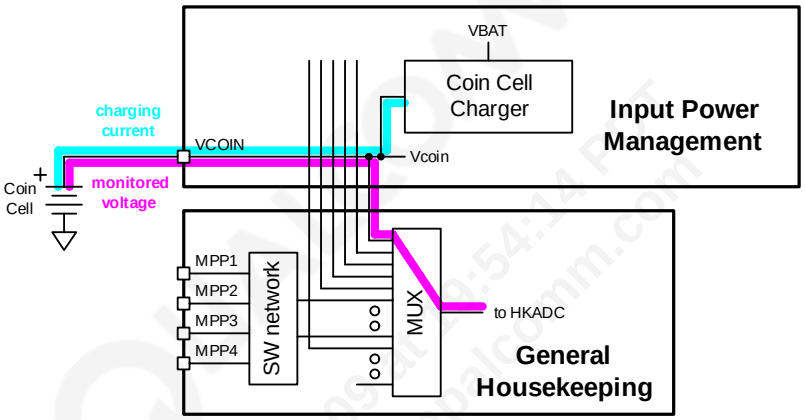


Figure 13-15 Coin-cell charging

The coin-cell charger's programmed settings remain valid even if the main battery is removed. If the coin-cell charger is enabled when the main battery is removed, charging will resume once the battery is reconnected. The charging resumes even if the phone is in an off state. The settings are reset to their default values only if the coin cell itself is removed. The coin-cell charger settings are not reset if an SMPL event occurs.

When the phone is off, the 32.768 kHz crystal oscillator and real-time clock circuits continue to run. These circuits are powered by the main battery, if present and valid. If not, these circuits are powered by the lithium manganese dioxide rechargeable coin cell that is connected to VCOIN (pin AA18).

A valid voltage on VCOIN is required to run the SMPL timer, but the coin cell may be replaced with an appropriate capacitor to maintain the SMPL feature only (see Section 13.16 for details).

To maximize the shelf life of a factory-installed coin cell, software can disable all coin-cell-related circuitry. When the new handset is originally initialized, software can enable coin-cell circuits and allow normal operation from that time forward.

13.14 Battery alarms

A programmable window detector continuously monitors the battery voltage at pin AB12 (VBAT). Both the upper and lower thresholds are programmable and include voltage hysteresis to ensure stability. The valid voltage settings are 2.8 to 4.4 V in 100-mV increments; the hysteresis is about 20 mV. To prevent brief voltage transients from interrupting processing unnecessarily, the out-of-range condition must stay triggered for a certain amount of time before an interrupt is generated. This delay, referred to as time hysteresis, is also programmable (values range from 0.122 to 15.63 msec). If the battery voltage goes back in-range before the programmed delay, the delay timer is reset and no interrupt is generated.

Whenever the battery voltage (V_{BAT}) drops below the lower threshold a low-battery alarm is generated (Figure 13-16). If it stays below the threshold longer than the programmable time hysteresis interval, a low-battery alarm is sent to the interrupt manager.

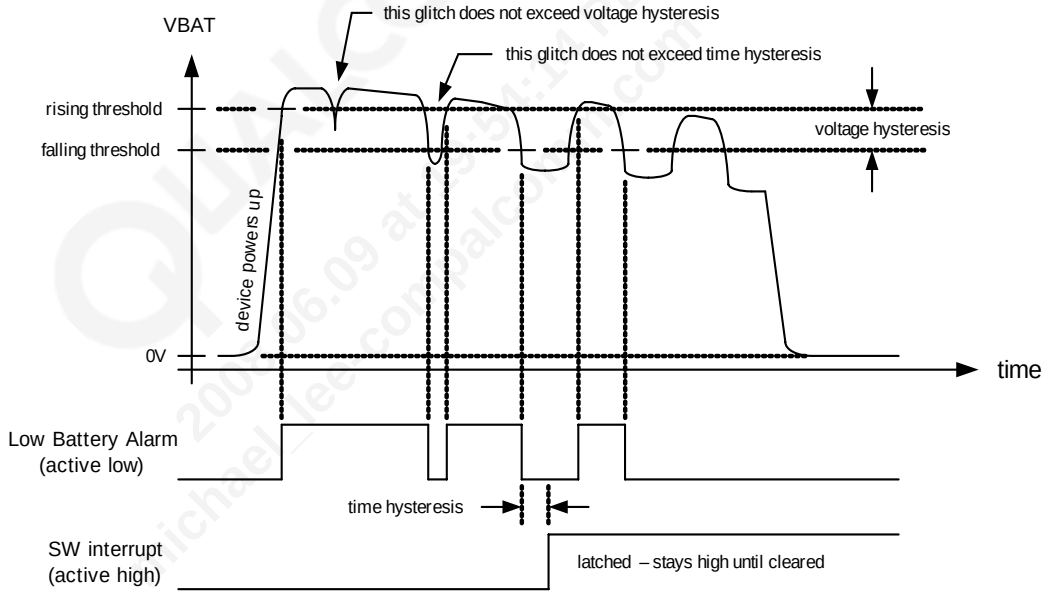


Figure 13-16 Low-battery detection

13.15 Undervoltage lockout

The handset supply voltage (V_{DD}) is monitored continuously by an undervoltage lockout (UVLO) circuit that automatically turns off the device at severely low V_{DD} conditions. Of course, the UVLO threshold (2.55 V nominal) is lower than the low-battery alarm discussed in the previous section.

Although UVLO is a hardware feature, it allows software interaction to realize additional features such as:

- Sudden momentary power loss (SMPL) recovery – Section 13.16
- Watchdog timeout soft reset – Section 16.3.5
- Power-on sequence abort – Section 16.3.1

QSC62x0 undervoltage lockout circuits monitor the power supply voltage at pins AA13 and AB13 (VPH_PWR — connected to the primary handset voltage V_{DD}) to assure minimum requirements are met for proper IC operation. As the device powers up, V_{DD} must exceed a rising threshold to initiate the power-on sequence. Voltage hysteresis and delays prevent minor glitches from being detected as UVLO events (Figure 13-17). If V_{DD} drops below the falling threshold for sufficient duration, a valid UVLO event is detected. The PON_RESET_N signal is cleared (LOW) and the device is powered down.

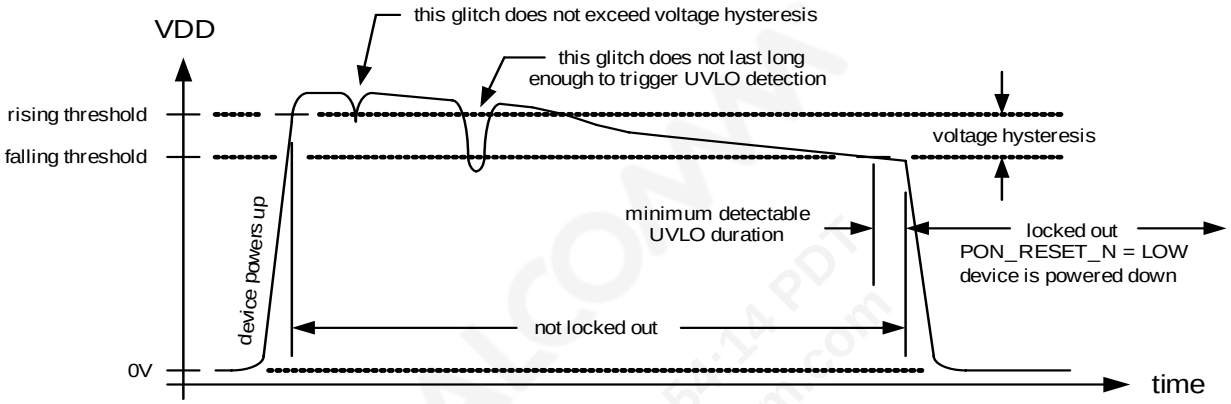


Figure 13-17 UVLO detection

Neither the baseband circuits nor software is involved in real-time UVLO detection. The voltage thresholds and time delays are programmable, but UVLO events do not generate interrupts. They are reported to the baseband circuits via the PON_RESET_N signal as part of the powerdown sequence.

13.16 Sudden momentary power loss

The QSC62x0 SMPL feature initiates a power-on sequence without software intervention if the monitored phone voltage (V_{DD}) drops out-of-range (< 2.55 V nominal) then returns in-range within a programmable interval of between 0.5 and 2.0 seconds (Figure 13-18). When enabled by software, SMPL achieves immediate and automatic recovery from momentary power loss (such as a brief battery disconnect when the phone is jarred).

NOTE An external 32.768 kHz sleep crystal is required for the SMPL feature.

When SMPL is enabled, a UVLO event clears PON_RESET_N and the QSC device is powered down. The coin-cell battery or capacitor takes over as the power source for the crystal oscillator, RTC, and SMPL circuits. The SMPL counter begins counting oscillator pulses on UVLO detection. If V_{DD} returns to its valid range before the SMPL counter expires, the PM circuits initiate a power-on sequence without software intervention. An interrupt is sent to the baseband circuits indicating that power was momentarily lost, the RTC may be corrupted due to insufficient voltage, and that the current PM actions are not a normal power-on sequence.

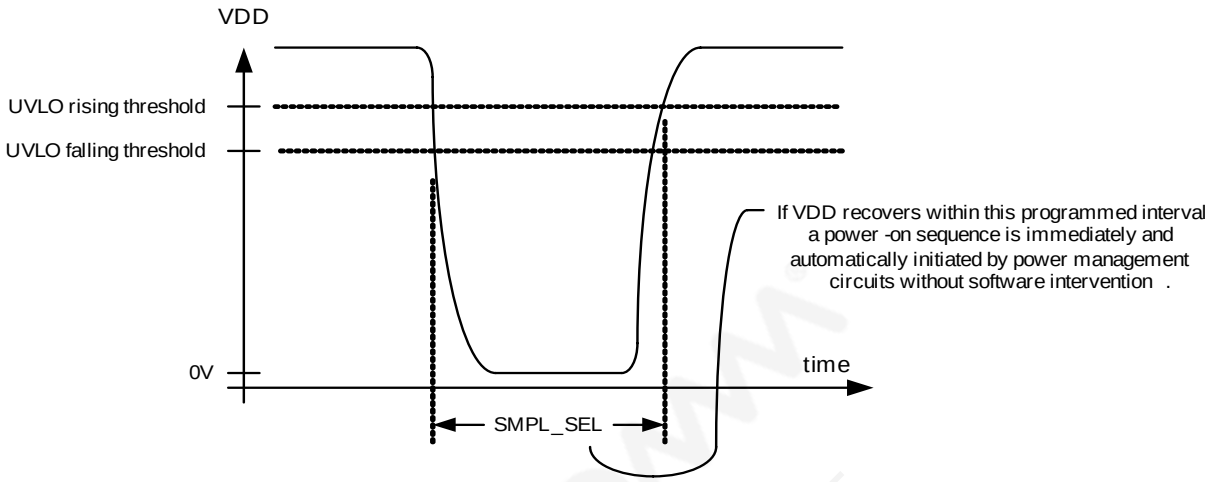


Figure 13-18 SMPL

If the SMPL counter expires without V_{DD} returning to its valid range the handset must undergo the normal power-on sequence whenever the next initializing event occurs (keypad controlled, supply detected, etc.).

SMPL operation requires that a coin cell or capacitor be installed and that the SMPL function be enabled by software. If a capacitor is used instead of a coin cell, it must be connected between VCOIN and ground. The capacitor must be charged (using the on-chip coin-cell charger as needed) to operate properly as the SMPL power source. The capacitor value depends on the SMPL timer setting (Table 13-2).

Table 13-2 Keep-alive capacitor values vs. SMPL timer settings

SMPL timer setting	Capacitor value	Capacitor package (X5R)
0.5 sec	1.5 µF	0805
1.0 sec	3.3 µF	0805
1.5 sec	4.7 µF	0805
2.0 sec	6.8 µF	1206

If SMPL is used and a normal powerdown is desired, the SMPL function must be disabled via software before deasserting PS_HOLD to avoid an inadvertent SMPL override. Without disabling SMPL, the PM circuits will detect and defeat the intended powerdown, instead executing a soft reset. See Section 16.3.5 for a discussion of the watchdog timer and soft reset functions. The differentiating factor between SMPL and watchdog timeout is the UVLO detector state. If a UVLO did not occur (V_{DD} stayed above the UVLO threshold), the powerup is a watchdog reset. If a UVLO did occur, the powerup is an SMPL event. Either event triggers its respective interrupt, informing software what caused the powerup.

SMPL circuits run off an internal voltage net (xV_{DD}) that is formed by diode-ORing voltages at the VBAT, VPH_PWR (V_{DD}), and VCOIN pins; this voltage normally runs the crystal oscillator and RTC as well. If this voltage drops below the lower xV_{DD} threshold an interrupt is generated that indicates the RTC may be corrupted due to insufficient voltage.

13.17 V_{DD} collapse-protection circuit

Some handset manufacturers may specify a low-current charger that cannot handle the peak phone plus charging current. To prevent a sudden load from inadvertently collapsing the V_{DD} voltage when a low-current charger is used, the QSC62x0 device monitors the voltage across the battery MOSFET (through the VPH_PWR and VBAT pins) and automatically turns it on if V_{DD} drops about 10 mV below VBAT (Figure 13-19).

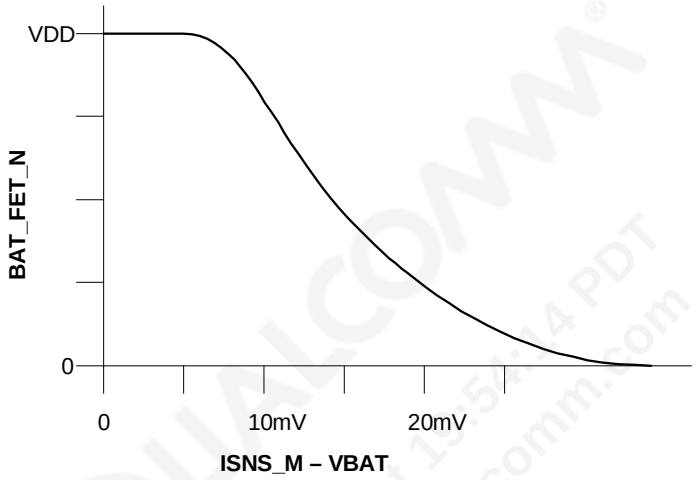


Figure 13-19 Hypothetical V_{DD} collapse response curve

Another way to prevent high transient loads from collapsing V_{DD} when using a low-current charger requires setting the CHG_CHARGE_BAT bit (always enabling the fast charge mode). In this case, charging is terminated by setting the CHG_CHARGE_DIS bit to open the charging pass transistor; this keeps the battery connected to V_{DD} at all times.

14 Output Voltage Regulation

The QSC62x0 device includes all the regulated voltages needed for most low-cost wireless handset applications (and many other applications). Independent regulated power sources are required for various electronic functions to avoid signal corruption between diverse circuits, support power-management sequencing, and meet different voltage-level requirements. Sixteen voltage regulators are provided — all programmable, all derived from a common bandgap reference circuit.

Three major types of voltage regulator circuits are on-chip:

- Three positive voltage switched-mode power supply (SMPS) circuits
  - One boost converter (rated for 600 mA)
  - Three buck converters (rated for 500 mA each)
- Thirteen positive voltage linear regulators
  - Four rated for 300 mA
  - Six rated for 150 mA
  - Three rated for 50 mA
- One negative voltage charge pump rated for 200 mA, referred to as a negative charge pump (NCP)

Each regulator has two logic-OR input bits; a logic high at either input enables that regulator:

- A master bit that enables all regulators according to their default condition
- A dedicated bit that enables only that regulator

The master enable reduces the number of write cycles needed when switching between the phone's sleep and active modes.

Additional comments that apply to regulator functions:

- If a regulator's default condition is on, that regulator will power on automatically at QSC startup.
- The MSMP regulator must be on to allow internal communications between major functional blocks.
- Each regulator and SMPS can provide more than its rated output current, though some performance characteristics might be degraded.
- All regulated output voltages are programmable.

- All regulators can be set to a low-power mode except the VREG_USB_3P3 and VREG_NCP circuits.
- Higher efficiency is achieved by using subregulation, a technique that is highly recommended to generate some QSC circuit supply voltages. See Section 14.6 for further discussion.

Table 14-1 provides a high-level summary of the on-chip regulators.

Table 14-1 Voltage regulator summary

Type/name [1] [2]	Default conditions [3]	Voltage range	Intended use
SMPS - boost (600 mA)	Off, 5.000 V	3.000 to 6.100 V	USB-OTG, camera flash, high-power audio
SMPS - buck			
MSMC (500 mA)	On, 1.225 V	0.750 to 3.050 V	Processor core
RF1 (500 mA)	On, 2.300 V	0.750 to 3.050 V	First stage regulator for subregulation set 1
RF2 (500 mA)	On, 1.400 V	0.750 to 3.050 V	First stage regulator for subregulation set 2
Linear - 300 mA			
MSME	On, 1.800 V	1.500 to 3.050 V	EBI circuits and external memory, pads
MSMP [4]	On, 2.600 V	1.500 to 3.050 V	Pad voltage for digital I/Os
RFA	Off, 2.200 V	1.500 to 3.050 V	RF and analog circuits
GP2	Off, 2.900 V	1.500 to 3.050 V	External WLAN circuits
Linear - 150 mA			
RFRX2	Off, 1.300 V	0.750 to 1.525 V	RF receiver circuits
RFTX2	Off, 1.300 V	0.750 to 1.525 V	RF transmitter circuits
CDC2	Off, 1.300 V	0.750 to 1.525 V	Audio codec circuits
MPLL	On, 1.300 V	0.750 to 1.525 V	Digital PLL circuits
USIM	Off, 1.800 V	1.500 to 3.050 V	USIM circuits and interface
GP1	Off, 2.850 V	1.500 to 3.050 V	External Bluetooth circuits
Linear - 50 mA			
TCXO	On, 2.850 V	1.500 to 3.050 V	TCXO and XO circuits
USB_2P6	Off, 2.600 V	1.500 to 3.050 V	USB circuits and interface at 2.6 V
USB_3P3	Off, 3.300 V	3.000 to 6.100 V	USB circuits and interface at 3.3 V
SMPS - NCP (200 mA)	Off, -1.800 V	N/A	Headphone negative voltage supply

1 All regulator names are based on their intended use, though some may be used to power alternate functions. For example, the USIM regulator is intended to power an external SIM card but may be used to power other circuits (or not used at all).

2 Each current listed in this table is its regulator's rated value – the current at which the regulator meets all its performance specifications. Higher currents are allowed, but higher input voltages may be required and some performance characteristics may become degraded. See the appropriate regulator sections for details.

3 All regulators have default output voltage settings, even if they default to an off condition.

4 VREG_MSMP powers key internal circuits and should be kept on at its default voltage setting.

14.1 Regulator connections

Regulator connections are listed and described in Table 14-2. Interface circuit diagrams are included for each regulator type in its section within this chapter. Specific recommendations for all regulator external components are given in the next section.

Table 14-2 Output voltage regulator connections

Pin name/function	Pin #	Pin type [1]	Functional description
Positive SMPS connections			
VSW_5V	AC12	AO	The switching output of the +5-V boost (step-up) SMPS circuit; connect to the primary phone power through a 2.2-µH inductor, and also connect to the Schottky diode's anode pin.
VREG_5V	AC11	AI	This senses the regulated output of the +5-V boost SMPS; connect to the Schottky diode's cathode pin. Bypass this pin with a 10-µF ceramic capacitor and route to the desired load circuits.
VSW_MSMC [2]	AC18	AO	The switching output of the core SMPS circuit; connect to the core regulator's 2.2-µH buck inductor, input side.
VSW_RF1	AC15	AO	The switching output of the RF1 SMPS circuit; connect to the RF1 regulator's 2.2-µH buck inductor, input side.
VREG_RF1	AB15	AI	The regulated output of the RF1 SMPS circuit, buck inductor output side; voltage is sensed by this pin. Bypass with a 10-µF ceramic capacitor and connect it to off-chip RF circuits and/or QSC linear regulator input pins as desired.
VSW_RF2	AC17	AO	The switching output of the RF2 SMPS circuit; connect to the RF2 regulator's 2.2-µH buck inductor, input side.
VREG_RF2	AB17	AI	The regulated output of the RF2 SMPS circuit, buck inductor output side; voltage is sensed by this pin. Bypass with a 10-µF ceramic capacitor and connect it to off-chip RF circuits and/or QSC linear regulator input pins as desired.
Linear regulator outputs			
VREG_CDC2	M18	AO	Linear regulator output that provides the low supply option for the internal codec circuits; connected internally and not recommended as a general-purpose regulated power source. Bypass with a 2.2-µF ceramic capacitor. The high supply option for the internal codec circuits is pin N18 (VDD_RFA).
VREG_GP1	AA22	AO	The output of the first of two general-purpose linear regulators; intended for powering Bluetooth circuits, but could be used for other purposes. Bypass this pin with a 2.2-µF ceramic capacitor and connect it to the desired external circuits.
VREG_GP2	Y22	AO	The output of the second of two general-purpose linear regulators; intended for powering WLAN circuits, but could be used for other purposes. Bypass this pin with a 2.2-µF ceramic capacitor and connect it to the desired external circuits.
VREG_MPLL	T13	AO	Linear regulator output that powers internal baseband PLL functions; not recommended as a general-purpose regulated power source. Bypass this pin with 2.2-µF ceramic capacitor; it is connected to the MPLL circuits internally.
VREG_MSME	W15	AO	Linear regulator output that powers peripheral functions (including EBI1); not recommended as a general-purpose regulated power source. Bypass this pin with a 2.2-µF ceramic capacitor. This pin is connected to several VDD_PX pins, depending upon the application (with appropriate filtering).
VREG_MSMP	W13	AO	Linear regulator output that powers peripheral functions; not recommended as a general-purpose regulated power source. Bypass with 2.2-µF ceramic cap. This pin is connected internally to pad group #7 (VDD_P7).

Table 14-2 Output voltage regulator connections (continued)

Pin name/function	Pin #	Pin type [1]	Functional description
VREG_RFA	W16	AO	Linear regulator output intended for powering internal RF and analog circuits. Bypass with a 2.2-µF ceramic capacitor and connect it to the desired QSC input power pins (with appropriate filtering).
VREG_RFRX2	W21	AO	Linear regulator output intended for powering internal and external receiver circuits. Bypass with a 2.2-µF ceramic capacitor and connect it to the desired QSC input power pins and external RF circuits (with appropriate filtering).
VREG_RFTX2	V21	AO	Linear regulator output intended for powering internal transmitter circuits. Bypass with a 2.2-µF ceramic capacitor and connect it to the desired QSC input power pins (with appropriate filtering).
VREG_TCXO	U22	AO	Linear regulator output that powers TCXO (XO) related circuits. Sensitive to external noise sources; minimize its use for powering external circuits. Bypass this pin with 1.0-µF ceramic cap. Internal TCXO regulator circuits are enabled by internal TCXO controllers.
VREG_USIM	W14	AO	Linear regulator output intended for powering USIM circuits or others. Bypass this pin with a 2.2-µF ceramic capacitor.
VREG_USB_2P6	V15	AO	The first of two USB linear regulator outputs; 2.6 V nominal. If used for the internal USB transceiver, do not use to power any external circuits. If not used for the USB transceiver, this output can be used externally. Bypass this pin with a 1.0-µF ceramic cap.
VREG_USB_3P3	V14	AO	The second of two USB linear regulator outputs; 3.3 V nominal. If used for the internal USB transceiver, do not use to power any external circuits. If not used for the USB transceiver, this output can be used externally. Bypass this pin with a 1.0-µF ceramic cap.
Negative SMPS connections (negative charge pump)			
VREG_NCP	AC20	AO	Negative output voltage; bypass with a 2.2-µF ceramic capacitor and connect to the desired loads (with appropriate filtering). Intended for powering headphone circuits; NCP_FB sense point is connected internally to HPH_VNEG.
NCP_CTC1	AB18	AI, AO	Connect these two pins to the two pins of the external 2.2-µF charge-transfer capacitor.
NCP_CTC2	AB19	AI, AO	
Voltage references			
REF_GND	AB9	AI	The ground for the internal reference – connect as directly as possible to the handset's reference ground.
REF_BYP	AC9	AO	Connect this pin to a 0.1-µF ceramic capacitor with its other side connected as directly as possible to pin AB9 (REF_GND). This capacitor is part of a lowpass filter for the internal reference. Do not load this pin with external circuitry.
REF_ISET	AC10	AI	Connect this pin to a 605 kΩ, ± 1% resistor with its other side connected as directly as possible to pin AB9 (REF_GND).
TX_DAC_BYP	B18	AO	Connect this pin to a 0.1-µF ceramic capacitor with its other side connected to GND_A_RF. This pin provides filtering of an internal reference voltage and should not be loaded externally.

[1] Pin type is AI = analog input or AO = analog output.

[2] The core regulator voltage sensing is achieved internally; there is not a dedicated VREG_MSMC pin. The regulated output of the core SMPS circuit is at the buck inductor output side; this node must be bypassed with a 2.2-µF ceramic capacitor and then routed to the VDD_CORE pins (with appropriate filtering).

14.2 External component requirements

External component requirements for all regulator types are summarized below.

Table 14-3 Regulator external-component requirements

Regulator	Load capacitor [1]		SMPS inductor	
	Value	Type	Value	Type
SMPS boost converter				
5V	10 µF	Ceramic	2.2 µH	High-power chip
SMPS buck converters				
MSMC	10 µF	Ceramic	2.2 µH	High-power chip
RF1	10 µF	Ceramic	2.2 µH	High-power chip
RF2	10 µF	Ceramic	2.2 µH	High-power chip
Linear regulators				
CDC2	2.2 µF	Ceramic	—	—
GP1 (BT)	2.2 µF	Ceramic	—	—
GP2 (WLAN)	2.2 µF	Ceramic	—	—
MPLL	2.2 µF	Ceramic	—	—
MSME	2.2 µF	Ceramic	—	—
MSMP	2.2 µF	Ceramic	—	—
RFA	2.2 µF	Ceramic	—	—
RFRX2	2.2 µF	Ceramic	—	—
RFTX2	2.2 µF	Ceramic	—	—
TCXO	1.0 µF	Ceramic	—	—
USIM	2.2 µF	Ceramic	—	—
USB_2P6	1.0 µF	Ceramic	—	—
USB_3P3	1.0 µF	Ceramic	—	—
Negative charge pump				
NCP	2.2 µF	Ceramic	—	—

[1] All regulators require X5R or X7R ceramic capacitors on their output pins. The designs are stable with half to five times the listed capacitance values, taking into account the worst-case capacitance changes of X5R or XR7 dielectric.

The external SMPS components are especially critical; recommendations are listed in Table 14-4 along with their dimensions.

Table 14-4 External SMPS component recommendations

Component	Manufacturer and description	Dimensions (mm)
Inductor		
2.2 µH		
Capacitor		
10 µF		
2.2 µF		

The boost converter's external Schottky diode should have a low forward voltage for best converter efficiency. The junction capacitance at 0 V should be 60 pF or less to reduce voltage spikes on the output. The reverse-leakage current needs to be low enough to meet USB_VBUS input leakage specifications. Values of 10 µA or less are recommended. Note that with the diode in place there is no protection if VREG_5V is shorted to ground. Table 14-5 provides recommended Schottky diodes with manufacturer information and physical device dimensions.

Table 14-5 Recommended Schottky diodes for boost regulator

Manufacturer and part information	Dimension (mm)
Panasonic MA2ZD18, 500 mA, 20 V	2.5 x 1.7 x 0.7
Rohm RB551V-30, 500 mA, 20 V	2.5 x 1.7 x 0.7

All regulated output voltages are programmable. Detailed descriptions for both regulator circuit types are presented in this chapter; but first, the common voltage reference circuit is discussed.

14.3 Reference circuit

The QSC62x0 regulator circuits, and other internal circuits, are driven by a common, on-chip voltage reference circuit (Figure 14-1). The voltage is trimmed in final device test and the optimal setting is permanently stored in nonvolatile memory (NVM). A precision external resistor (605 kΩ, ±1%) sets the reference circuit's bias current and is connected from pin AC10 (REF_ISET) to ground. The reference circuit's ground is brought off-chip by the dedicated pin AB9 (REF_GND). An on-chip series resistor supplements an off-chip 0.1 µF bypass capacitor at pin AC9 (REF_BYP) to create a lowpass function that filters the reference voltage that is distributed throughout the QSC device. The filtered reference node is buffered and available as an output by using the proper MPP configuration.

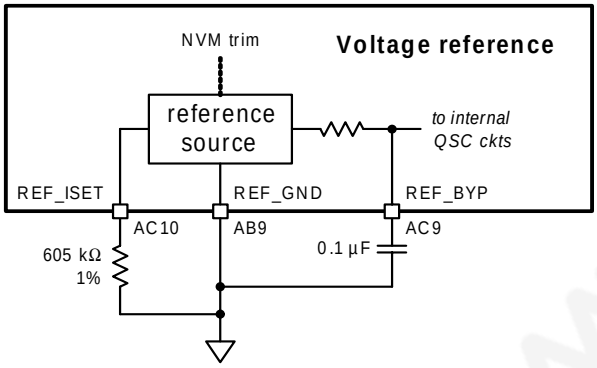


Figure 14-1 Reference voltage circuits

14.4 Switched-mode power supplies

Three types of switched-mode power supplies (or DC-to-DC converters) are implemented by the QSC62x0 device: boost and buck positive voltage converters, and a negative charge pump (NCP) converter. The boost converter provides output DC voltages larger than its input DC voltage and is also known as a step-up converter. Buck converters provide an output voltage smaller than their input voltage and are therefore known as step-down converters. And the NCP converter creates a negative supply voltage for headset circuits.

Both positive converter types are implemented using pulse-width modulation, a technique that gates transistors on and off to vary the output pulse widths, which in turn varies the output DC voltage level. Although the output pulse width varies, the clock rate or operating frequency remains fixed, thereby limiting most of the undesired AC content to the clock rate and its harmonics. The clock rate is programmable, but the TCXO frequency divided by 12 (1.6 MHz) is usually selected. To further minimize switching transients, the switched-mode power supply circuits are gated by a four-phase clock so they are not all switching simultaneously.

To increase efficiency at low currents (such as during the phone's sleep intervals), the positive converters can be run in a low-power mode; the converter is off for long periods, turning on just long enough to maintain the output voltage. In its low-power mode, the boost converter outputs bursts of pulses in a pulse-burst modulation (PBM) mode, whereas the buck converters decrease the rate of pulses using a pulsed-frequency modulation (PFM) mode. Handset designers should be aware that for very light loads, the pulse rate can drop into the audio range and could corrupt audio performance.

The final SMPS creates a -1.8 V supply for capless stereo-headphone drivers using a capacitor-based negative charge-pump switching regulator.

Further discussions of all SMPS converters are given in the following subsections.

14.4.1 Boost (step-up) converter

The boost switched-mode power supply (SMPS) is rated for 600 mA output current and is intended for generating +5 V to power circuits such as the USB-OTG host, white LEDs, and a camera flash. Although rated for 600 mA, higher currents are allowed but higher input voltages may be required and some performance characteristics may become degraded.

The boost converter schematic diagram is shown in Figure 14-2.

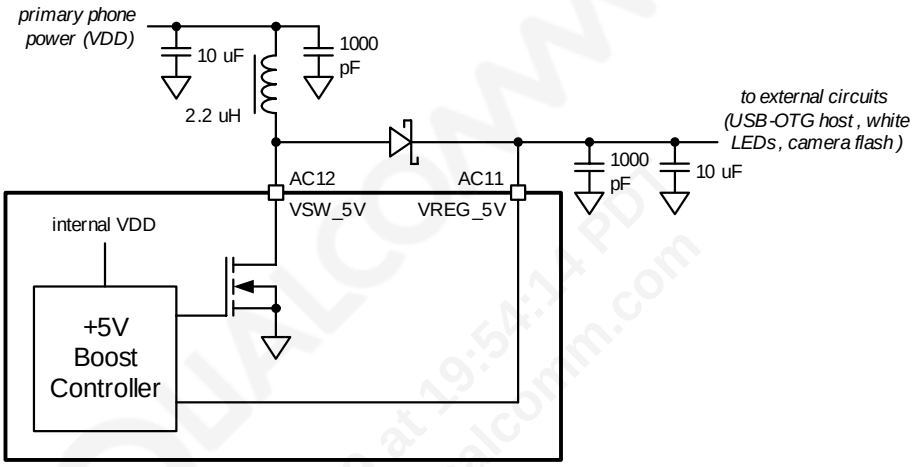


Figure 14-2 Boost converter schematic diagram

Key inductor, capacitor, and Schottky diode recommendations were listed in Table 14-4 and Table 14-5. For high efficiency, an inductor rated to 4 MHz with a series resistance of 0.25 Ω or less should be used. The capacitors need to be ceramic to assure circuit stability; X5R or X7R dielectric types are recommended due to their thermal stability.

The boost converter uses a constant-frequency, current mode control architecture. The main switch (N-channel MOSFET) is internal. At light loads, PBM is used to achieve high efficiency at the price of higher noise. The control block switches to burst mode operation when the error signal falls below some minimum value, which corresponds to a minimum output load current. A zero current detector across the external Schottky diode causes the main switch to turn off when the current reaches zero during discontinuous conduction mode. Current limiting is provided, with the current sensed directly across the main switch.

14.4.2 Buck (step-down) converter

The three buck converters are rated for 500 mA output current each; their intended applications are:

- The MSMC buck converter is intended for powering the processor core circuits.
- The RF1 buck converter is intended as the first regulator that feeds the MSME and RFA linear regulators for subregulation.
- The RF2 buck converter is intended as the first regulator that feeds the MPLL, CDC2, RFRX2, and RFTX2 linear regulators for sub-regulation.

Although rated for 500 mA, higher currents are allowed but higher input voltages may be required and some performance characteristics may become degraded. The buck converters (Figure 14-3) have better efficiency than linear regulators when the input-to-output voltage differential is significant.

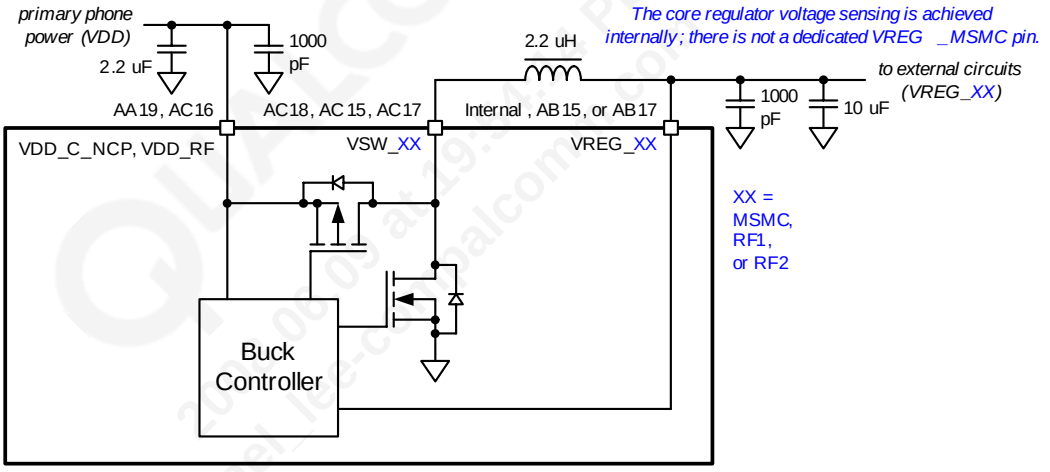


Figure 14-3 Buck converter schematic diagram

The nominal output inductor and capacitor values are 2.2 µH and 10 µF respectively. The capacitor needs to be ceramic to assure circuit stability. See Table 14-4 for recommended inductors and capacitors with manufacturer information and physical device dimensions.

The buck converters uses a constant frequency, current mode control architecture. The main switches (N-channel and P-channel MOSFETs) are internal.

Each buck regulator has two control modes: PFM control for low power operation, and PWM control for normal operation. For best efficiency, the buck regulator switches automatically between PFM and PWM, but it can be also switched manually via software (depending upon the system mode, sleep or active).

14.4.3 Negative charge pump

The QSC62x0 device includes a capacitor-based negative charge pump switching regulator (Figure 14-4) that generates a -1.8 V supply for capless stereo-headphone drivers. This supply is rated for nearly 200 mA.

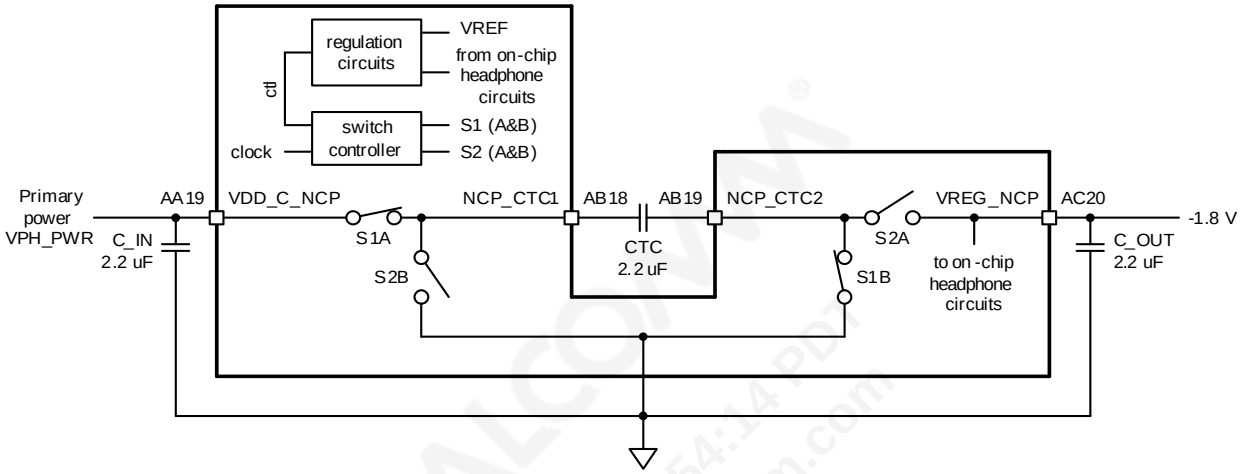


Figure 14-4 Negative charge-pump functional schematic diagram

The negative charge pump requires just three external capacitors: input, output, and charge transfer capacitors. A value of 2.2 µF is recommended for each, but alternate values may be acceptable. Smaller capacitor values results in higher ripple voltages. Although this can be compensated for by increasing the switching frequency, the efficiency decreases. The equivalent series resistance (ESR) of each external capacitor should be less than 0.1 Ω.

An internal clock signal provides the timing for opening and closing the switches, and the regulator feedback is sensed at the headphone circuits (a QSC audio function).

PCB layout is critical to NCP performance. For example, the total series resistance of all the external connections must be minimized by keeping traces short, direct, and as wide as possible. See the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* (80-VF846-5) for more details.

14.5 Linear regulators

A low dropout voltage regulator (Figure 14-5) consists of four functional blocks: 1) a reference voltage, 2) a controlled element, 3) a feedback path, and 4) an error amplifier. These blocks are configured to form a closed-loop control system that regulates the output voltage to match a scaled version of the reference voltage regardless (nearly) of the current delivered to the load.

The reference provides one input to the error amplifier; its second input is a scaled sample of the circuit's output provided by the feedback path. Any difference between these two inputs causes an amplified error voltage that is applied to the controlled element. The closed-loop system varies the controlled element so that the error is zeroed, indirectly adjusting the output voltage until it accurately scales the reference voltage. The output

voltage accuracy is maintained even as the current delivered to the load varies significantly.

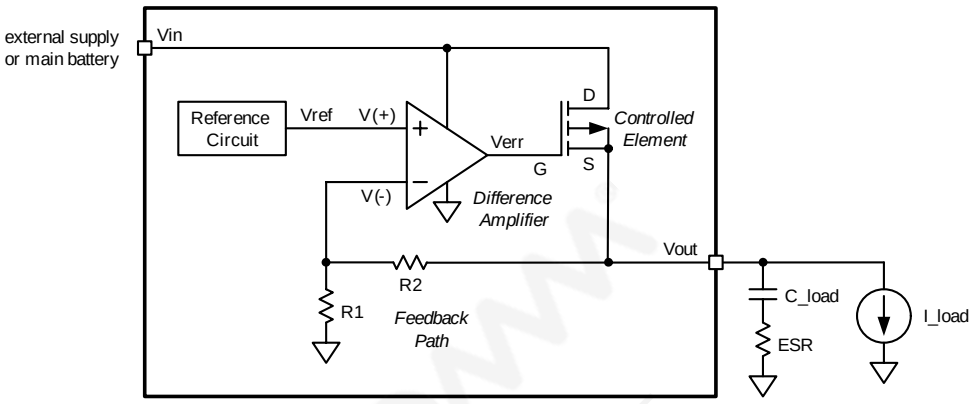


Figure 14-5 Low dropout regulator schematic diagram

The following sections provide additional information about each functional block.

Reference voltage

The QSC device uses a trimmed bandgap reference to generate the reference voltage for the closed-loop regulator — its accuracy and stability scale directly to output voltage performance. Any voltage value can be used as a reference since the closed-loop system includes scaling between its input and output.

Controlled element

The controlled element used in the QSC device is a PMOS transistor that provides two functions: 1) it creates a voltage drop across the gate and source that is varied by the difference amplifier output, and 2) it passes a large range of current between drain and source with little effect on the drain-to-source voltage drop. The current passed by the controlled element varies greatly as circuits turn on and off to meet different handset operating mode demands (sleep, Rx, Rx/Tx, etc.). The PMOS transistor must withstand heat generated by its internal power dissipation (load current times drain-to-source voltage drop).

Feedback path and error amplifier

The QSC uses an operational amplifier (op amp) circuit to implement the feedback path and difference amplifier functions. The regulator output voltage (V_{out}) is divided by the resistor network such that $V(-) = R1xV_{out}/(R1 + R2)$. The op amp presents very high input impedance so it has negligible effect on R1. By the op amp's virtual ground, V(-) = V(+); since V(+) equals the reference voltage (V_{in}), V(+) is constant. The closed loop will achieve $V(-) = V_{in}$ by driving the output voltage to whatever voltage is needed. Substituting and manipulating terms yields the well-known gain factor for a noninverting op amp circuit: $V_{out} = (1 + R2/R1)xV_{in}$. The output voltage is directly proportional to the input (reference) voltage. The op amp output voltage (V_{err}) and controlled element gate-to-source voltage are within the closed-loop and can be considered part of the op amp function for modeling purposes.

Low dropout regulators implemented using this functionality are duplicated twelve times within the QSC device. Three separate designs generate the following voltages:

- Design #1 – rated for 300 mA, intended for powering MSME, MSMP, RFA, and GP2 circuits. MSMP circuits are on during the phone's sleep mode, so this regulator has very low ground current.
- Design #2 – rated for 150 mA, intended for powering CDC2, GP1, MPLL, RFRX2, RFTX2, and USIM circuits. These outputs power sensitive analog and RF circuits, so they provide low noise and low spurious levels.
- Design #3 – rated for 50 mA, intended for powering TCXO, USB_2P6, and USB_3P3 circuits. These are also sensitive circuits, requiring low noise and low spurious levels.

The USIM regulator output includes a MOSFET circuit that pulls down the output pin (VREG_RUIM, pin W14) when the regulator is disabled.

NOTE The USB regulators are intended for powering internal USB transceiver circuits; if the USB transceiver is used, do not use that VREG_USB output (2P6 or 3P3) to power off-chip circuits. Any VREG_USB outputs not used to power on-chip USB transceiver circuits can be used to power other circuits.

NOTE The MSMP regulator should always be kept on at its default 2.6 V setting. It is used for powering internal digital circuits; turning it off or changing its voltage can cause unpredictable results.

14.6 Subregulation

Rather than dropping large voltages across linear regulators (such as 3.6 V down to 1.3 V), the SMPS buck converters efficiently create an intermediate voltage (such as 1.4 V). This intermediate voltage is then used by the linear regulators to create the desired output voltages. This technique is more efficient, yielding less overall power dissipation.

The recommended connections for subregulation are shown in Figure 14-6.

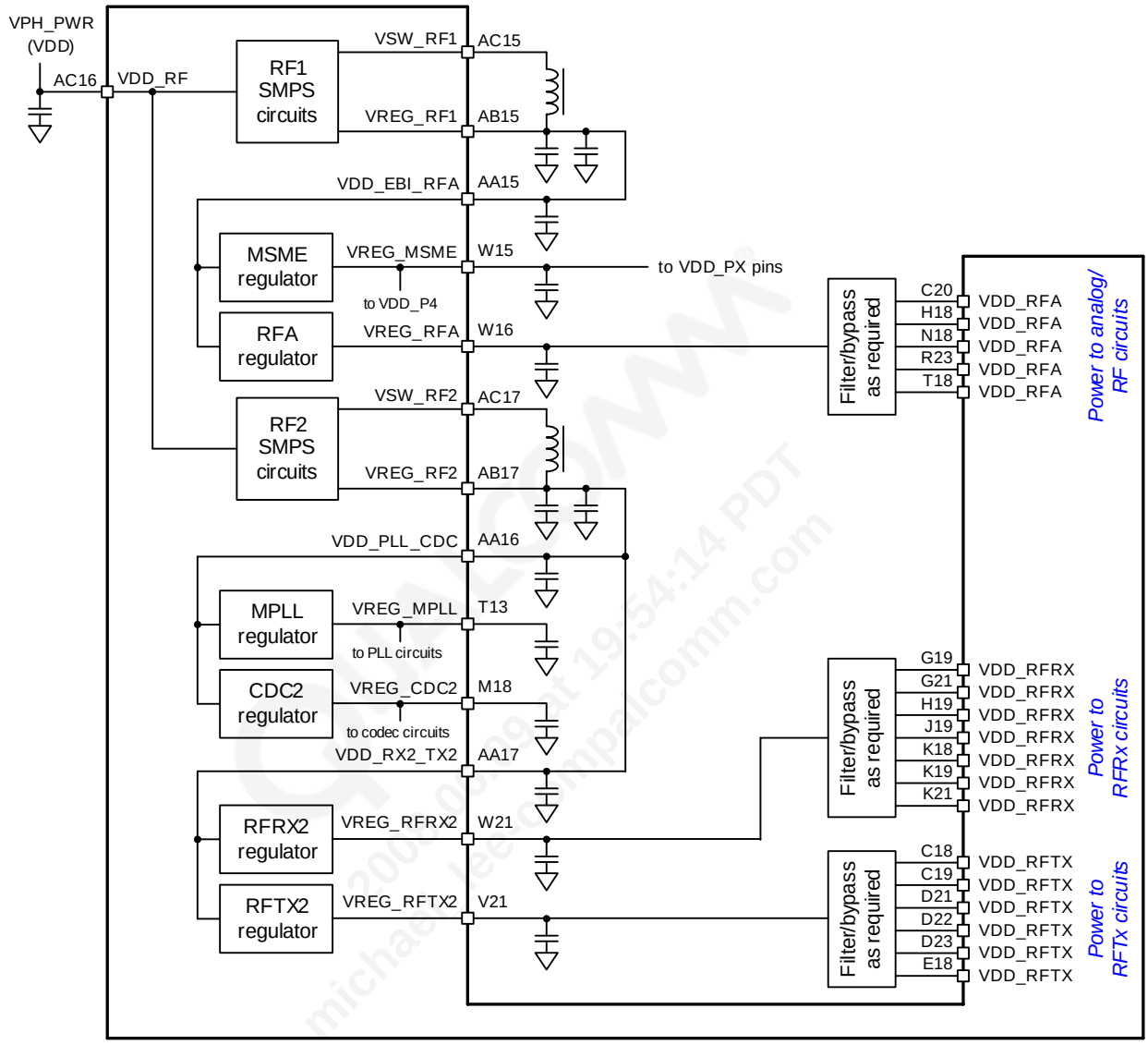


Figure 14-6 Subregulation connections

14.7 Low power modes

The SMPS and linear regulators support low power modes that reduce the quiescent currents. This is especially useful during the phone's sleep mode, enabling maximum standby time. See the *QUALCOMM Single-chip QSC6240/QSC6270 Device Specification* (80-VF846-1) for the current differences in normal and low-power modes.

Different regulator types implement their low-power modes differently, as described below.

Linear regulators

The linear regulators implement their low-power mode by reducing the current required by the feedback loop. During low-power operation, the regulator performance is degraded — lower PSRR, less output current capability, etc. If the load is greater than 1 mA, the output voltage is likely to be out of specification.

Boost SMPS

At light loads, the boost SMPS switches from PWM to PBM. This achieves high efficiency at the price of higher noise. The control block switches to burst mode operation when the error signal falls below some minimum value, which corresponds to a minimum output load current. A zero current detector across the external Schottky diode causes the main switch to turn off when the current reaches zero during discontinuous conduction mode.

For normal (active) phone operation, the PWM mode should be used.

Buck SMPS

The buck SMPS implements its low power mode by switching to PFM from its usual PWM mode. In PFM mode, the controller shuts down and a comparator is left on that monitors the output voltage. Starting with the pass device off, eventually the output dips below the programmed output voltage. The pass device is then turned on until the output voltage slightly exceeds the programmed voltage (single pulse), and then it is turned off. The on/off process repeats. If the buck SMPS is loaded too heavily in PFM mode, the output ripple is degraded.

During PFM operation, the pulse frequency varies with load current while the ripple stays constant at about 30 to 50 mV peak-to-peak. Depending on the load, the pulse frequency can drop into the audio range and could become audible if it couples into the audio system.

For normal (active) phone operation, the PWM mode should be used.

14.8 Special USB_3P3 regulator considerations

The 3.3 V USB regulator is powered by one of two voltage sources (Figure 14-7):

- VDD_USB – the primary phone power supply (VPH_PWR at pin V16) is used during USB-OTG operation. The handset recognizes that a USB peripheral device is connected and the QSC device provides DC power to the peripheral device and to the handset transceiver circuits. The peripheral device is indirectly powered off VREG_5V through the USB_VBUS pin, and the handset's transceiver circuits are powered off VREG_USB_3P3 (which is regulated off the VDD_USB voltage). The VDD_USB voltage is the default supply for the USB regulator, and both the 5 V and the USB_3P3 regulators must be software enabled (they default to their off states).
- USB_VBUS – when a host USB device is connected, it provides power to the handset. In this case, software needs to select the external power source as the USB regulator supply voltage. The USB regulator still powers the USB transceiver circuits, but now it is regulating off the USB_VBUS voltage rather than the VDD_USB voltage.

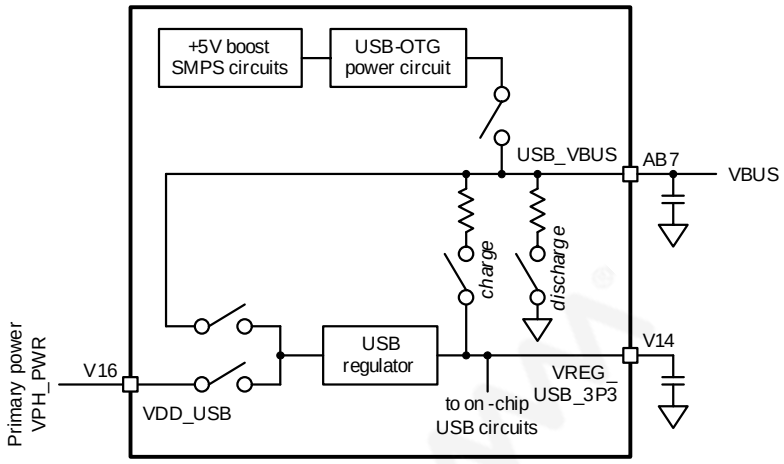


Figure 14-7 USB regulator input-voltage sources

As suggested above, if a USB device is connected, software must make different selections based upon whether the handset is in its USB-OTG mode or not. For OTG operation, use the default source for the USB_3P3 regulator (VDD_USB) and enable both the 5 V and the USB_3P3 regulators. For non-OTG operation, the external source (at the USB_VBUS pin) is selected to power the USB_3P3 regulator and the USB_3P3 regulator is enabled. The VREG_5V is not used as the regulator supply but could be enabled and used for other purposes.

NOTE The two switches on the input of the USB regulator cannot both be closed at the same time. Thus, it is not possible to route current directly from VDD_USB to VBUS.

15 General Housekeeping

The QSC62x0 device includes many circuits that support handset-level housekeeping functions — various tasks that must be performed to keep the handset in order. Integration of these functions reduces the external parts count and the associated size and cost. Housekeeping functions include an analog multiplexer with input scaling, system clock circuits, real-time clock for time and alarm functions, and overtemperature protection. All are discussed in this chapter.

15.1 General housekeeping connections

General housekeeping connections are listed and described in Table 15-1. Detailed functional block diagrams and interface circuit diagrams (where appropriate) are included for each circuit type in their section within this chapter.

Table 15-1 General housekeeping connections

Pin name/function	Pin #	Pin type [1]	Functional description
XTAL_19M_IN	V23	AI	External connections required for generating the TCXO signal; two possible configurations [2]: 1) If an external 19.2-MHz crystal is used: each pin is connected to one side of the crystal. 2) If an external VCTCXO module is used: XTAL_19M_IN (pin V23) is AC-coupled to the VCTCXO output; XTAL_19M_OUT (pin U23) is not used and must be left unconnected.
XTAL_19M_OUT	U23	AO	
XO_EN_GP1	W11	DI	Enable this for the general-purpose XO controller circuit #1; enables the XO_OUT_GP1 (W10) output signal.
XO_OUT_GP1	W10	DO	The first of two general-purpose 19.2 MHz outputs; jointly enabled by the internal bus and XO_EN_GP1 (W11). This XO output is a square-wave that is powered from the VREG_GP2 voltage.
XO_OUT_GP2	V18	DO	The second of two general-purpose 19.2 MHz outputs; enabled by the internal bus only. This XO output is a sine-wave that is powered from the VREG_TCXO voltage.
XO_ADC_REF	W23	AI	Reference voltage for on-chip XO tuning ADC circuit; connect to the same *clean* supply that is used for the XO thermistor.
XO_ADC_IN	W22	AI	XO ADC circuit input voltage; connect to the XO thermistor network analog output voltage.
XTAL_32K_IN	AA23	AI	Each pin is connected to one side of the 32.768 kHz crystal.
XTAL_32K_OUT	AB23	AO	
SLEEP_CLK	V12	DO	Sleep clock output. This pin is connected to baseband functions internally; it can also be connected to the FM radio.

Table 15-1 General housekeeping connections (continued)

Pin name/function	Pin #	Pin type [1]	Functional description
R_REF_EXT	AA9	AI	External USB reference resistor; 6.34k to ground.
VREF_THERM	AC13	AO	Output reference voltage for an external thermistor network.
AMUX_IN	any MPP	AI	Any MPP can be configured as an input to the analog multiplexer whose output is routed to the HKADC (when selected).

[1] Pin type is AI = analog input, AO = analog output, DI = digital input, or DO = digital output.

[2] See the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* (80-VF846-5) for details about 19.2-MHz XO and VCTCXO implementations.

15.2 Analog multiplexer with input scaling

The QSC62x0 device includes a 16-to-1 analog multiplexer (Figure 15-1) to select a single analog signal for internal routing to the HKADC for analog-to-digital conversion. This allows handset software to monitor various external supply voltages, charging current, auxiliary inputs, and the die temperature using a single HKADC. The sixteen multiplexer inputs are summarized in Table 15-2.

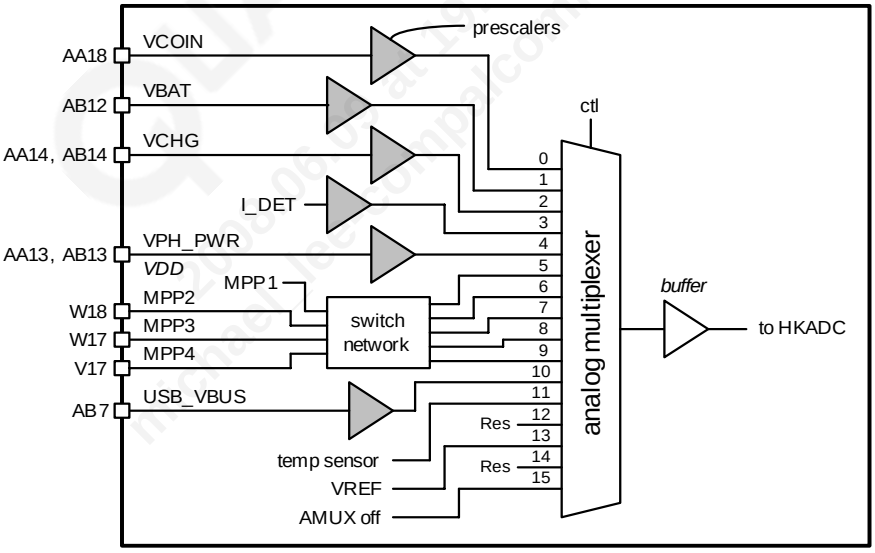


Figure 15-1 Analog multiplexer and input-scaling circuits

Table 15-2 Analog multiplexer inputs

Input #	Affiliated QSC pins	Functional description
0	AA18 – VCOIN	Monitors the coin cell voltage; this input path includes a fixed gain of TBD. Input range = 0 to 3.6 V.
1	AB12 – VBAT	Monitors the main battery voltage; this input path includes a fixed gain of TBD. Input range = 0 to 4.5 V.
2	AA14 and AB14 – VCHG	Monitors the external (charger) supply voltage; this input path includes a fixed gain of TBD. Input range = 0 to 18 V.
3	None – internal current sensor (I_DET)	Monitors the charging pass transistor current: battery charging current, QSC supply current, and the phone's operating current drawn from V_{DD}.
4	AA13 and AB13 – VPH_PWR	Monitors the primary phone power-supply voltage (V_{DD}); this input path includes a fixed gain of TBD. Input range = 0 to 4.5 V.
5 - 9	Multipurpose pins (4) MPP1 (internal); W18 – MPP2; W17 – MPP3; V17 – MPP4	These input paths include a switch matrix that allows any of them to be applied as inputs to the analog MUX. Input range = 0 to 2.5 V.
10	AB7 – USB_VBUS	Monitors the external USB supply voltage; this input path includes a fixed gain of TBD. Input range = 0 to 5.25 V.
11	None – internal temp sensor	Monitors the on-chip temperature sensor. Input range = 30 °C to +150 °C; output range = 1.216 to 2.116 V.
12	None – reserved	This input is reserved for internal test functions.
13	None – internal VREF	Monitors the on-chip reference voltage (V_{REF}). Input voltage = 1.25 V typical, output voltage = 1.25 V typical.
14	None – reserved	This input is reserved for internal test functions.
15	None – multiplexer off	Disables the multiplexer and disconnects the inputs.

NOTE The output voltage from the buffer amplifier that drives the HKADC cannot reliably go below 0.05 V and should not be used below this lower limit.

NOTE Gain and offset errors vary between multiplexer channels; cal values apply to the specific channel being calibrated only. Calibrate each channel separately.

The analog multiplexer output is connected directly to one of the HKADC MUX inputs. The configurable HKADC provides sufficient resolution for all intended applications.

15.3 System clocks

PM functions include several clock circuits (Figure 15-2) whose outputs are used for general housekeeping and elsewhere within the handset system. These circuits include the 19.2 MHz source, controllers, and buffers; RC oscillator; 32 kHz crystal oscillator; SLEEP clock; and SMPS clocks. As depicted in Figure 15-2, some circuits offer multiple implementation options. Detailed descriptions of all these functions and their options (where applicable) are presented in the following subsections.

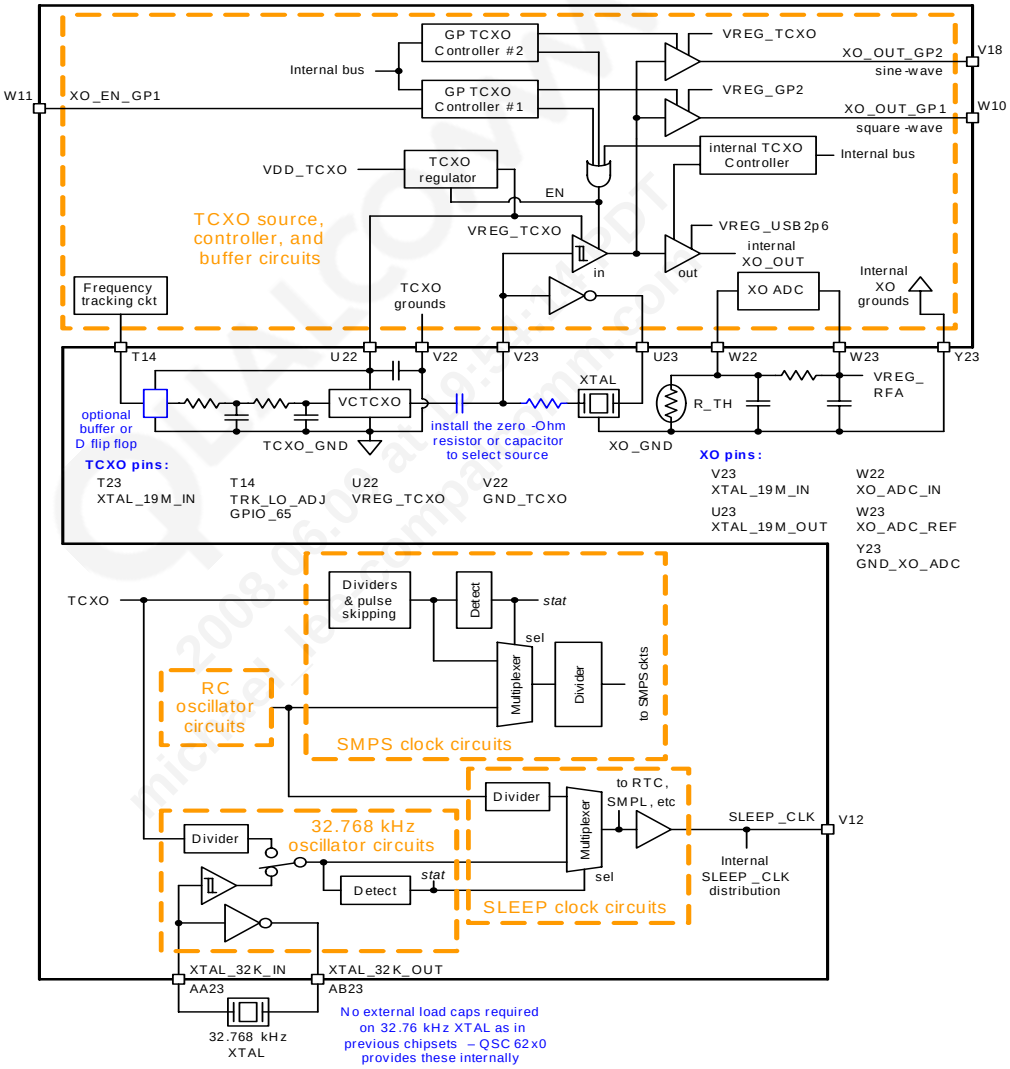


Figure 15-2 System clocks functional block diagram

If a coin cell is included within the handset, it may be used to power the 32.768 kHz crystal oscillator, real-time clock, and some registers if the main battery is removed. This coin cell is a keep-alive power source that can be charged by the QSC device (see Section 13.13 for details).

15.3.1 19.2 MHz TCXO source, controllers, and buffers

The QSC62x0 device includes new and improved search and acquisition algorithms that eliminate the need for a VCTCXO module (typically ± 2.5 ppm), allowing a less expensive 19.2 MHz crystal (typically ± 12 ppm) to be used; this is referred to as the XO circuit.

NOTE Initial board designs must accommodate both the XO circuit and VCTCXO solutions. A final decision on which technique will be commercially supported is pending. Until that final decision is made, both solutions must be on product PCBs to allow assembly of either without revising the PCB layout.

Detailed layout guidelines are provided in the *QUALCOMM Single Chip QSC240/QSC6270 Design Guidelines*, 80-VF846-5.

Functional descriptions and interconnection details are provided below for both 19.2 MHz sources — the XO circuit and the VCTCXO — followed by a discussion of the controller and buffer circuits.

15.3.1.1 19.2 MHz TCXO source

The QSC device supports two methods for generating the 19.2 MHz TCXO signal that is used throughout the handset as the primary timing and frequency reference:

- A free-running crystal oscillator using an external crystal supplemented by on-chip inverters, buffers, and thermal compensation
- A frequency tuned external VCTCXO module

NOTE Regardless of the method, XTAL_19M_IN and XTAL_19M_OUT must not be loaded by external circuits. The QSC provides two XO signal outputs for driving external loads: is available at pins W10 and V18 (XO_OUT_GP1 and XO_OUT_GP2).

The following paragraphs describe both methods and a technique for selecting between them.

Free-running crystal oscillator

The first method (Figure 15-3) uses an external 19.2 MHz crystal to generate the signal and a tightly coupled thermistor network to monitor its temperature. The thermistor must be located very close to the crystal; they must share the same surface ground fill (an island of ground fill) and should be thermally isolated from the QSC device to prevent heating.

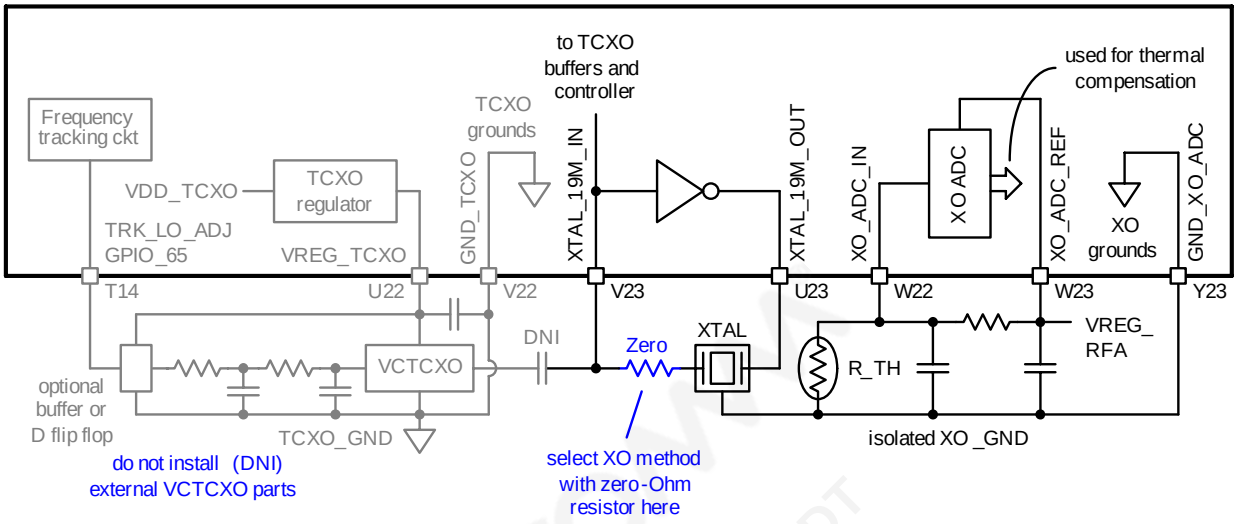


Figure 15-3 Free-running crystal oscillator (XO) method

The crystal ground is a surface layer island that must be isolated from other ground fill areas; do not connect it to PCB ground or to any other ground grouping. XO ground connections are critical to thermal management, See the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* (80-VF846-5) for details.

The output of the thermistor network is the node between the resistor and the thermistor; this analog voltage is routed to QSC pin W22 (XO_ADC_IN). Pin W23 (XO_ADC_REF) is connected to VREG_RFA — the same supply used to bias the thermistor network. This ensures that the on-chip ADC is using the same voltage reference as the thermistor network.

The XO_ADC_IN signal requires a 0.01 µF filter capacitor directly between the thermistor output node and the QSC input (pin W22), placing the capacitor very close to the QSC device. Likewise, XO_ADC_REF requires a 0.1 µF decoupling capacitor that should be fairly close to pin W23. Both capacitors must be grounded to the island ground only.

External VCTCXO module

The second 19.2 MHz TCXO option uses an external VCTCXO module (the typical implementation of previous generation MSM devices). In this case (Figure 15-4), the crystal oscillator is replaced by the VCTCXO module and the TRK_LO_ADJ output tunes the VCTCXO.

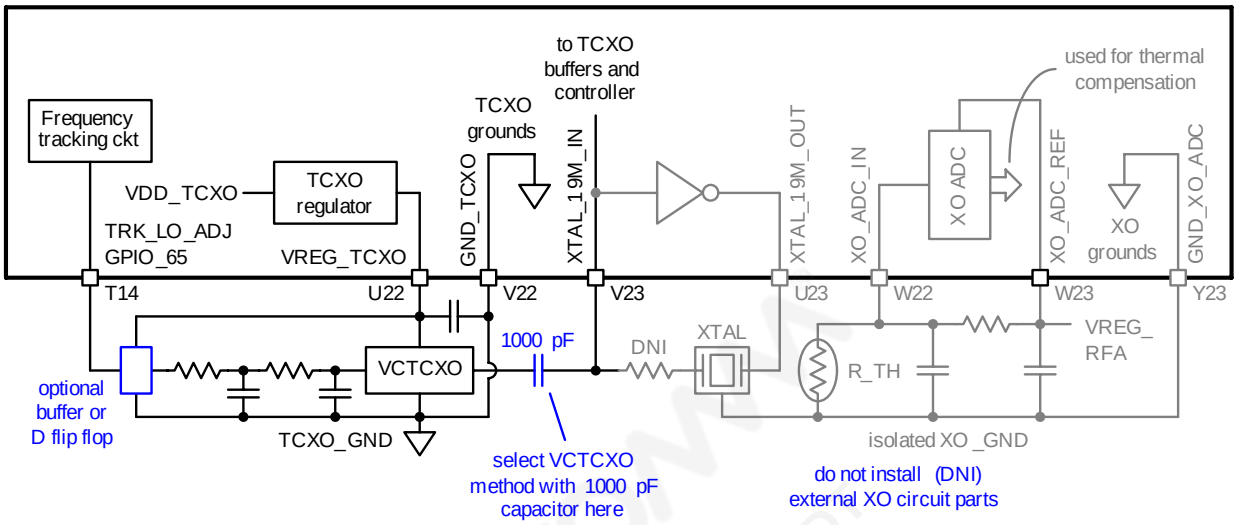


Figure 15-4 External VCTCXO module method

The VCTCXO module tuning port is driven by the QSC TRK_LO_ADJ output (pin T14, GPIO_65). This is a PDM output signal that is RC-filtered near the VCTCXO to provide an analog control voltage. A two-pole RC-filter is required, located very close to the tuning port. This sensitive analog signal must be kept away from all possible noise sources (audio, RF, high frequency digital, switching power signals, etc.).

The example shows an option to include either a buffer or a D flip-flop between the QSC device and the RC-filter. QSC pin T14 is routed to the buffer or flip-flop input; the buffer or flip-flop output is routed to the VCTCXO tuning port through the two-pole RC filter. Circumstances that require the addition of this buffer are discussed in *radioOne Solutions for GPS Position Location Application Note* (80-V1647-1).

The VCTCXO is selected as the 19.2 MHz source by omitting the resistor near the crystal output and installing the resistor near the VCTCXO output. The output of the installed resistor is routed directly to XTAL_19M_IN (QSC pin V23). This signal routing must avoid any potential noise sources (audio, RF, high frequency digital, switching power signals, etc.). XTAL_19M_OUT (QSC pin U23) is left unconnected.

The VCTCXO module's DC power is delivered from the QSC device's VREG_TCXO output (pin U22). This supply is bypassed near the VCTCXO module. If the buffer or flip-flop is used, its power must also be provided by VREG_TCXO. A common bypass capacitor is recommended (shared by the VCTCXO and buffer or flip-flop). VREG_TCXO does not power any other components, so it should be a very clean supply.

Like the XO circuit discussed earlier, the VCTCXO implementation requires special grounding. The grounds of VCTCXO-related components must connect directly to an isolated surface ground fill area, including:

- VCTCXO module
- Buffer or flip-flop (if used)
- Two capacitors within the two-pole RC-filter
- VCTCXO (and buffer or flip-flop, if used) power-supply bypass capacitor

- VDD_TCXO bypass capacitor for pin U21 (not shown; used to power the TCXO regulator)
- QSC pin V22 (GND_TCXO)

The TRK_LO_ADJ signal is available through proper configuration of GPIO_65. This pad is powered off VDD_P7, an internal voltage.

Selecting the 19.2 MHz source

The XTAL_19M_IN input (pin V23) connects to two components that allow selection of the 19.2 MHz source: either a zero-Ω resistor selects the crystal or a 1000 pF capacitor selects the VCTCXO. Since only one of these components can be installed and active within a phone, special routing is required. Using one of two components allows the unused circuits to be totally disconnected, thereby avoiding any stubs on the XTAL_19M_IN input. The two components can share a single pad between them, with this common pad routing to the XTAL_19M_IN input. This minimizes the trace length added to the XTAL_19M_IN input while supporting both sources. The two components are shown in Figure 15-3 and Figure 15-4, with all unused components grayed out in both cases.

15.3.1.2 TCXO controllers and buffers

When a WCDMA handset is not actively processing a call, its receiver alternates between periods of reception and sleeping and is said to be in its slotted mode. The receive/sleep cycle is called a slot cycle and varies from system to system. Since WCDMA handsets usually operate in the slotted mode, with the sleep portion significantly longer than reception, the power dissipation while sleeping is critical to standby time and battery life. Ideally, all circuits would be off until the receiver has turned on. Once the receive interval was satisfied and the handset was sure a call was not being attempted, all circuits would be turned off. This is a reasonable process for most circuits, but one circuit requires its DC power to be on for a suitable duration to assure proper operation — the TCXO requires a warm-up time.

The QSC62x0 device optimizes TCXO operation during the slotted mode using a dedicated controller that enables and disables appropriate circuits in the proper sequence (see the TCXO source, controller, and buffer circuits portion of Figure 15-2). An internal TCXO controller is enabled by the baseband circuits via the internal bus. When enabled, the controller immediately turns on the TCXO regulator and the input buffer and begins counting either sleep clock cycles or TCXO cycles, depending on its operating mode (Figure 15-5). Within a predictable period, the TCXO signal will stabilize near its final frequency; this predictable period (or number of clock cycles) is programmed into a timer within the controller. When the timer expires, the internal output buffer is enabled, synchronized with the TCXO input such that the internal TCXO_OUT signal is glitch-free. Only valid TCXO pulses are output. This output is distributed internally only. Two additional TCXO controller and buffer circuits are available that generate XO_OUT signals for external distribution (discussed later in this section).

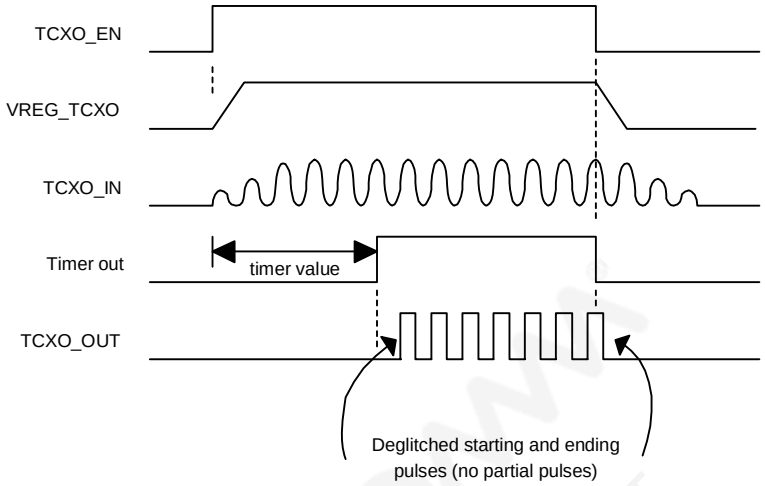


Figure 15-5 TCXO controller timing

Since TCXO circuits do not require any special cool-down intervals, the TCXO regulator is turned off along with the input and output buffer circuits soon after the TCXO_EN signal is removed (via the internal bus); the circuit controls are registered to ensure the last output clock pulse is clean.

On powerup, the QSC device defaults to a software-controlled mode with the TCXO defaulted on. This assures the baseband circuits will always have a clock available immediately at powerup.

In addition to the internal TCXO controller and buffer circuits, identical controller and buffer circuits are available for clocking external functions (such as WLAN, Bluetooth, and GPS circuits), allowing those functions to operate even while the QSC is in its sleep mode. The first general-purpose controller (GP1) is enabled by XO_EN_GP1 at pin W11 or by the internal bus; the second (GP2) is enabled by the internal bus only. The first is a square-wave output at pin W10 (XO_OUT_GP1) that is powered by VREG_GP2; the second output is a sine-wave at V18 (XO_OUT_GP2) that is powered by VREG_TCXO. A typical application (like the reference design) uses XO_OUT_GP1 for Bluetooth and XO_OUT_GP2 for GPS.

15.3.2 RC oscillator

The RC oscillator is the default clock source during QSC62x0 powerup. It continues to be used until the QSC clears interrupts that cause switchovers to the 32.768 kHz crystal oscillator and the 19.2 MHz TCXO source. Transitions between the clock sources are synchronized to ensure valid pulses at the SLEEP_CLK output and the SMPS input (wide pulse widths, no glitches). Once the switchovers are made, the internal RC oscillator is powered down to reduce power consumption. This circuit draws too much current to run off the coin-cell keep-alive battery; it requires the QSC device to be on.

The RC oscillator circuit starts again automatically if detection circuits indicate that either of the following occurs:

- The selected SLEEP_CLK source stops oscillating (either the 32.768 kHz crystal oscillator or the 19.2 MHz TCXO source, whichever is selected).
- The 19.2 MHz TCXO source stops oscillating (and therefore the SMPS clock source) while the buck converter (VREG_MSMC) is enabled.

15.3.3 32.768 kHz crystal oscillator

See the 32.768 kHz oscillator circuits portion of Figure 15-2 for a functional block diagram.

The 32.768 kHz crystal oscillator is the primary sleep clock source; this signal is normally generated by an external crystal supplemented by a QSC62x0 inverter and buffer. The QSC62x0 device includes the load caps previously required externally for the 32.768 kHz XTAL. No external load capacitors are required. An external oscillator module could be used rather than a crystal by connecting the module output directly into XTAL_32K_IN (pin AA23) and with proper software control. In either case, the crystal I/Os are not buffered outputs and are incapable of driving a load; the oscillator will be significantly disrupted if either I/O is loaded. Pin AB23 (XTAL_32K_OUT) should be unconnected when using an external oscillator module.

The TCXO source (divided by 586) can be used as the SLEEP_CLK source in place of the 32.768 kHz crystal oscillator if the handset application does not support the SMPL feature or the real-time clock (and its calendar and alarm functions are not used). This eliminates the external 32.768 kHz crystal, but increases the sleep-mode current consumption; the 32.768 kHz oscillator consumes about 1 µA, while the 19.2 MHz oscillator consumes about 100 µA. The input to the SLEEP_CLK circuit is selected by software.

NOTE If the SMPL feature or RTC functions are supported, the 32.768 kHz crystal oscillator is required.

The selected SLEEP_CLK source is monitored to ensure continuous oscillation. If the source stops oscillating, a multiplexer automatically switches to the RC oscillator and an interrupt is generated. Narrow pulses at the SLEEP_CLK output may occur during this switchover.

If the 32.768 kHz crystal oscillator is used, once it is powered up and qualified, it continues to run as long as a valid power supply is present (external supply, battery, or coin cell), even when the QSC device is powered down. This provides a continuous and accurate 32.768 kHz source to the real-time clock, maintaining the correct time. If a coin cell is not installed, the oscillator halts when power from the external supply and main battery are removed.

The 32.768 kHz crystal oscillator dissipates little power, adjusting its bias current to the minimum required to maintain oscillation. Since the oscillator drives SMPL timers, it operates with supply voltages as low as 1.5 V. This allows a capacitor (a lower cost component) to replace the coin cell while still supporting the SMPL feature.

NOTE The connections between the crystal and the QSC device are highly sensitive analog traces that must be protected against corruption from digital clock and logic signals. PCB layout recommendations are presented in the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* (80-VF846-5).

15.3.4 Sleep clock

See the SLEEP clock circuits portion of Figure 15-2 for a functional block diagram.

The SLEEP_CLK output is derived from one of three sources:

- RC oscillator – an on-chip circuit with coarse frequency accuracy; not used in a normal mode. This path includes a divide-by-100 circuit that reduces the 3.2 MHz RC oscillator output to a 32 kHz clock (nominal values).
- A 32.768 kHz crystal source, usually implemented using an internal inverting buffer and an external crystal plus two capacitors. This low power source can have high accuracy and stability, depending upon the external crystal.
- The 19.2 MHz TCXO source, divided by 586. This can be used as the SLEEP_CLK source (32.7645 kHz nominal) only if the RTC functions and SMPL features are not supported. If RTC or SMPL are supported, the external 32.768 kHz crystal is required.

The sleep clock is available at pin V12 (SLEEP_CLK); it toggles only when the QSC62x0 device is on and stays low when the device is off even though the crystal oscillator continues to run.

The RC oscillator should only be used as the sleep clock and real-time clock (RTC) sources during initialization, or if the crystal oscillator fails. It should not be used in a normal mode because of its coarse accuracy and extra current consumption.

15.3.5 SMPS clock

See the SMPS clock circuits portion of Figure 15-2 for a functional block diagram.

The switched-mode power supplies are driven by one of two clock sources:

- RC oscillator – an on-chip circuit with coarse frequency accuracy; not used in a normal mode. The nominal RC oscillator frequency is at the desired input clock rate of 3.2 MHz, so a divider is not used.
- 19.2 MHz TCXO source – this path includes a programmable and variable divide-by-6 circuit that reduces the 19.2 MHz signal to the desired 3.2 MHz input clock (nominal values).

The TCXO source is normally used, with a detector circuit continually monitoring its signal. If this source stops oscillating, the QSC automatically switches to the RC oscillator and generates an interrupt.

The TCXO path to the SMPS drivers includes programmable counter circuits that are capable of skipping pulses (also programmable). This allows adjustment of the converters' switching frequency so that discrete spectra due to transients can be shifted to minimize RF interference. The nominal SMPS output clock rate, the rate of the switching power supply, is 1.6 MHz (3.2 MHz divided by 2). A phase-quadrature circuit is used for this divider that minimizes transients since the SMPS circuits are switched at different times (not all at once).

15.4 Real-time clock and related functions

The real-time clock (RTC) and its related functions (Figure 15-6) are implemented by a 32-bit real-time counter and four 8-bit alarm registers, both settable in one-second increments. Present time is set relative to a user-definable reference such as the CDMA reference time (12:00:00 AM on January 1, 1980). Time coding and decoding is handled in software.

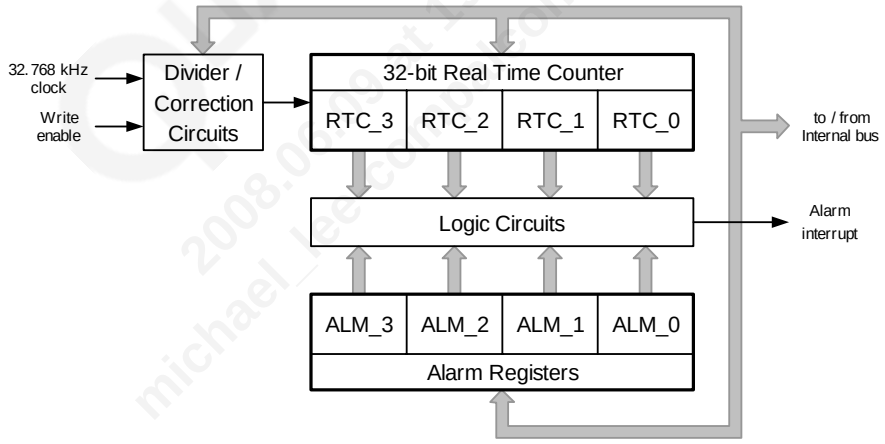


Figure 15-6 Real-time clock and related functions block diagram

The primary input to the RTC circuits is the 32.768 kHz clock from the crystal oscillator. Since the minimum time increment within the RTC is one second, this clock input is nominally divided by 32,768 before application to the counter circuits. To improve the RTC accuracy when the installed crystal oscillator is not precisely 32.768 kHz, a programmable clock adjustment (or correction) is provided. The adjustment changes the nominal divide-by-32768 over a range of ±192 parts-per-million (ppm) with a resolution of 3.05 ppm. If this feature is not used, the default leaves the divider at its nominal value and the RTC accuracy is determined entirely by the crystal oscillator.

Time is set by setting the write enable which stops the clock and clears the divider. The RTC counter is set with the desired value then the write enable is cleared to start the clock and release the divider. There is no default setting for the RTC counter.

Time is read from the 32-bit counter value, comprised of four 8-bit bytes. The read is performed in order from the least significant to most significant bytes. These 32 bits represent the number of seconds from reference time.

The alarm is set by programming the four 8-bit alarm registers to the desired values. Logic circuits continually compare the RTC counter output to the alarm setting; an alarm interrupt is triggered when the two 32-bit values match. See Section 16.5 for details about masking and unmasking interrupts. To disable the alarm, set the alarm registers to values lower than the RTC counter.

There is no default setting for the alarm register, but the interrupts are masked by default.

When the phone is off, the 32.768 kHz crystal oscillator and RTC continue to run off the main battery. If the main battery is not present, these circuits run off the coin cell attached to VCOIN. A valid voltage at VCOIN is also required to run the timer for the SMPL feature. If only the SMPL feature is required, a capacitor can be used in place of the coin cell. See Section 13.13 for details.

An interrupt is generated if the coin cell voltage drops too low (and the main battery is not present). If this interrupt occurs, the RTC might be corrupted. A different interrupt is generated if the crystal oscillator stops; this signifies that handset timing is no longer accurate. Again, the RTC is corrupted.

15.5 32.768 kHz crystal oscillator and RTC power source

The crystal oscillator circuit and the real-time clock functions are powered by one of three voltage sources.

- If PON_RESET_N is high, the QSC62x0 device is fully on, and so VREG_MSMP is the power source.
- If PON_RESET_N is low, the QSC device is in one of its off states. If the device is off and its supply voltage (V_{DD}) is less than 2.4 V, the internal logic supply (dVdd[1]) is the power source.
- If the device is off, its supply voltage is less than 2.4 V, and its internal logic supply (dVdd) is also less than 2.4 V, the coin cell (V_{COIN}) is the power source.

If the coin cell voltage drops too low, the sleep clock will halt.

[1] QSC62x0 logic is powered by an internal supply node called dVdd. When PON_RESET_N is high, dVdd is connected to VREG_MSMP. When PON_RESET_N is low, an internal regulator is enabled that generates dVdd (typically 2.8 V) from the best source available: VDD, VBAT, VCHG, or VBUS. dVdd is used to power all logic – except logic backed up by the coin cell – so the logic can function as long as a valid source is applied.

15.6 Overtemperature protection

The QSC62x0 device provides overtemperature protection in multiple stages, depending on the level of urgency. Interrupts are generated and appropriate action is taken as the internal temperature exceeds a set of thresholds:

- Beyond the lowest threshold, an interrupt is generated without shutting down.
- When a second threshold is exceeded, an interrupt is generated and nonessential functions such as LED drivers, speaker drivers, etc. are shut off.
- Beyond the third threshold, an interrupt is generated and the phone is shut off.

Temperature hysteresis is incorporated so the internal temperature must cool to below the lowest threshold (into its normal operating range) before the device can be powered on.

16 PM Interfaces and Multipurpose Pins

A variety of power management (PM) interfaces extend the QSC62x0 capabilities and reduce the external support circuits required in typical handset applications. The QSC device's PM interfaces include:

- Four configurable multipurpose pins (MPPs) – three are externally accessible
  - Two can also be used as high-current drivers intended for lighting the keypad and the LCD.
- A vibration motor driver, compatible with 1.3 to 3.0 volt devices (programmable), that supports silent alarms
- A programmable class-D speaker driver for audible alarms and higher audio power applications such as speakerphones or melody ringers (500 mW typical)
  - By powering the speaker driver circuit off the on-chip VREG_5V supply, audio power can be increased to 1 W.
- One-touch headset control (headset send-end detect) and microphone bias
- Monitoring circuits that trigger power-on or power-off sequences
- An interrupt manager that coordinates status messaging and responses with the baseband circuits
- A power amplifier controller with an on-chip regulator to generate its own supply voltage, and five outputs for controlling WCDMA and GSM PAs

All of these circuits and functions are addressed in this chapter.

16.1 Interface and multipurpose pin connections

External connections supporting these functions are listed and described in Table 16-1. Detailed functional diagrams and interface circuit diagrams (where appropriate) are included for each circuit type in their section within this chapter.

Table 16-1 PM interface functions (IC-level and user-level)

Pin name/function	Pin #	Pin type [1]	Functional description
User interface drivers			
KPD_DRV_N (MPP3)	W17	AO	These two pins are not only MPPs, they are also high-current drivers - they provide dual functionality. Pins W17 and V17 are both expected to be configured as a high-current driver and used as the keypad and LCD backlight drivers, respectively.
LCD_DRV_N (MPP4)	V17	AO	
VIB_DRV_N	U18	AO	Connect to the vibration motor (-) terminal. The (+) terminal of the motor connects to VDD. An external diode is required to protect this pin from the motor's inductive kickback.
SPKR_OUT_P	AB20	AO	Speaker driver (+) output. Connect directly to the speaker.
SPKR_OUT_M	AB21	AO	Speaker driver (-) output. Connect directly to the speaker.
HSED_BIAS	U19	AO	Headset send/end detect input and microphone bias output
HEADSET_DET_N GPIO_41	H12	AI	Detects headset insertion and removal
Power-on circuit connections			
KPD_PWR_N	AA10	AI	Connect to the keypad power button. This signal is pulled up internally to dVDD. When the QSC device is off, pulling this pin low initiates a powerup and generates an interrupt.
PON_RESET_N	V10	DO	Connected internally to RESIN_N; logic low causes the baseband circuits to reset
PS_HOLD	V11	DI	Baseband circuits drive this input high to keep power on, low to shut down.
Power amplifier controller			
U_PA_ON_0	AA21	DO	First of three control bits for UMTS power amplifiers
U_PA_ON_1	AB22	DO	Second of three control bits for UMTS power amplifiers
U_PA_ON_2	Y21	DO	Third of three control bits for UMTS power amplifiers
G_PA_ON_0	W19	DO	First of two control bits for GSM power amplifiers
G_PA_ON_1	V19	DO	Second of two control bits for GSM power amplifiers
VOUT_PA_CTL	AA11	AO	Regulator output used to power PA control circuits via internal connections; connect 1.0 µF ceramic capacitor to ground.
Multipurpose pins			
MPP1	---	configurable	First of four MPPs; this one is connected internally
MPP2	W17	configurable	Second of four MPPs
MPP3	V17	configurable	Third of four MPPs
MPP4	V17	configurable	Fourth of four MPPs

[1] Pin type is AI = analog input, AO = analog output, DI = digital input, or DO = digital output.

16.2 User interfaces

In addition to housekeeping functions, the QSC device also supports these common handset-level user interfaces: current drivers, vibration motor driver, speaker driver, and one-touch headset control. All the relative QSC circuits are described in this section.

16.2.1 Current drivers

Two MPPs (Figure 16-1) can be used as high-current drivers. Both are independently programmable, ground-referenced current sinks with low-voltage compliance, making them suitable for many applications. Their intended applications and pin names are:

- MPP3 (KPD_DRV_N), pin W17, is intended to drive the keypad backlight.
- MPP4 (LCD_DRV_N), pin V17, is intended to drive the LCD backlight.

Since these high current drivers share pins with MPP functions, their external voltages are limited to VDD + 0.5 V; they cannot tolerate 5 V supplies.

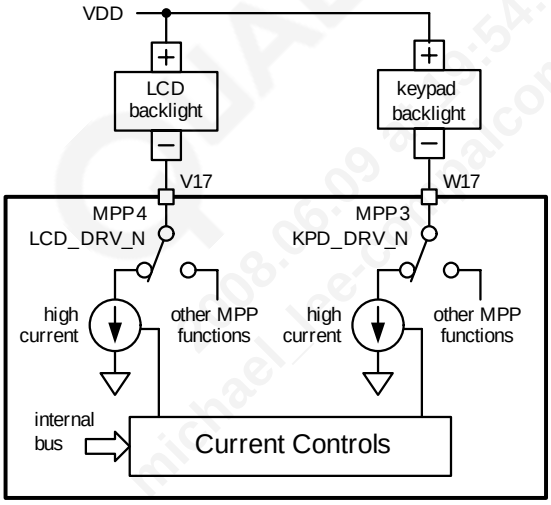


Figure 16-1 Current drivers

Since the drivers are ground-referenced current sinks, the devices they are driving must provide a current path from a high voltage to the appropriate driver output pin. The drivers' low-voltage compliance allows significant voltage drop across the device being driven. The two high-current drivers are capable of up to 150 mA each; they are essentially disabled when programmed for 0 mA.

16.2.2 Vibration motor driver

The QSC62x0 device supports silent incoming-call alarms with its vibration motor driver (Figure 16-2). The vibration driver is a programmable voltage output that is referenced to V_{DD}; when off, its output voltage is V_{DD}. The motor is connected between V_{DD} and pin U18 (VIB_DRV_N); the voltage across the motor is: $V_m = V_{DD} - V_{out}$ where V_{out} is the voltage at QSC pin U18.

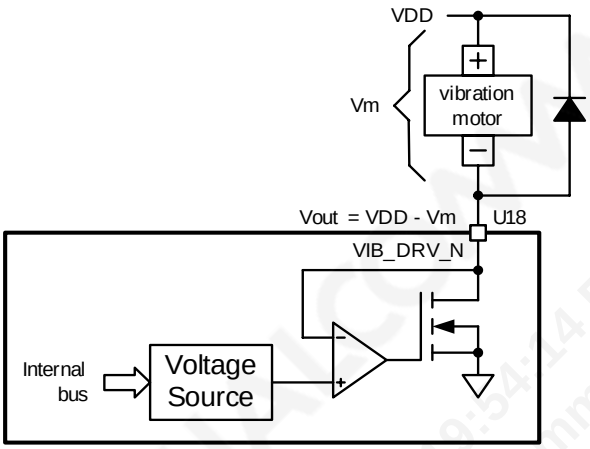


Figure 16-2 Vibration motor driver

A flyback diode is connected across the motor terminals to prevent inductive kickback during turnoff, thereby suppressing voltage transients that could damage the IC. Short-circuit current limiting is also provided to limit the current when the motor is stalled or shorted. The driver is programmable for motor voltages from 1.2 to 3.1 V in 100 mV increments.

The QSC device provides the option to control the vibration motor driver through an MPP (Figure 16-3). The following API software steps should be taken to control the vibration motor driver through an MPP. This technique provides greater flexibility in defining on and off vibration intervals.

1. Configure the MPP to be a digital input.
2. Define that MPP to be one of three DBUS signals.
3. Define the polarity of the control signal.
4. Define the vibration motor driver on/off control to the same DBUS signal.

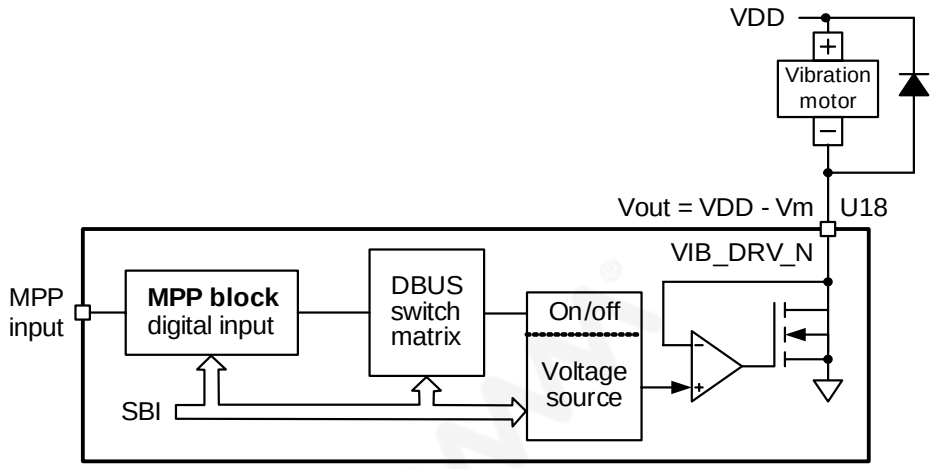


Figure 16-3 Using an MPP to control the vibration motor driver

16.2.3 Class-D speaker driver

The next user interface is the speaker driver circuit (Figure 16-4). This circuit supports audio outputs with its variable gain audio amplifier and class-D speaker driver. The output pins are configured differentially, with a rated output of 500 mW into an 8 Ω speaker. This drive capability can be increased to as much as 1 W by powering the driver circuit (VDD_SPKR) off the VREG_5V output (pin AC11).

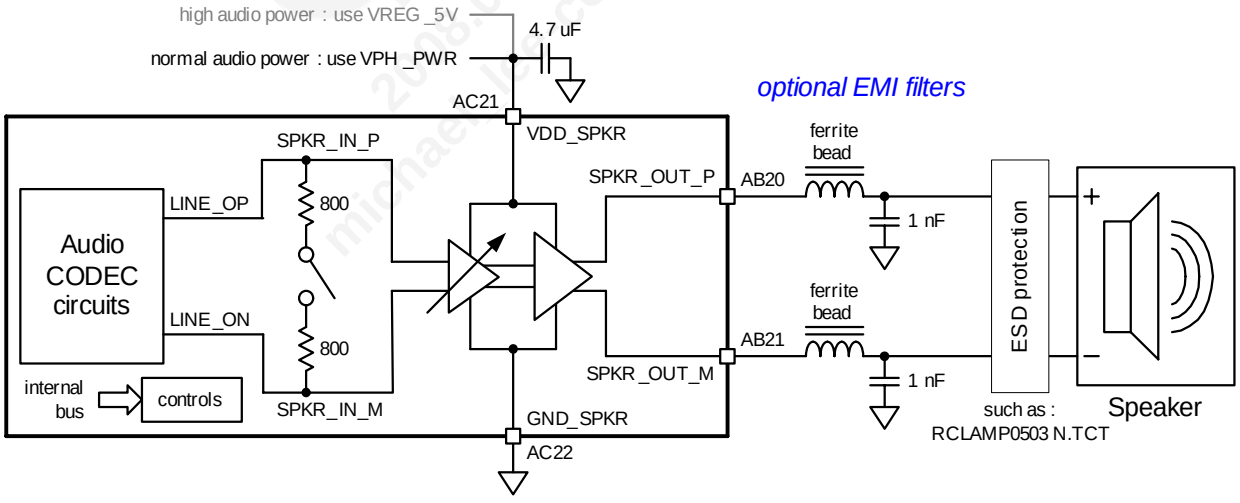


Figure 16-4 Speaker driver

Optional EMI filtering is shown at the QSC speaker driver output; these components (two ferrite beads and two capacitors) can be added to reduce electromagnetic interference. If used, they should be located near the QSC pins. Considerable current flows between the audio output pins and the speaker, so wide PCB traces are recommended (~ 20 mils).

Whether the EMI filters are used or not, the speaker terminals should include ESD protection (as shown).

Several speaker parameters are SBI-programmable:

- Gain – can be set from -16 to +12 dB in 2 dB increments.
- On/off – speaker circuits can be enabled or disabled.
- Mute – speaker output can be muted or not.
- Delay – a short (~10 msec) or long (~100 msec) delay is selectable.

The amplifier is well behaved during turnon, turnoff, mute/unmute, and gain-step transitions and does not produce any unusual or audible transients. It changes gain or mute conditions immediately when commanded, without applying envelope-shaping or zero-crossing delays. An algorithm performs envelope control or zero-crossing gain switching if completely inaudible switching is desired.

The QSC62x0 device supports speakerphones that are external to the handset (Figure 16-5). A handset connector is required to interface the QSC to the external microphone and loudspeaker devices. A detector automatically senses when the external microphone is connected and signals this condition to software through a GPIO. If the external microphone is not connected, the MIC1 and EAR1 I/Os are used and the internal microphone and speaker are active. If the external microphone is connected, it and the external loudspeaker are used in conjunction with the MIC2 and LINE I/Os. The speaker driver circuits are used to drive the external loudspeaker, exploiting its 500 mW capability (VPH_PWR at VDD_SPKR) or its 1 W capability (VREG_5V).

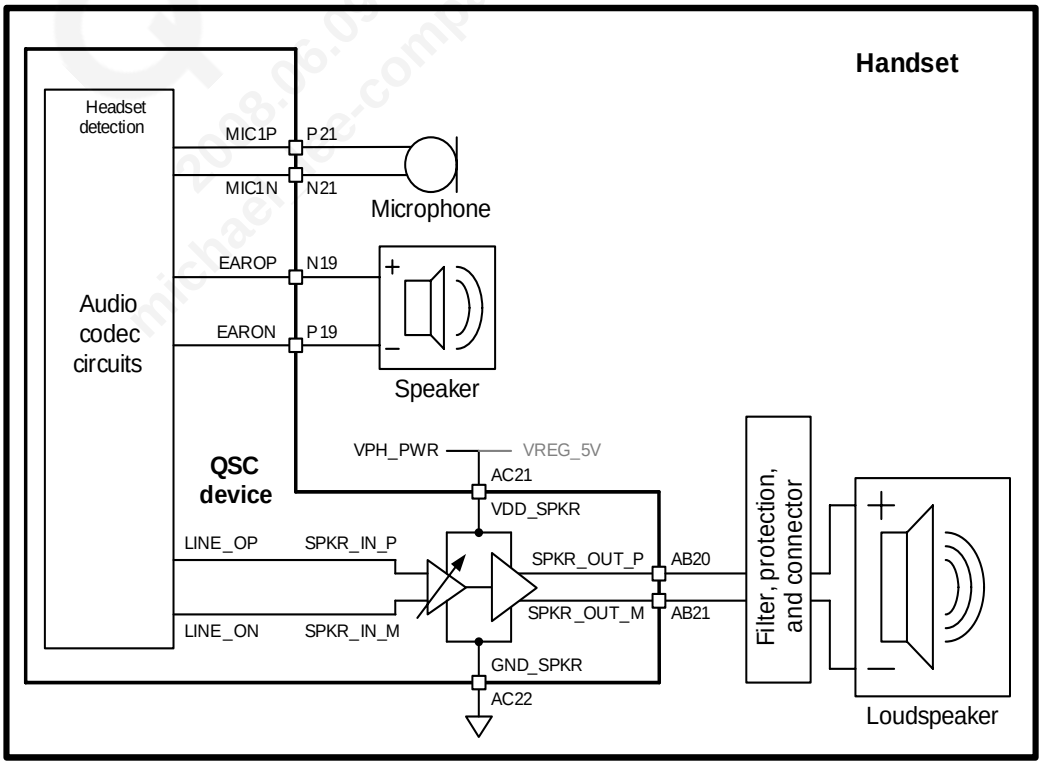


Figure 16-5 External speakerphone application

16.2.4 One-touch headset detection and MIC bias

The QSC62x0 device provides a current source for microphone biasing. The reference design (Figure 16-6) uses this on-chip source to bias the handset microphone and a headset microphone.

NOTE Because both the handset and headset microphones are biased from the same pin, HSED_BIAS, it is not possible to detect button presses from an open-circuit type headset. **Therefore, the QSC62x0 device only supports short-circuit type headsets.** If an open-circuit style needs to be supported, an additional MIC_BIAS source is required.

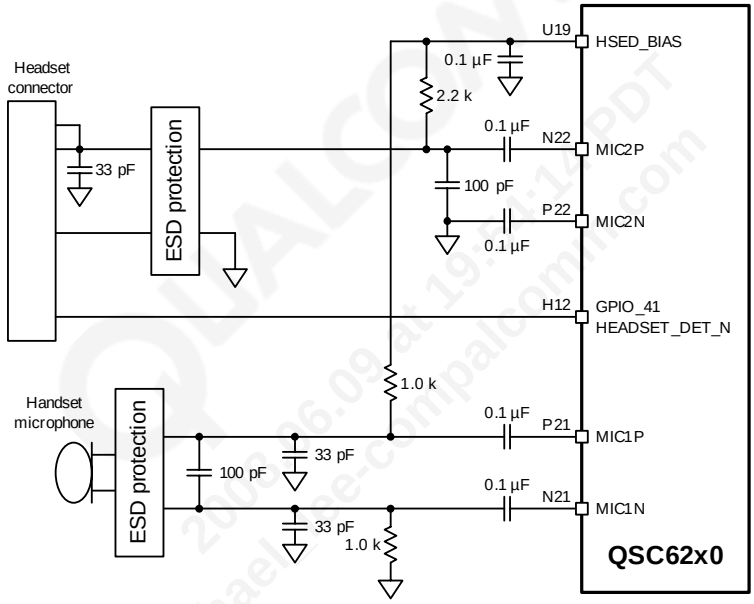
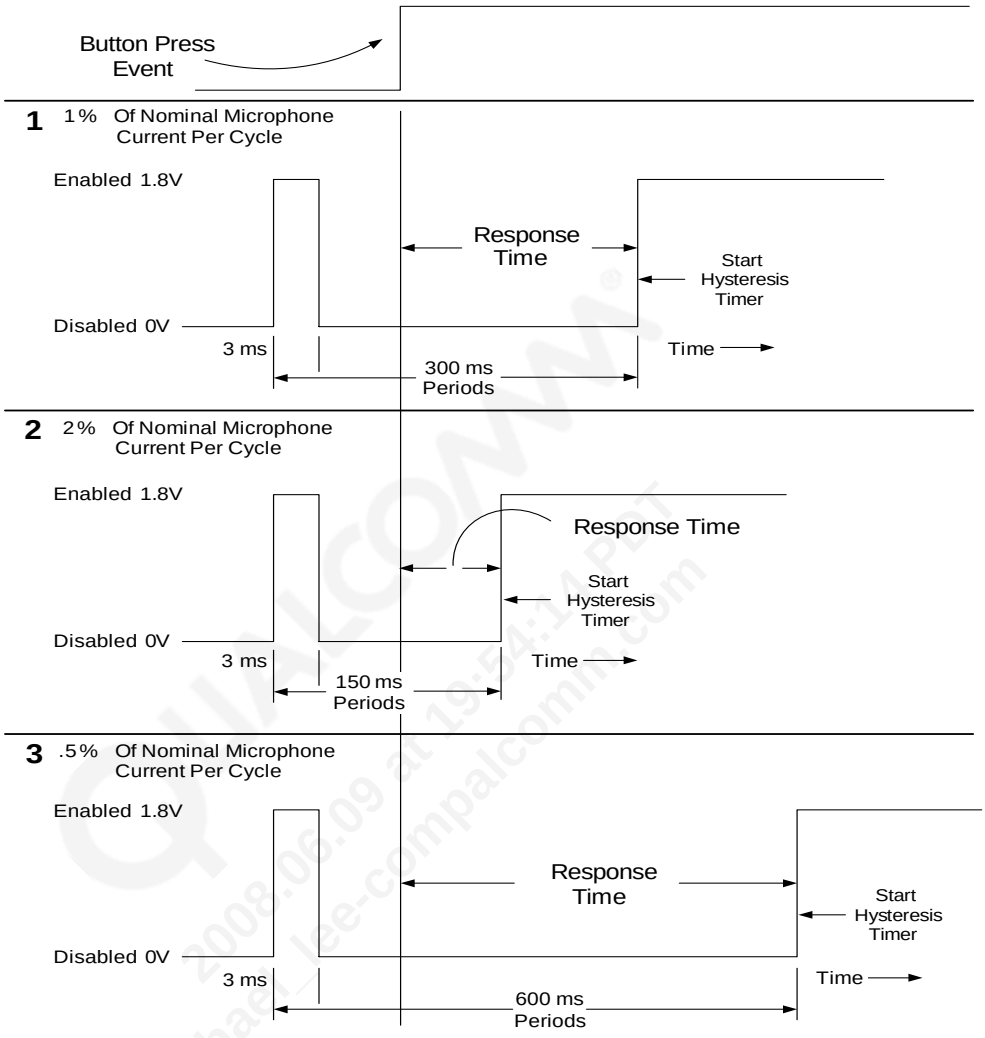


Figure 16-6 Example microphone biasing using HSED_BIAS

The microphone bias voltage is generated by a programmable pulse width modulator rather than a DC power source, which saves DC power when a microphone is connected but not being used. Three MIC bias waveform examples are shown in Figure 16-7.

When in a low-power mode (like when the end/send button not pressed), the MIC bias output is only enabled for 3 milliseconds at a time. The sampling interval defines the detection system's response time, and it is programmable from 1.9 milliseconds to 4 seconds. The average current consumed by the microphone is also controlled by this programmable interval; Figure 16-7 provides three examples:

- If the interval between each rising edge of the enable pulse is 300 milliseconds, the MIC bias signal is enabled 1% of the time (3/300). If the microphone draws a nominal bias current of 300 µA, the average current will be just 3 µA.
- If the interval is reduced from 300 to 150 milliseconds, the response time is quicker but the average microphone current increases to (3/150) x 300 µA = 6 µA.
- If the interval is increased from 300 to 600 milliseconds, the average microphone current drops to (3/600) x 300 µA = 1.5 µA but the response time is slower.



NOTE: A GPIO is required to provide the interrupt for the headset removal detection .

Figure 16-7 Programmable MIC bias pulse waveforms

GPIO_41 (pin H12), shown as HEADSET_DET_N in Figure 16-6, is required to detect headset insertion and removal. Future revisions of the device may allow HSED_BIAS to detect insertion and removal such that the HEADSET_DET_N GPIO is not required.

16.3 Power-on circuits and the power sequences

This section describes the QSC power-on circuits and presents an overview of the phone's power sequences. Subsections provide details of each mode within the power sequence.

Dedicated circuits continuously monitor four events that might trigger a power-on sequence. If any of the four events occurs, these circuits power on the QSC device, determine the handset's available power sources, enable the correct source, and take the device out of reset.

The inputs to the power-on circuit (KPD_PWR_N and PS_HOLD) are basic digital control signals, with the first two pins pulled-up internally. The only external output is PON_RESET_N, a basic digital output signal.

For applications that must power up immediately whenever a battery is inserted (such as a two-way pager), KPD_PWR_N may be tied to ground. In this case, the QSC device is always on and software should clear the KPDPWR_PU bit to turn off the internal pull-up resistor, thereby reducing quiescent current.

Discussion of the phone's power sequences begins with the simple, straightforward example that happens most often: a phone being powered from its battery that is turned on and off via the keypad (Figure 16-8). The power-on and power-off algorithms have to allow for alternate power sources, failed sequences, resets, and so on, and these will be addressed throughout this section, but the most common sequence serves as a good example with which to begin. Four distinct operating states occur:

- Power-on sequence
- On state
- Power-off sequence
- Off state

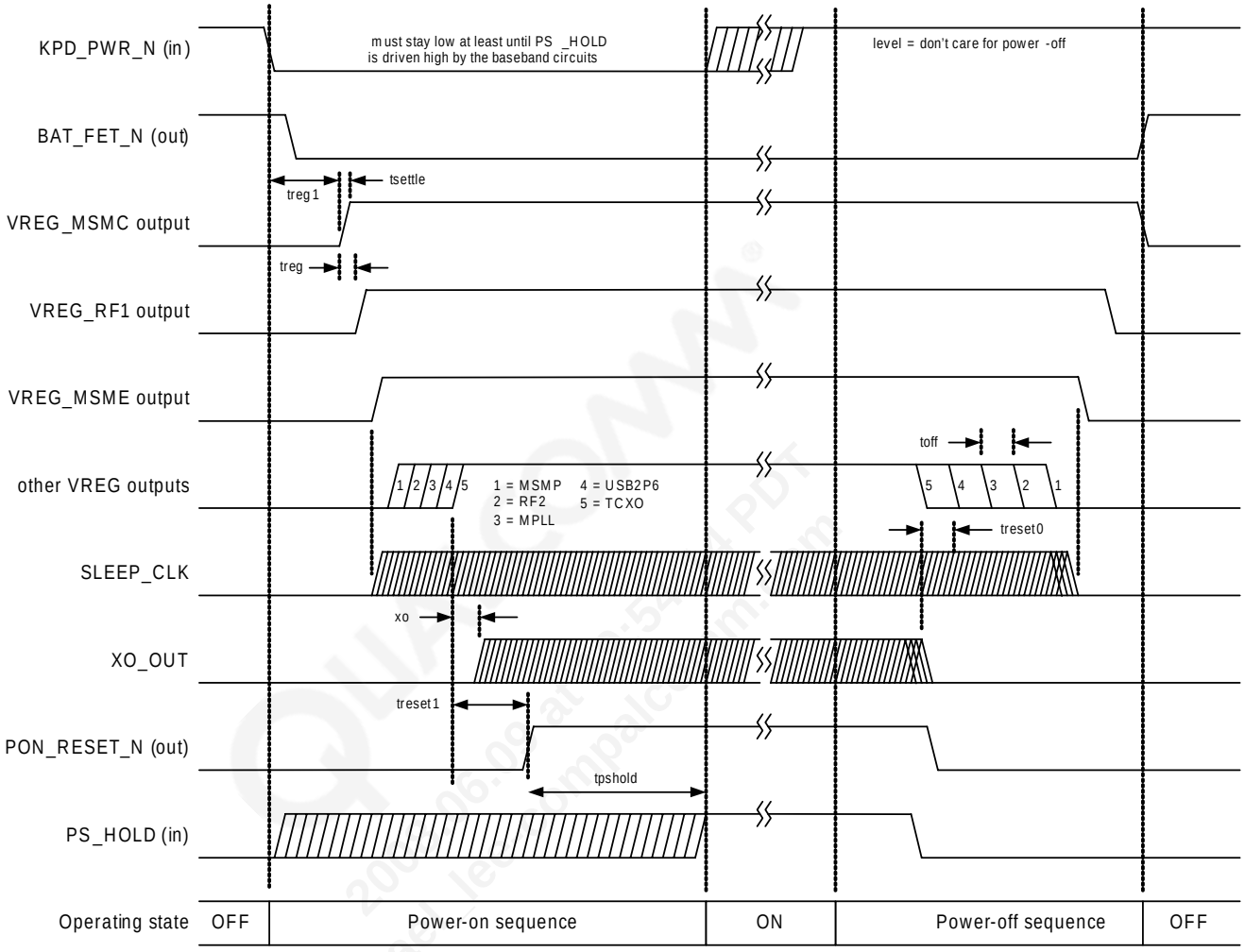


Figure 16-8 Example power sequences

This high-level power sequence diagram (Figure 16-8) tracks events on six QSC signals and key voltage regulator outputs during each of the four operating states. The power-on sequence is discussed first:

1. The power-on sequence begins when the keypad power button is pressed; this pulls the KPD_PWR_N pin low. For a successful power-on sequence, this signal must stay low at least until software drives the PS_HOLD signal high.
2. This example assumes the main battery is used to power the phone. The battery MOSFET is closed (the BAT_FET_N pin is driven to ground), connecting the battery to the phone's VDD node.
3. After a short delay (t_{reg1}) beyond the KPD_PWR_N transition default-on regulators are turned on in the following sequence:
   a. VREG_MSMC
   b. VREG_RF1
   c. VREG_MSME

d. VREG_MSMP

e. VREG_RF2

f. VREG_MPLL

g. VREG_USB_2P6

h. VREG_TCXO

4. As each regulator is enabled, detector circuits confirm that it powers up properly before triggering a wait interval. After the wait interval (t_{reg}) expires, the next regulator is enabled. This process continues until all the default-on regulators have powered up successfully.
5. As the MSME regulator turns on, so does the sleep clock oscillator and its output (SLEEP_CLK).
6. As the TCXO regulator turns on, the internal TCXO controller is enabled and provides the correct timing to ensure that the pulses at the internal XO_OUT node are synchronized and glitch-free. The delay from the TCXO regulator turnon until clean pulses are available at the XO_OUT node is t_{tcxo}.
7. After the clean XO_OUT pulses begin, another wait interval (t_{reset1}) is observed before the PON_RESET_N signal is driven high. Note that this signal is held low while the regulators are turned on to make sure SRAM is not corrupted. In fact, their chip-selects (active high) should be tied to PON_RESET_N. The t_{reset1} delay also makes sure the baseband circuits have time to power up and respond to the reset command correctly.
8. The PS_HOLD signal from the baseband circuits is allowed to be in any state during the early stages of the power-on sequence, but must be high when the t_{pshold} timer expires to keep the QSC device on. A completed, successful power-on sequence ends with the PS_HOLD signal driven high by the baseband circuits before the t_{pshold} timeout.

The QSC device is now in its on state. Note that if the PS_HOLD signal is low at the t_{pshold} timeout the power-on sequence has failed and the power-down sequence is started immediately.

The phone operates normally in its on state and PS_HOLD stays high to maintain the on state. If a power-on event occurs while the device is already on, the appropriate interrupt is generated but the phone continues operating normally.

Next, consider the powerdown sequence:

1. The power-off sequence begins when the baseline circuits drive the PS_HOLD pin low, requesting the PM circuits to orchestrate the powerdown sequence.
2. PON_RESET_N is driven low to reset the on-chip baseband circuits and other external devices, and disables the VREG_TCXO regulator and XO_OUT output.

3. After the wait interval t_{reset0} the remaining regulators are turned off in the following sequence:
   a. All regulators that do not default on, and then the default-on regulators (if on) as listed below
   b. VREG_USB_2P6
   c. VREG_MPLL
   d. VREG_RF2
   e. VREG_MSMP
   f. VREG_MSME
   g. VREG_RF1
   h. VREG_MSMC

   An interval of t_{off} is allowed for the each regulator output to discharge before the next regulator is turned off.
4. The battery is disconnected from the phone's primary power node (V_{DD}) by driving the BAT_FET_N signal high.

The QSC device is now in its off state and remains off until the next power-on event occurs.

This discussion provided a high-level overview of the power sequences; additional details of the basic sequences and potential complexities are presented in the following subsections.

16.3.1 Power-on sequence

While the PS_HOLD signal is low, the QSC device is in one of its off states. Under this condition, the power-on circuits continually monitor four events that could trigger a power-on sequence:

1. The keypad power-on button is pressed and the KPD_PWR_N signal is pulled low.
2. An external supply source is detected (the voltage on the VCHG pin exceeds its valid threshold).
3. The real-time clock alarm is triggered.
4. The sudden momentary power loss (SMPL) condition was detected and an SMPL recovery is initiated.

A valid supply on the USB_VBUS pin will not cause a powerup event because USB_VBUS is not part of the charger circuit. The exception is if USB charging is enabled (as discussed in Section 13.9), in which case both USB_VBUS and VCHG interrupts are triggered when a valid USB supply is attached.

Each of these four events is described later in this section; each generates a start signal and a unique interrupt informing software what caused the power to turn on. The interrupts are sent even if the QSC device was already on when the event occurred. Regardless of the triggering event, the QSC device executes the following power-on sequence:

1. One of four triggering events has occurred or continues to occur.
2. Validate and select the appropriate power source:
   a. If the voltage on the VCHG pin exceeds its valid threshold, the external supply is selected and the QSC voltage regulation circuits are enabled.
   b. If voltage on the VCHG pin is lower than its valid threshold, the battery is selected by driving the BAT_FET_N signal low.
3. The internal bandgap voltage reference is enabled and allowed to stabilize.
4. The rising V_{DD} voltage is monitored; the power-on sequence continues after the undervoltage lockout threshold is exceeded.
5. The PON_RESET_N signal is driven low even though it may not propagate yet because the MSMP regulator is not on. This assures the correct PON_RESET_N state as the baseband circuits power up.
6. Internal registers are set to their default values. If the voltage at VCOIN is below its valid threshold, the register bits backed up by the coin cell are reset and the RTCRST interrupt is set.
7. If the SMPL function is enabled and the SMPL restart signal is true, the SMPL interrupt status is set and the SMPL timer is reset (preparing it for another SMPL event if enabled).
8. Following a short delay (t_{reg1}) after the power-on event occurred, the voltage regulators that default on are enabled in the following sequence:
   a. VREG_MSMC
   b. VREG_RF1
   c. VREG_MSME
   d. VREG_MSMP
   e. VREG_RF2
   f. VREG_MPLL
   g. VREG_USB_2P6
   h. VREG_TCXO

   As each regulator is enabled, detector circuits confirm that it powers up properly before triggering a wait interval. After the wait interval (t_{reg}) expires, the next regulator is enabled. This process continues until all the regulators that default on have powered up successfully.
9. As the MSME regulator turns on, so does the sleep clock oscillator and its output (SLEEP_CLK).

10. As the TCXO regulator turns on, the internal TCXO controller is enabled and provides the correct timing to ensure that the pulses at the internal XO_OUT node are synchronized and glitch-free. The delay from the TCXO regulator turnon until clean pulses are available at the XO_OUT node is t_{tcxo}.
11. After the clean XO_OUT pulses begin, another wait interval (t_{reset1}) is observed before the PON_RESET_N signal is driven high. Note that this signal is held low while the regulators are turned on to make sure SRAM is not corrupted. In fact, their chip-selects (active high) should be tied to PON_RESET_N. The t_{reset1} delay also makes sure the baseband circuits have time to power up and respond to the reset command correctly.
12. The PS_HOLD timer is started and the PS_HOLD signal from the baseband circuits is monitored continuously; successful completion of the power-on sequence requires that it transitions from low to high before the PS_HOLD timer expires.
    a. If the timer expires without a PS_HOLD low to high transition, the QSC device returns to its off state and the power-on sequence returns to step 1. If a triggering event is still present (such as KPD_PWR_N still low), the power-on sequence repeats.
    b. If PS_HOLD goes high before the timer expires, the QSC device is latched in one of its on states (sleep or active) and the appropriate interrupts are made available to software so it can determine the triggering event. The QSC device is now ON.

A couple of events, if they occur during the power-on sequence, will prevent the successful completion of the sequence.

- If the start signal is dropped, the sequence stops and returns to step 1.
- If V_{DD} collapses below the UVLO threshold (nominally 2.55 V) before PS_HOLD is driven high, the device will abort the power-on sequence and will attempt the sequence two more times. After the third aborted attempt, it will reset and wait for another triggering event (start signal edge). This can occur when the external supply current is much too low and a battery is not present to assist the start up.

Note that if V_{DD} collapses below the UVLO threshold after the PS_HOLD signal has been driven high, then the restart circuit is not activated. Instead, this event is handled as an SMPL event (see Section 13.13 for details).

Once a successful power-on sequence is completed, the QSC device stays on until PS_HOLD is forced low by software, an undervoltage lockout condition occurs, or an overtemperature thermal shutdown event occurs. On any of these events, the power-off sequence is executed (Section 16.3.3).

Details of the four power-on triggering events are as follows:

- Keypad power-on button is pressed and the KPD_PWR_N signal is pulled low.

  The QSC pin AA10, KPD_PWR_N, is dedicated to monitoring the handset's keypad power button. This pin is pulled up internally with a 200 k resistor and should not be driven high externally. When the keypad button is pressed, its switch is closed and the power-on sequence is initiated. The keypad switch must remain closed (the button must remain pressed) until software forces the PS_HOLD signal high; otherwise the QSC device will turn itself off.

- External supply source is detected — the voltage on VCHG (pins AA14 and AB14) exceeds its valid threshold and the power-on sequence is initiated.

  If a valid source is applied, V_{DD} will rise above the UVLO threshold, the minimum voltage required for QSC operation. When the entire power-on sequence is successful, the appropriate interrupt is generated. Registers are read to identify the power source and whether battery charging is required. If a valid supply voltage is applied while the QSC device is already on, the power-on sequence is skipped but the interrupts are still generated.

- Real-time clock alarm is triggered.

  While the QSC device is off, the real-time clock (RTC) and its oscillator source are still active, provided a coin cell battery is installed. This allows continued monitoring of RTC alarms programmed via software. If an alarm occurs while the device is off an alarm interrupt is generated and the power-on sequence is initiated. See Section 15.4 for a complete discussion of the real-time clock circuits and functions.

- Sudden momentary power loss (SMPL) condition was detected and an SMPL recovery is initiated.

  If the power-on circuits detect that the SMPL function is enabled when the PS_HOLD signal is cleared, the power-on sequence is initiated. This can be used for soft resets as described in Section 16.3.5. For more information about the SMPL feature, see Section 13.13. Note that sufficient voltage at the VCOIN pin is required for SMPL operation. The SMPL timer is reset whenever the QSC device is turned on, making the SMPL feature available for multiple, successive SMPL events.

16.3.2 Power-on state

After the power-on sequence is finished, the phone operates normally in its on state and PS_HOLD stays high to maintain the on state. If a power-on event occurs while the QSC device is already on, the appropriate interrupt is generated but the phone continues operating normally. See Section 16.4 for more details about the interrupts.

16.3.3 Power-off sequence

While the PS_HOLD signal is high, the QSC device is in one of its on states. Under this condition, the device continually monitors three events that could trigger a power-off sequence:

1. Software drives the PS_HOLD signal low in response to the handset user pressing the keypad power button.
2. The detected V_{DD} voltage measured at VPH_PWR (pins AA13 and AB13) drops below the UVLO threshold.
3. The QSC die temperature exceeds the overtemperature threshold.

The most common power-off sequence begins with the keypad power button being pressed while the QSC device is powered on and operational. As discussed earlier, the KPD_PWR_N pin is connected to the handset keypad power button and is pulled up internally. This button may be used for multiple functions. For example, many handsets label this button END and use it to end calls and to turn the handset on and off.

The QSC monitors this button through interrupt logic. When the KPD_PWR_N interrupt is triggered it initiates the following power-down sequence:

1. The user presses and holds the END key for at least the minimum interval (t_{end}). The power-down sequence is aborted if the END key is not pressed long enough.
2. A message such as “Powering off …” is displayed.
3. Any information that needs to be saved is written to flash ROM.
4. The SMPL feature is disabled to prevent the QSC device from powering up again.
5. The power-off sequence waits until the user releases the END key; otherwise the QSC device will power up again.
6. Software drives the PS_HOLD signal low to cause the QSC device to power down the handset.
7. The power management circuits drive PON_RESET_N low to reset the on-chip baseband circuits and other external devices, and disables the TCXO regulator and XO_OUT output.
8. After a t_{reset0} interval, several conditions are checked to determine the next action:
   a. If the overtemperature threshold was exceeded, then the PON_RESET_N signal is immediately driven low and all QSC circuits are turned off. The overtemperature feature protects the QSC device and cannot be disabled.
   b. If the primary phone voltage V_{DD} is below the UVLO threshold, then the PON_RESET_N signal is immediately driven low and all QSC circuits are turned off. The UVLO feature protects the battery and cannot be disabled.
   c. If the overtemperature threshold was not exceeded, the primary phone voltage V_{DD} is above the UVLO threshold, and the watchdog restart bit is set, then the watchdog interrupt status bit is set, the QSC device is restarted without fully powering down, and its watchdog timer is reset.
   d. If the overtemperature threshold was not exceeded, the primary phone voltage V_{DD} is above the UVLO threshold, and the SMPL function is enabled, then the SMPL recovery is executed as described in Section 16.3.5.
   e. If none of the above combinations occur, then the normal powerdown sequence is continued.

9. The remaining regulators are turned off in the following sequence:
   a. All regulators that do not default on, and then those regulators that default on (if on) as listed below
   b. VREG_USB_2P6
   c. VREG_MPLL
   d. VREG_RF2
   e. VREG_MSMP
   f. VREG_MSME
   g. VREG_RF1
   h. VREG_MSMC

   An interval of t_{off} is allowed for the each regulator output to discharge before the next regulator is turned off.
10. Another interval passes (t_{reg}), and then the bandgap reference and all other unneeded circuits are disabled.
11. The QSC device turns off the battery MOSFET by driving BAT_FET_N high and makes sure the external supplies are disconnected by driving the charging pass transistor's control signal high. With all sources disconnected, the V_{DD} has no applied power and the phone is powered down.

The QSC device is now off.

16.3.4 Power-off state

Once the power-off sequence is finished, the phone stays in its off state until a power-on sequence is triggered.

When the QSC device is off, it might be exposed to a wide range of voltage conditions at the VPH_PWR pins (the V_{DD} voltage). If a battery MOSFET is not used, V_{DD} might be as high as 4.2 V. If a battery MOSFET is used, V_{DD} might be a diode drop below the battery voltage. If the battery is removed, V_{DD} might be 0 V. Regardless of the V_{DD} voltage the following conditions must be met during the off state:

- There must not be any leakage between V_{DD} and the coin cell (VCOIN, pin AA18).
- The battery MOSFET must be off – BAT_FET_N is pulled high (to within 0.1 V of V_{DD} or V_{BAT}, whichever is higher).
- The charging pass transistor must be off – its control signal is pulled high (to within 0.1 V of V_{DD} or V_{CHG}, whichever is higher).

16.3.5 Watchdog timeout and software reset

The QSC device has a watchdog timer to reset itself if the firmware locks up. The software reset is executed as follows:

- When the watchdog timer expires, software drives PS_HOLD low.
- The power management circuits respond by immediately driving PON_RESET_N low to reset the on-chip baseband circuits and other external devices.
- After a t_{reset0} interval, several conditions are checked; the result in this case is to realize that a watchdog timeout has occurred and a software reset is being executed.
- A soft reset requires that the SMPL feature is enabled; if so, the QSC device does not power down. Instead it resets registers to default values, waits 20 milliseconds, and then drives PON_RESET_N high to take the QSC device out of reset.
- The QSC device does not turn the regulators off and on when SMPL occurs or during a soft reset.

If SMPL is used and a normal powerdown is desired, software must disable the SMPL function before deasserting PS_HOLD to avoid an inadvertent SMPL override. Without disabling SMPL, the power management circuits will detect and defeat the intended powerdown, instead executing a soft reset. The differentiating factor between SMPL and watchdog timeout is the UVLO detector state. If a UVLO did not occur (V_{DD} stayed above the UVLO threshold), the powerup is a watchdog reset. If a UVLO did occur, the powerup is an SMPL event. Either event triggers its respective interrupt, allowing software to learn what caused the powerup.

16.4 PM interrrupt manager

An interrupt manager receives internal reports on numerous PM functions and conveys real time and latched status signals to software, thereby supporting the baseband circuit's interrupt processing.

Each interrupt event has the following associated register bits:

- Interrupt mask (read/write) – When set, this bit allows the baseband circuits to ignore the interrupting event. Setting the mask bit prevents the PM circuits from generating a latched status and corresponding interrupt.
- Interrupt real-time status (read only) – This bit follows the real-time interrupt event status (active or inactive).
- Interrupt latched status (read only) – This bit is set when the interrupt event is active and the interrupt mask bit is cleared. It stays set until the interrupt clear bit is set. This bit can only be read while the interrupt mask bit is cleared.
- Interrupt clear (read/write) – Setting this bit clears the interrupt event's latched status. It is cleared automatically after the latched status is read, which has no effect other than allowing it to be set later to again clear the event's latched status.

When one or more interrupt events occur that are not masked, the PM interrupt signal notifies software that at least one interrupt has occurred. This signal stays active until the corresponding interrupts are cleared using the appropriate clear bits. When all interrupts have been cleared (or masked) by baseband circuits the interrupt signal returns to its inactive state.

Of the many functions monitored for interrupt events, most are standard interrupts and four are special interrupts (Figure 16-9). The special interrupts are OVERTEMP, RTCRST, SMPL, and WDOG. Special interrupts are not cleared by a QSC reset, do not provide truly real-time status bits, and provide latched versions of the real-time status bits whether the mask bit is set or not. These special characteristics are necessary because the associated interrupts often occur just before or during a QSC reset. The device latches the interrupt just before resetting and these particular interrupts are backed up by the coin cell. When the device restarts, the latched interrupts inform software that they were triggered. Note that the transparent latch holds the interrupt event during the read strobe even if it transitions midway through the read.

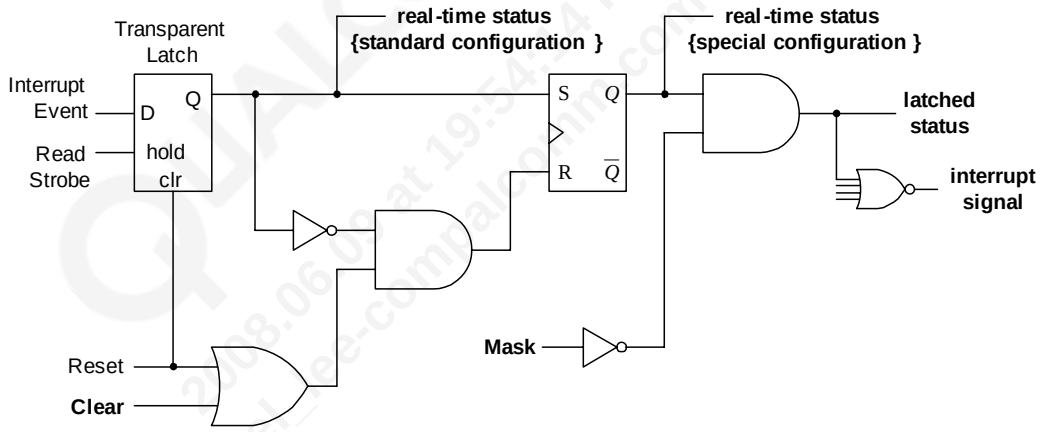


Figure 16-9 PM interrupt functional block diagram

On powerup, software should check the RTCRST interrupt; if set, this indicates that the coin cell voltage is too low and that the RTC contents and SMPL and WDOG interrupts are unreliable.

Interrupt timing is illustrated in Figure 16-10.

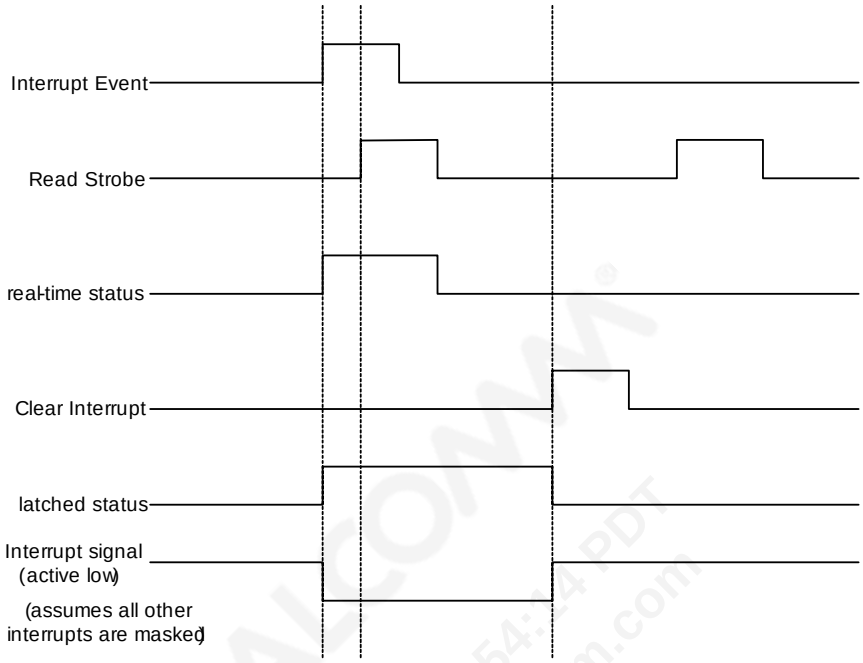


Figure 16-10 PM interrupt timing

A summary of the interrupt functions is presented in Table 16-2.

Table 16-2 PM interrupt events (listed alphabetically)

Name	Description	Trigger condition/consequence
BATFET	Battery MOSFET was closed	BAT_FET_N = 0 through software programming or automatically by the QSC device due to $V_{DD} < V_{BAT}$.
BDACFE	TBD	TBD
BDACRE	USB B disconnect, A connect	A USB B-device was disconnected (bus has been inactive), the QSC device turned off its D+ pull-up, and the USB B-device turned on its D+ pull-up indicating that the B-device is now ready to respond as a peripheral.
CHGILIM	Charger in current limiting	Sensed current has exceeded its limit (IMAXSEL) and the charger loop has started to limit current. Note that this interrupt may trip when enabling fast charge, depending upon how quickly the external charger drops its voltage.
CHGINVAL	Charger became invalid	External supply voltage (V_{CHG}) went outside its valid range.
CHGPLIM	Charger in power limiting	Sensed charging pass transistor power has exceeded its limit (PMAXSEL) and the charger loop has started to reduce current. This interrupt may trip when enabling fast charge, depending upon how quickly the external charger drops its voltage.
CHGVAL	Charger became valid	External supply voltage (V_{CHG}) came inside its valid range.
DMHI	USB D- high	The USB D- signal went high.
DPHI	USB D+ high	The USB D+ signal went high.
DPSRP	USB D+ SRP	A session request was no longer detected (both data lines went low).

Table 16-2 PM interrupt events (listed alphabetically (continued))

Name	Description	Trigger condition/consequence
IDFLTFE	USB ID not floating	The USB ID pin was floating then became shorted to ground (a mini-A plug was attached) or became resistive (an RS-232 dongle was attached).
IDFLTRE	USB ID floating	The USB ID pin was floating, indicating that a mini-A plug was removed or a mini-B plug was attached.
IDGNDFE	USB ID not grounded	The USB ID pin was not grounded anymore.
IDGNDRE	USB ID grounded	The USB ID pin was grounded.
IDRES	USB ID resistive	The resistor was disconnected from the ID line, indicating an RS-232 dongle was disconnected.
KPDPWRON	Power button has been pressed	KPD_PWR_N pin was pulled to ground for longer than the KPD_DLY timer setting.
KBDPWROFF	Power button has been released	KPD_PWR_N pin was floating for longer than the KPD_DLY timer setting.
MPPxx	State of MPP configured as digital input has changed	Each of the four MPPs is monitored for state changes when it is configured as a digital input.
OSCHALT	32.768 kHz crystal oscillator has stopped (halted)	Crystal oscillation detector tripped; the system clock is switched to the internal RC oscillator.
OVERTEMP	Die is overtemperature	Die temperature exceeds temperature thresholds; the device is immediately powered down.
PCTDONE	Pulse charging is done	The number of consecutive TOFF cycles equaled the TDONE register setting.
RTCALRM	RTC alarm	The RTC time equals the RTC alarm setting.
RTCRST	RTC contents may be corrupted	The QSC device has lost all power, including coin cell keep-alive, therefore the RTC contents are no longer valid. Even if the voltage is restored and the RTC starts to run, this interrupt status is retained as a warning flag to the baseband circuits.
SESEND	USB session end	The USB session has ended.
SESVLDFE	USB session invalid	V_{BUS} went below its session valid threshold.
SESVLDRE	USB session valid	V_{BUS} went above its session valid threshold.
SMPL	Phone powered up due to an SMPL event	The battery disconnected then immediately reconnected; V_{DD} went below the UVLO threshold, the QSC device was powered down, then the threshold was exceeded within the SMPL timeout causing an automatic restart.
TEMPSTAT	Temperature status change	The QSC die has exceeded one of its temperature thresholds.
VBAT	Battery became invalid	Battery voltage (V_{BAT}) has stayed outside its programmed valid range of VB_LOWER to VB_UPPER for longer than the programmed delay VB_INTDLY.
VBATDET	V_{BAT} reached the top-off threshold	The voltage at the VBAT pin (V_{BAT}) met or exceeded the programmed VBATDET value.
VBSVLDFE	USB VBUS invalid	The phone is configured as a USB A-device and V_{BUS} went below its valid threshold.
VBSVLDRE	USB VBUS valid	The phone is configured as a USB A-device and V_{BUS} went above its valid threshold.

Table 16-2 PM interrupt events (listed alphabetically (continued))

Name	Description	Trigger condition/consequence
VCP	V_{DD} collapse protection was triggered	V_{DD} went below V_{BAT} so the battery FET was turned on to prevent a shutdown.
VIDLD	Video amp loaded	A video amplifier load attachment was detected.
VIDUNLD	Video amp unloaded	A video amplifier load removal was detected (unloaded).
WDOG	Watchdog timer expired	PS_HOLD was driven low by the baseband circuits without V_{DD} going below UVLO and without the watchdog timer being disabled.

16.5 PA controller

The QSC62x0 PA controller is implemented within the PM block, but operates under the direction of the baseband circuits. A dedicated BB-PM high-speed interface supplements the internal bus to configure the PA control signals. Configurable parameters include the output logic levels (high or low), polarity, output channel selection, and output driver supply voltage. Figure 16-11 shows a tri-band UMTS, quad-band GSM example.

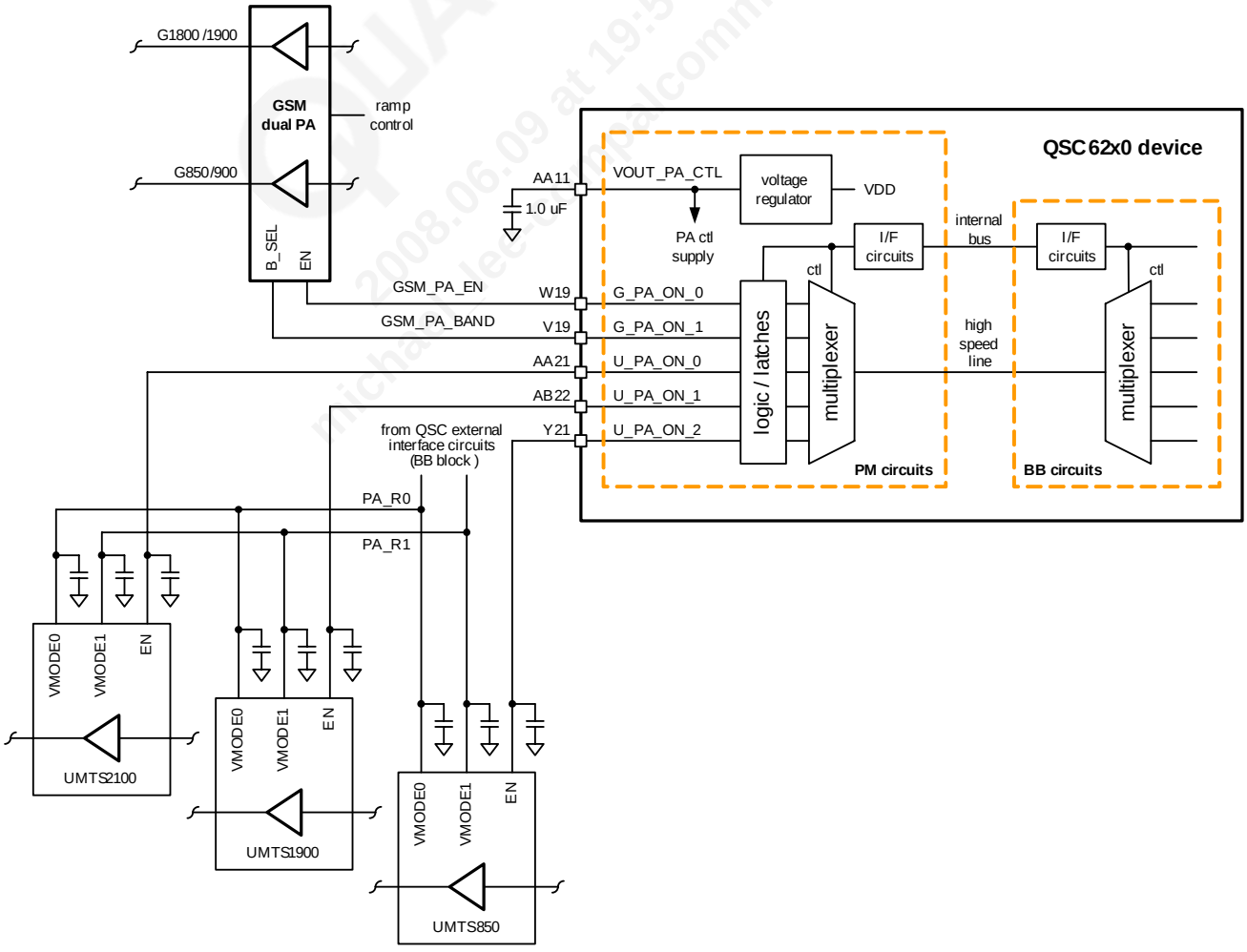


Figure 16-11 PA controller functional block diagram and connections

Three output control pins are intended for enabling the UMTS power amplifiers. Each is connected directly to its PA module's enable input, with a filter capacitor near the PA pin. The expected logic for each pin is: high --> on; low --> off.

The two GSM output control signals are intended for controlling a quad-band, dual-PA GSM module. They are direct connections (filtering should not be required), and their expected functions and logic are:

- Pin W19, G_PA_ON_0 = GSM_PA_EN
  - High --> on; low --> off
- Pin V19, G_PA_ON_1 = GSM_PA_BAND
  - High --> GSM1800/1900; low --> GSM850/900

16.6 Multipurpose pins

This section addresses the four QSC MPPs, their configuration options, and their programmable characteristics.

NOTE MPP3 and MPP4 (pins W17 and V17) provide dual functionality:

- They can be configurable MPPs as defined below.
- They can be high-current drivers, suitable for driving the keypad and LCD backlights.

See Section 16.2.1 for details regarding their high-current driver configuration.

The MPP configurations applicable to all four pins are discussed in this section.

16.6.1 MPP configuration options

The QSC62x0 device includes four MPPs (Figure 16-12) having the following software-configurable options:

- Digital input – Digital inputs applied to the pin can be read via software, can trigger an interrupt, or can be routed to another MPP (making this pin the input side of a level translator or current sink controller). The logic level is programmable, providing compliance between I/Os running off different power supplies.
- Digital output – The output signal can be set via software to logic LOW or HIGH, can come from this pin's complementary MPP (making this pin the output side of a level translator), or can be tri-stated for use as a switch. The logic level is programmable, providing compliance between I/Os running off different power supplies.
- Analog input – Inputs are routed to the analog multiplexer switch network; if selected that analog voltage is routed to the HKADC for digitization.
- Analog output – A buffered version of the on-chip voltage reference (V_{REF}) is provided as an analog output.

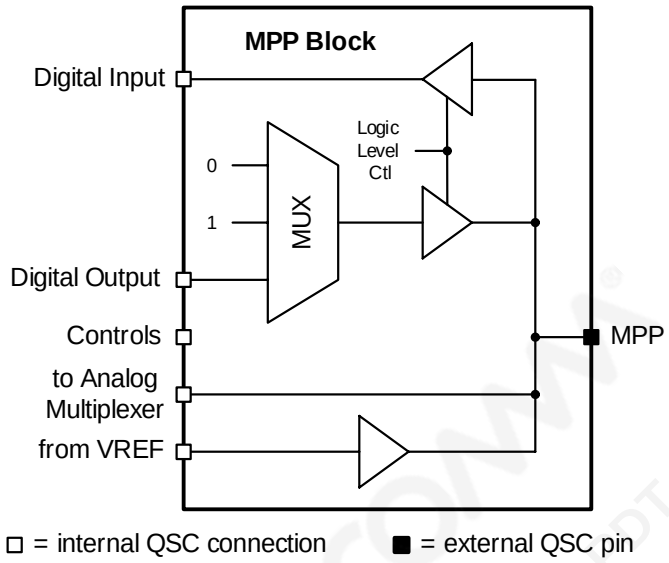


Figure 16-12 MPP pin functional block diagram

16.6.2 Programmable MPP characteristics

Depending on which one of the four possible configurations is selected, additional MPP characteristics (Table 16-3) are also programmed via software.

Table 16-3 Programmable MPP characteristics

MPP type	Programmable characteristics	Valid values
Digital input	Input logic level	Derived from the selected power supply: VREG_MSME, VREG_MSMP, VREG_USIM, or V_{DD}
Digital output	Output status Selects the controlling input Output logic level	Enabled or disabled (tri-stated) Internal bus or the paired MPP input status (inverting or noninverting) Derived from the selected power supply: VREG_MSME, VREG_MSMP, VREG_USIM, or V_{DD}
Analog input	Selects an analog multiplexer input channel	Channel 5, 6, 7, 8, or 9
Analog output	Selects the buffered version of V_{REF}	$V_{out} = V_{REF}$

17 Power and Ground

Most power supply voltages for QSC62x0 circuits are generated and controlled by internal power management (PM) functions. Off-chip power sources, such as a wall charger or battery, provide the raw voltage source from which the power management functions generate the needed supplies for the entire handset (including other internal QSC circuits). This chapter details methods for connecting the off-chip power sources to the QSC power management functions, and for connecting the generated supply voltages to other QSC circuits. QSC62x0 grounding recommendations are also provided.

17.1 QSC input supply voltage from off-chip sources

Several pins are used to apply the raw phone power supply input voltage to internal QSC62x0 circuitry (Figure 17-1). This raw supply voltage is connected to the QSC device's VPH_PWR pins (AA13 and AB13); it is the node usually referred to as VPH_PWR or VDD. The QSC's operating supply current can be sourced by the external supply via the on-chip charging pass transistor through the VPH_PWR pins, or it can be sourced from the main battery.

A handset-external supply (such as a wall charger at VCHG) or the main battery (VBAT) is used as the power source. The raw supply is regulated (if needed) by the PM function's input power-management circuits to establish the desired VPH_PWR voltage. VPH_PWR is then physically routed in two different directions: 1) outwardly to the power amplifier, and 2) back into the QSC device for powering the voltage regulators and other internal circuits.

All raw input power pins are powered by the same power source, but each has its own routing path and bypass capacitors. Each pin has at least a dedicated 0.1 µF bypass capacitor nearby. Switching regulator input supply pins (AA19 and AC16 – supplies for the MSMC, NCP, RF1, and RF2 regulators) are allocated a 2.2 µF capacitor. The VDD_SPKR supply pin (AC21) is included since it will often be powered off VPH_PWR. Zero-Ohm resistors are shown that allow this input pin to be connected to either VPH_PWR for normal audio output power levels (500 mW typical), or to VREG_5V for high audio power applications (up to 1 W).

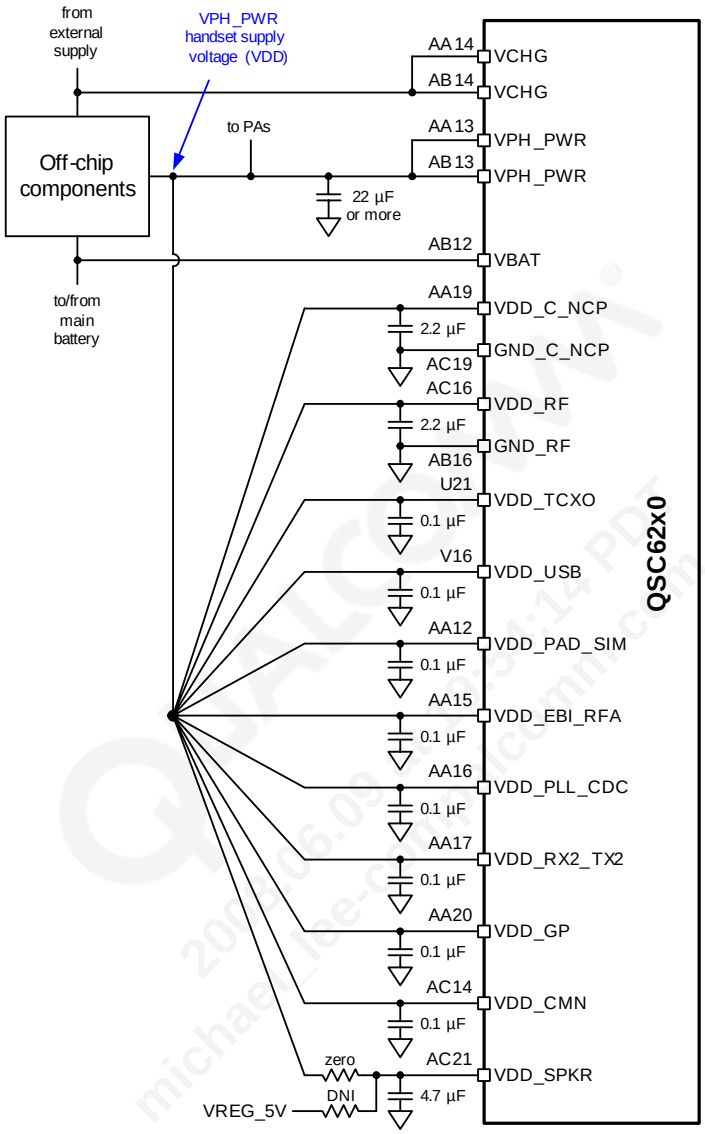


Figure 17-1 Input connections from an off-chip power source

17.2 QSC power-supply interconnections

On-chip circuits regulate the raw supply voltage into all the voltages needed by other QSC circuits plus the voltages needed by other handset circuits. All QSC I/Os associated with the regulator circuits are summarized in Table 17-1. Access to key nodes provides the option to subregulate several key voltages; the recommended configuration includes subregulation of some voltages (as discussed in Section 14.6 and highlighted within Table 17-1).

Table 17-1 QSC on-chip regulator connections

Pin name/function	Pin #	Pin type [1]	Functional description
Positive SMPS connections			
VSW_5V	AC12	AO	The switching output of the +5-V boost (step-up) SMPS circuit; connect to the primary phone power through a 2.2-µH inductor, and also connect to the Schottky diode's anode pin.
VREG_5V	AC11	AI	This senses the regulated output of the +5-V boost SMPS; connect to the Schottky diode's cathode pin. Bypass this pin with a 10-µF ceramic capacitor and route to the desired load circuits.
VDD_C_NCP	AA19	P	Input power to the MSMC buck SMPS and NCP negative charge pump circuits; bypass with a 2.2-µF ceramic capacitor.
VSW_MSMC [2]	AC18	AO	The switching output of the core SMPS circuit; connect to the core regulator's 2.2-µH buck inductor, input side.
VDD_RF	AC16	P	Input power to the two RF SMPS (RF1 and RF2) circuits; bypass with a 2.2-µF ceramic capacitor.
VSW_RF1	AC15	AO	The switching output of the RF1 SMPS circuit; connect to the RF1 regulator's 2.2-µH buck inductor, input side.
VREG_RF1	AB15	AI	Regulated output of the RF1 SMPS circuit, buck inductor output side; voltage is sensed by this pin. Bypass with a 10-µF ceramic capacitor and connect it to off-chip RF circuits and/or QSC linear regulator input pins as desired. **Recommended subregulation: connect to VDD_EBI_RFA (pin AA15).**
VSW_RF2	AC17	AO	The switching output of the RF2 SMPS circuit; connect to the RF2 regulator's 2.2-µH buck inductor, input side.
VREG_RF2	AB17	AI	Regulated output of the RF2 SMPS circuit, buck inductor output side; voltage is sensed by this pin. Bypass with a 10-µF ceramic capacitor and connect it to off-chip RF circuits and/or QSC linear regulator input pins as desired. **Recommended subregulation: connect to VDD_PLL_CDC (pin AA16) and VDD_RX2_TX2 (pin AA17).**
Linear regulator connections			
VDD_PLL_CDC	AA16	P	Input power to the MPLL and CDC2 linear regulators; bypass with a 0.1-µF ceramic capacitor. **Recommended subregulation: connect to VREG_RF2 (pin AB17).**
VREG_MPLL	T13	AO	Linear regulator output that powers internal baseband PLL functions; not recommended as a general-purpose regulated power source. Bypass this pin with 2.2-µF ceramic capacitor; it is connected to the MPLL circuits internally.
VREG_CDC2	M18	AO	Linear regulator output that provides the low supply option for the internal codec circuits; connected internally and not recommended as a general-purpose regulated power source. Bypass with a 2.2-µF ceramic capacitor. The high supply option for the internal codec circuits is pin N18 (VDD_RFA).
VDD_GP	AA20	P	Input power to the two general-purpose (GP1 and GP2) linear regulators; bypass with a 0.1-µF ceramic capacitor.

Table 17-1 QSC on-chip regulator connections (continued)

Pin name/function	Pin #	Pin type [1]	Functional description
VREG_GP1	AA22	AO	Output of the first of two general-purpose linear regulators; intended for powering Bluetooth circuits, but could be used for other purposes. Bypass this pin with a 2.2-µF ceramic capacitor and connect it to the desired external circuits.
VREG_GP2	Y22	AO	Output of the second of two general-purpose linear regulators; intended for powering WLAN circuits, but could be used for other purposes. Bypass this pin with a 2.2-µF ceramic capacitor and connect it to the desired external circuits.
VDD_EBI_RFA	AA15	P	Input power to the MSME and RFA linear regulators; bypass with a 0.1-µF ceramic capacitor. **Recommended subregulation: connect to VREG_RF1 (pin AB15).**
VREG_MSME	W15	AO	Linear regulator output that powers peripheral functions (including EBI1); not recommended as a general-purpose regulated power source. Bypass this pin with a 2.2-µF ceramic capacitor. This pin is connected to several VDD_PX pins, depending upon the application (with appropriate filtering).
VREG_RFA	W16	AO	Linear regulator output intended for powering internal RF and analog circuits. Bypass with a 2.2-µF ceramic capacitor and connect it to the desired QSC input power pins (with appropriate filtering).
VDD_PAD_SIM	AA12	P	Input power to the MSMP and USIM linear regulators; bypass with a 0.1-µF ceramic capacitor
VREG_MSMP	W13	AO	Linear regulator output that powers peripheral functions; not recommended as a general-purpose regulated power source. Bypass with 2.2-µF ceramic cap. This pin is connected internally to pad group #7 (VDD_P7).
VREG_USIM	W14	AO	Linear regulator output intended for powering USIM circuits or others. Bypass this pin with a 2.2-µF ceramic capacitor.
VDD_RX2_TX2	AA17	P	Input power to the RFRX2 and RFTX2 linear regulators; bypass with a 0.1-µF ceramic capacitor. **Recommended subregulation: connect to VREG_RF2 (pin AB17).**
VREG_RFRX2	W21	AO	Linear regulator output intended for powering internal and external receiver circuits. Bypass with a 2.2-µF ceramic capacitor and connect it to the desired QSC input power pins and external RF circuits (with appropriate filtering).
VREG_RFTX2	V21	AO	Linear regulator output intended for powering internal transmitter circuits. Bypass with a 2.2-µF ceramic capacitor and connect it to the desired QSC input power pins (with appropriate filtering).
VDD_TCXO	U21	P	Input power to the TCXO linear regulator; bypass with a 0.1-µF ceramic capacitor.
VREG_TCXO	U22	AO	Linear regulator output that powers TCXO (XO) related circuits. Sensitive to external noise sources; minimize its use for powering external circuits. Bypass this pin with 1.0-µF ceramic cap. Internal TCXO regulator circuits are enabled by internal TCXO controllers.
VDD_USB	V16	P	Input power to the two USB linear regulators; bypass with a 0.1-µF ceramic capacitor.

Table 17-1 QSC on-chip regulator connections (continued)

Pin name/function	Pin #	Pin type [1]	Functional description
VREG_USB_2P6	V15	AO	First of two USB linear regulator outputs; 2.6 V nominal. If used for the internal USB transceiver, do not use to power any external circuits. If not used for the USB transceiver, this output can be used externally. Bypass this pin with a 1.0-µF ceramic cap.
VREG_USB_3P3	V14	AO	Second of two USB linear regulator outputs; 3.3 V nominal. If used for the internal USB transceiver, do not use to power any external circuits. If not used for the USB transceiver, this output can be used externally. Bypass this pin with a 1.0-µF ceramic cap.
Negative SMPS connections (negative charge pump)			
VDD_C_NCP	AA19	P	Input power to the MSMC buck SMPS and NCP negative charge pump circuits; bypass with a 2.2-µF ceramic capacitor.
VREG_NCP	AC20	AO	Negative output voltage; bypass with a 2.2-µF ceramic capacitor and connect to the desired loads (with appropriate filtering). Intended for powering headphone circuits; NCP_FB sense point is connected internally to HPH_VNEG.

1 Pin type is AI = analog input, AO = analog output, or P = power.

2 The core regulator voltage sensing is achieved internally; there is not a dedicated VREG_MSMC pin. The regulated output of the core SMPS circuit is at the buck inductor output side; this node must be bypassed with a 2.2-µF ceramic capacitor and then routed to the VDD_CORE pins (with appropriate filtering).

Many of the regulator outputs listed in Table 17-1 must be routed back on chip to supply other internal QSC circuits; some regulator outputs are connected internally to the circuits they power. All QSC power supply input pins that are powered by on-chip regulators are listed and described in Table 17-2.

Table 17-2 QSC input voltages from on-chip regulators

Supply voltage	Pin numbers	Description
Recommended connections for subregulation		
VDD_EBI_RFA	AA15; own routing and bypass cap	Connect this pin to the VREG_RF1 output (pin AB15) for higher efficiency via sub-regulation.
VDD_PLL_CDC VDD_RX2_TX2	AA16; own routing and bypass cap AA17; own routing and bypass cap	Connect these pins to the VREG_RF2 output (pin AB17) for higher efficiency via sub-regulation.
Supplies to analog/RF circuits (regulated by internal PM circuits) [1]		
VDD_RFA	C20, H18, N18, R23, T18; each with its own bypass capacitor	Power supply inputs to QSC RF and analog circuits; all should be connected to the VREG_RFA regulator output.
VDD_RFRX	Grouped as follows: 1) G19, J19; each with its own bypass cap 2) G21, K21; each with its own bypass cap plus a shared higher-value capacitor 3) H19 with its own bypass capacitor 4) K18 with its own bypass capacitor 5) K19 with its own series resistor and bypass capacitor	Power supply inputs to RF receiver circuits; all should be connected to the VREG_RFRX2 regulator output.

Table 17-2 QSC input voltages from on-chip regulators (continued)

Supply voltage	Pin numbers	Description
VDD_RFTX	Grouped as follows: 1) C18 with its own series resistor and bypass capacitor 2) C19, E18; one shared bypass capacitor 3) D21 with its own bypass capacitor 4) D22 with its own bypass capacitor 5) D23 with its own bypass capacitor	Power supply inputs to RF transmitter circuits; all must be connected to the VREG_RFTX2 output.
Supplies to baseband circuits (regulated by internal PM circuits) [2]		
VDD_CORE	Grouped as follows: 1) A1, B2, C3; share two bypass capacitors 2) K10, K11, K12, K13; share three bypass capacitors 3) L10, L11, L12, L13; share three bypass capacitors 4) U1 and U2; each with its own bypass cap	Power supply inputs to the digital core circuits; all must be connected to the VREG_MSMC node.
VDD_P1	G3, J3, L3, N3, R3; each with its own bypass capacitor	Power supply inputs to pad group 1 circuits; all must be connected to the VREG_MSME output.
VDD_P2	A6, A9; each with its own bypass capacitor	Power supply inputs to pad group 2 circuits; all must be connected to the VREG_MSME or VREG_MSMP output (VREG_MSME is expected).
VDD_P3	None	No external connections; internal VREG_USIM connection.
VDD_P4	None	No external connections; internal VREG_MSME connection.
VDD_P5	A12 with its own bypass capacitor	Power supply inputs to pad group 5 circuits; must be connected to the VREG_MSME or VREG_MSMP output (VREG_MSME is expected).
VDD_P6	A15 with its own bypass capacitor	Power supply inputs to pad group 6 circuits; all must be connected to the VREG_MSME or VREG_MSMP output (VREG_MSMP is expected).
VDD_P7	None	No external connections; internal VREG_MSMP connection.
VDD_EFUSE	V13	Power supply input to the eFuse (or Qfuse) circuits. When not being programmed, must be connected to the VREG_USIM output.

[1] These analog and RF input supply voltages are generated by the QSC internal voltage regulators included within the power management group. These inputs should be connected to the appropriate VREG outputs as identified in the description column.

[2] These digital input supply voltages are generated by the QSC internal voltage regulators included within the power management group. Some pad groupings can be connected to either VREG_MSME or VREG_MSMP depending on the application. Others must be connected to one of these sources or the other as listed in the table. The expected external connections are: VDD_P1 = VDD_P2 = VDD_P5 = VREG_MSME; VDD_P6 = VREG_MSMP. VDD_P3, VDD_P4, and VDDP7 are connected internally.

Table 17-1 and Table 17-2 list and describe the QSC power supply interconnections. In addition, some supply pins are powered directly from VPH_PWR:

- Pin AC14, VDD_CMN, powers circuits that are common to multiple QSC functions.
- Pin AC21, VDD_SPKR, powers the class D speaker driver circuits. Audio output power is enhanced when this pin is connected to VREG_5V rather than VPH_PWR.

All these power supply connections are shown in Figure 17-2 and Figure 17-3, including all their recommended external components.

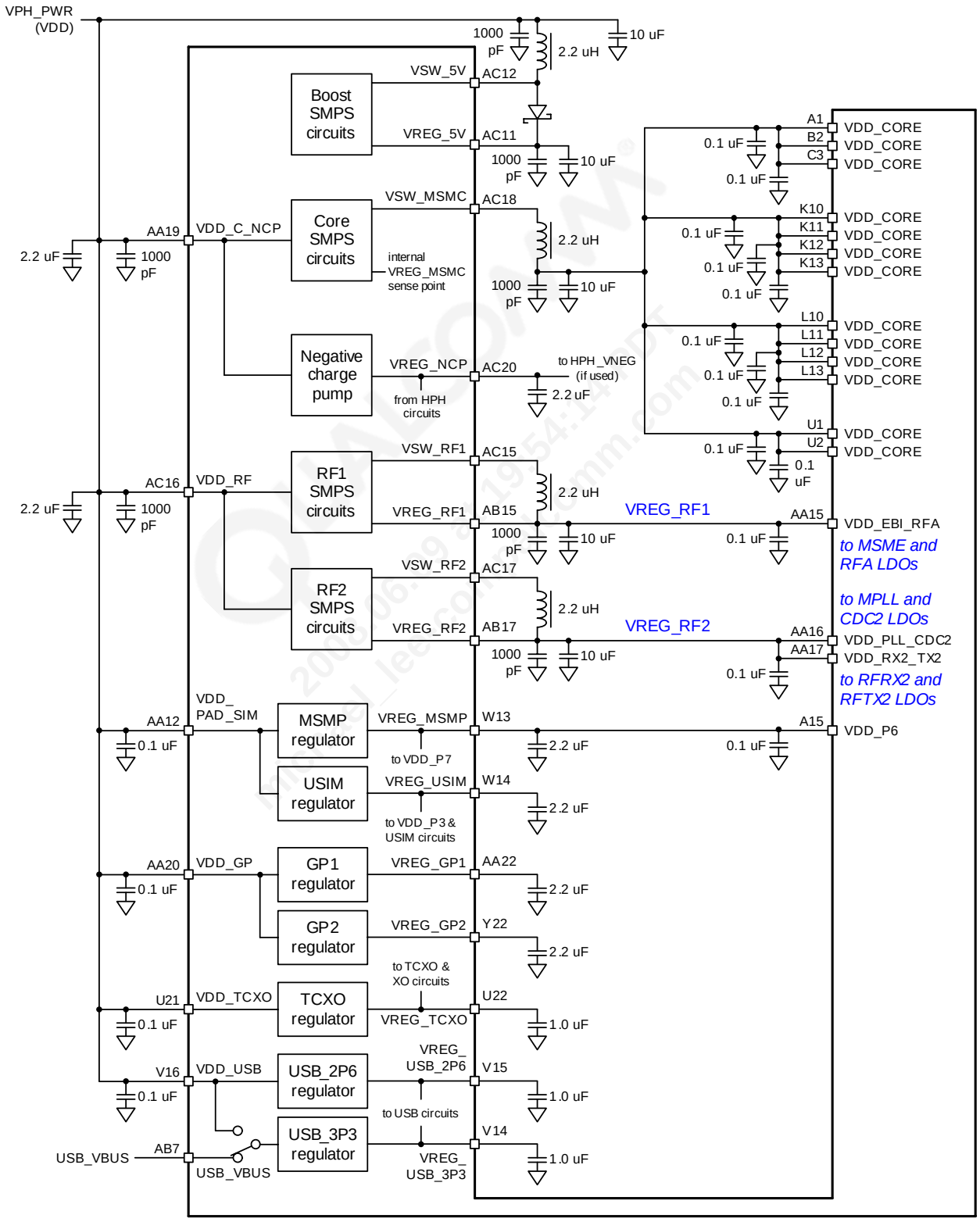


Figure 17-2 On-chip regulator outputs that power other QSC circuits (1 of 2)

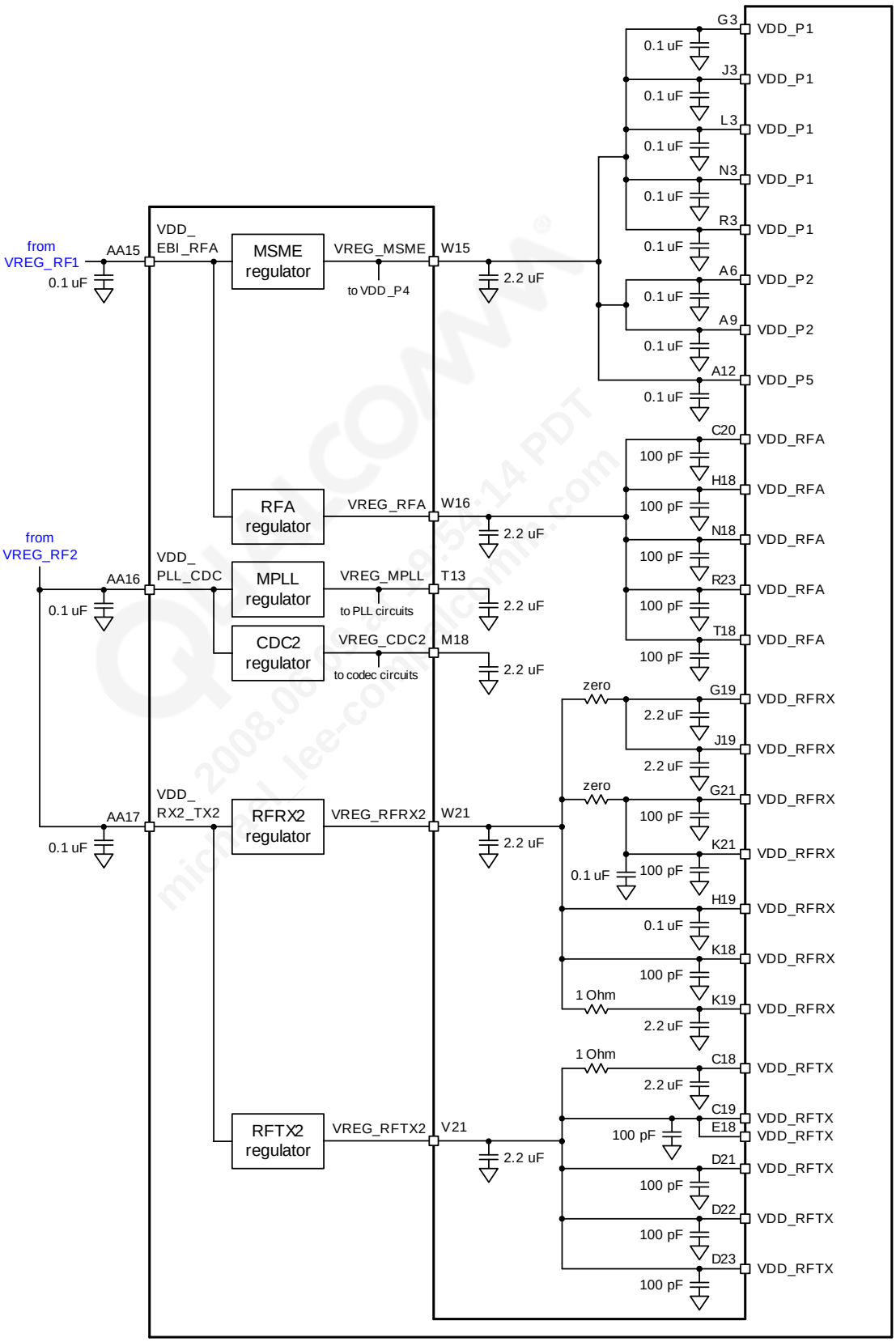


Figure 17-3 On-chip regulator outputs that power other QSC circuits (2 of 2)

The VREG_RFRX2 and VREG_RFTX2 supply distributions include some series resistors. Some are zero-Ω resistors that reserve space for an isolating element (resistor, inductor, or ferrite bead), if this is proven necessary during developmental testing. These placeholder components are recommended for initial PCB layouts. The series resistors have nonzero values in two places; these resistors are required:

- Pin K19, VDD_RFRX2: 1.0 Ω is suggested as a starting point.
- Pin C18, VDD_RFTX2: 1.0 Ω is suggested as a starting point.

17.3 QSC ground connections

Most sets of QSC ground pins must be connected into ground clusters before they are tied to the PCB ground plane. Each set of ground pins provides the reference potential and return current paths for a circuit or set of circuits within the QSC device. Connecting each subset together externally ensures all the associated internal circuits are referenced to the same potential and allows current flow with minimal resistance between circuit nodes. Ultimately each set of ground pins (each cluster) must be connected to the PCB ground plane. This concept is illustrated in Figure 17-4.

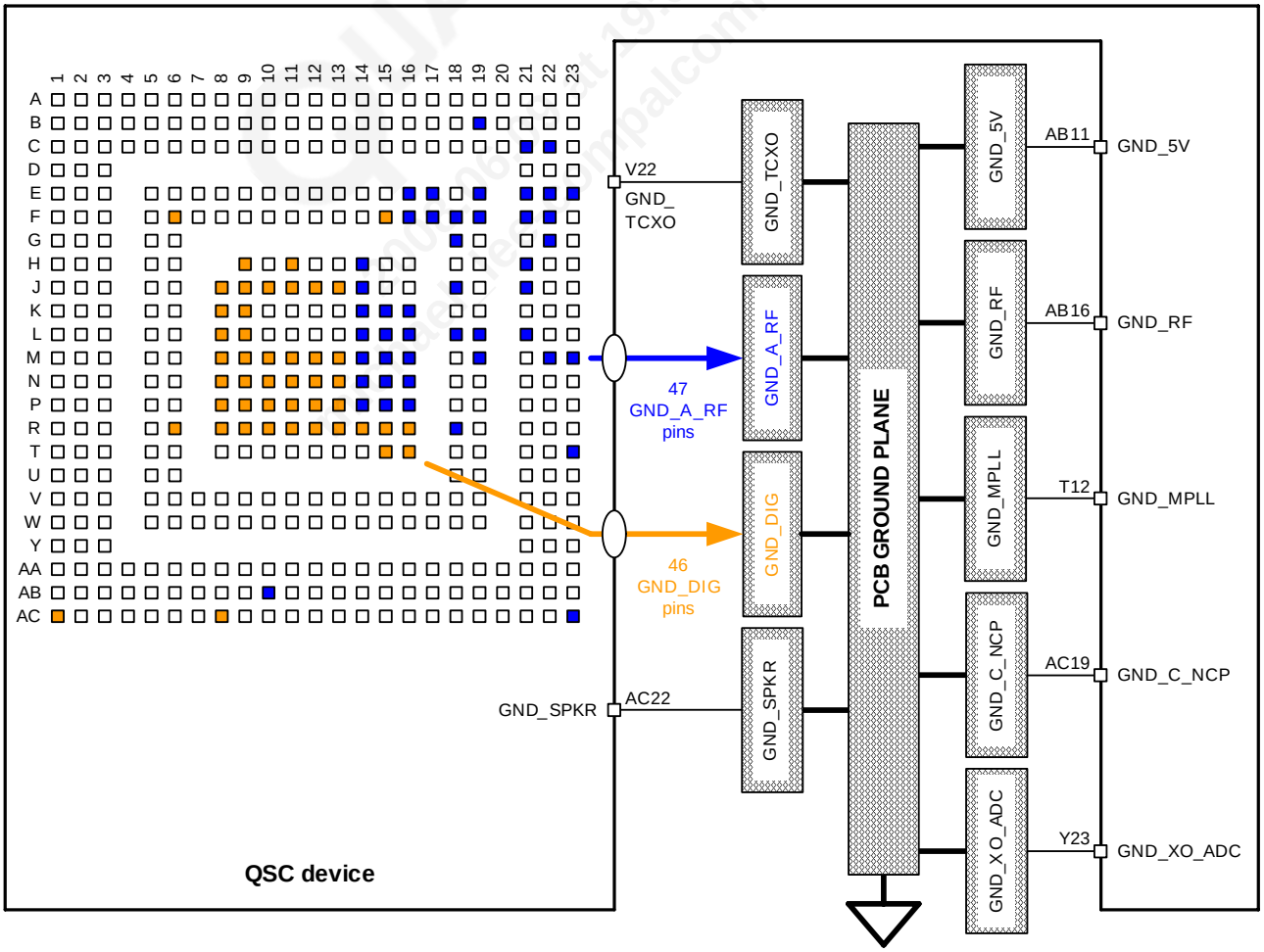


Figure 17-4 Ground connections

The PCB layout design — ground connections in particular — directly below the QSC device is extremely critical to performance. See *QUALCOMM QSC6240/QSC6270 Single Chip Design Guidelines* (80-VF846-5) for details.

The QSC ground connections are listed and described in Table 17-3.

Table 17-3 Grounds

Ground	Pin numbers	Description
GND_5V [1]	AB11	Circuit ground for the +5-V boost SMPS circuit; be aware of switching currents
GND_A_RF [2]	B19, C21, C22, E16, E17, E19, E21, E22, E23, F16, F17, F18, F19, F21, F22, G18, G22, H14, H21, J14, J18, J21, K14, K15, K16, L14, L15, L16, L18, L19, L21, M14, M15, M16, M19, M22, M23, N14, N15, N16, P14, P15, P16, R18, T23, AB10, AC23	A grouping of ground pins supporting analog and RF functions
GND_DIG [2]	F6, F15, H9, H11, J8, J9, J10, J11, J12, J13, K8, K9, L8, L9, M8, M9, M10, M11, M12, M13, N8, N9, N10, N11, N12, N13, P8, P9, P10, P11, P12, P13, R6, R8, R9, R10, R11, R12, R13, R14, R15, R16, T15, T16, AC1, AC8	A grouping of ground pins supporting digital functions
GND_C_NCP[1]	AC19	Circuit ground for negative charge pump circuits; be aware of switching currents
GND_MPLL[1]	T12	Circuit ground for the digital phase-locked loop circuits used to generate key clock signals
GND_RF[1]	AB16	Circuit ground for the RF SMPS circuits
GND_SPKR[1]	AC22	Circuit ground for the speaker driver circuits
GND_TCXO[1]	V22	Circuit ground for the TCXO circuits
GND_XO_ADC[1]	Y23	Circuit ground for the ADC circuits used for XO frequency compensation

[1] Each of these ground connections provides the ground paths for specific PM circuits. They should each be isolated from all other grounds to control return currents, then they should be connected to PCB ground in a single location.

[2] Most QSC6240/QSC6270 ground connections are divided into two categories: analog/RF (GND_A_RF) and digital (GND_DIG). These two grounds should be isolated from each other on all nonground layers (such as layers 1, 2, 3, 4, 6, 7, and 8). Each set of grounds is then tied to the primary PCB ground layer (such as layer 5) many places. See the *QSC6240/QSC6270 QUALCOMM Single Chip Design Guidelines* document (80-VF846-5) for further details.