User Manual for the BCM2035

16215 Alton Parkway • P.O. Box 57013 • Irvine, CA 92619-7013 • Phone: 949-450-8700 • Fax: 949-450-8710

2035-UM100-R
04/23/03

REVISION HISTORY

Revision	*Date*	*Change Description*
2035-UM100-R	04/23/03	Initial Draft Release.

Broadcom Corporation
P.O. Box 57013
16215 Alton Parkway
Irvine, CA 92619-7013

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Section 1: Introduction

SCOPE

The BCM2035 is a single-chip, HCI compliant bluetooth transceiver. Figure 1 illustrates the typical layers in a bluetooth system. The BCM2035 is versatile and provides several user-configurable options in hardware and software. This document describes the configurable hardware and software features for the BCM2035. This document also serves as a reference for general operation and configuration of the BCM2035 and should be used in conjunction with the BCM2035 Data Sheet when creating a BCM2035 based bluetooth design.

The BCM2035 communicates to the host processor via the bluetooth standard HCI interface. The host processor, PC or embedded, executes the bluetooth upper stack and application software. The customer is responsible for the host processor software.

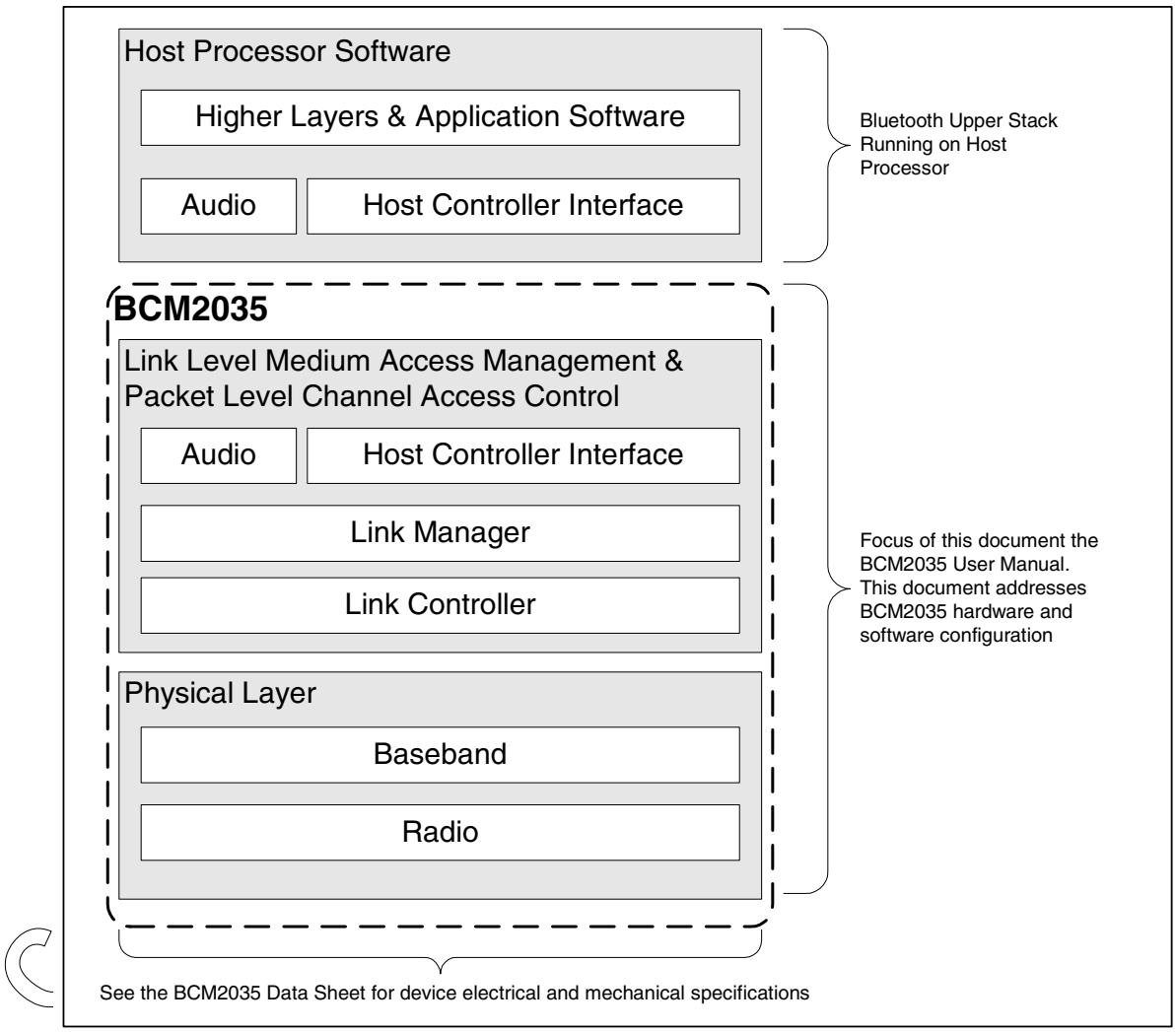

Figure 1: BCM2035 Bluetooth System Overview

Section 2: “Hardware System Overview” on page 5 provides a hardware system overview of the BCM2035. The BCM2035 is partitioned into several blocks and the overall functions of the various blocks are described in detail. This chapter also briefly describes the different configurations associated with each functional block.

Section 3: “BCM2035 Usage Models” on page 19 describes typical operational/application modes in which the BCM2035 can be configured. Currently, only usage models for mobile phones and PC based application are presented.

Section 4: “Hardware Configuration” on page 22 describes hardware configurable features for the BCM2035.

Section 5: “Software Configuration” on page 32 describes software configurable features for the BCM2035.

Section 6: “Configuration Data File System” on page 63 describes the configuration data image file format and provides a detailed listing of all firmware configurable features.

Section 7: “Vendor-Specific HCI Command Reference” on page 75 provides a reference for vendor specific commands.

Section 8: “ASCII Hex File Download Protocol” on page 90 through Section 11: “Configuration Data Image Format” on page 105 discuss in detail the file format and protocols used for downloading mini-drivers and configuration data images to the BCM2035.

Section 12: “UART Start-up Sequence” on page 108 and Section 13: “USB Start-up Sequence” on page 111 provide a startup sequence example for specific applications.

Due to the flexibility of the BCM2035, not all application configurations are addressed in this User Manual. Please contact a Broadcom representative to determine how to use the BCM2035 in your bluetooth application.

BCM2035 User Configurable Features

The BCM2035 supports the following hardware and software features.

Hardware Configurable Features

- Hardware Mode Configuration
  - Internal vs. external LPO
  - Crystal power-down configuration for active-low or active-high.
- System Clock Configuration
  - Main system clock between 12-26 Mhz for normal operation.
  - LPO clock (32.768 KHz) for low-power operation.
  - Frequency trimming
- Low Power Oscillator Configuration
- Transport Configuration
  - UART
  - USB
  - Debug UART
- EEPROM Configuration
  - Connectivity with and without EEPROM.
- External Memory Configuration
  - 8 Bit SRAM or FLASH (128 KB)
- PCM Configuration
  - Connectivity to CODEC
- Internal vs. External Voltage Regulator

SOFTWARE CONFIGURABLE FEATURES

- Startup/Boot sequence
- Transport Configuration
  - USB descriptors
  - UART baud rates
- PCM Configuration
  - Master/slave, frame sync
  - PCM data rate (128, 256, 512, 1024, 2048 kHz)
  - Data format - linear, a-law, u-law
- SCO Configuration
  - SCO over UART
  - SCO over USB
  - Test mode
- Power Management
  - Out-of-band signaling for low-power operation.
  - USB suspend and resume
  - Sniff, hold & park
- Sleep Mode Operation
- EEPROM Access Speed
  - I2C data rate at 100K, 200K & 400K
- AFH Hoping Sequence
- Fast Connection
- Configuration Data File System
- Mini-drivers
  - EEPROM update
  - Flash update

REFERENCES

- BCM2035 Data Sheet (2035-DS03-x or later).
- BCM2035 Firmware Release Notes
- Bluetooth SIG Specification of the Bluetooth System Core v1.1, February 22, 2001.
- Bluetooth SIG Specification of the Bluetooth System Profiles v1.1, February 22, 2001.

Section 2: Hardware System Overview

INTRODUCTION

The BCM2035 is Bluetooth Core Specification version 1.1 compliant and designed for use in standard HCI UART or HCI USB applications. Figure 2 shows a detailed hardware functional block diagram of the BCM2035. The combination of the Bluetooth Baseband Controller (BBC), the Peripheral Transport Unit (PTU), and the ROM based Microprocessor Unit (uPU) provide a complete lower layer Bluetooth protocol stack including the Link Controller (LC), the Link Manager (LM), and the HCI interface.

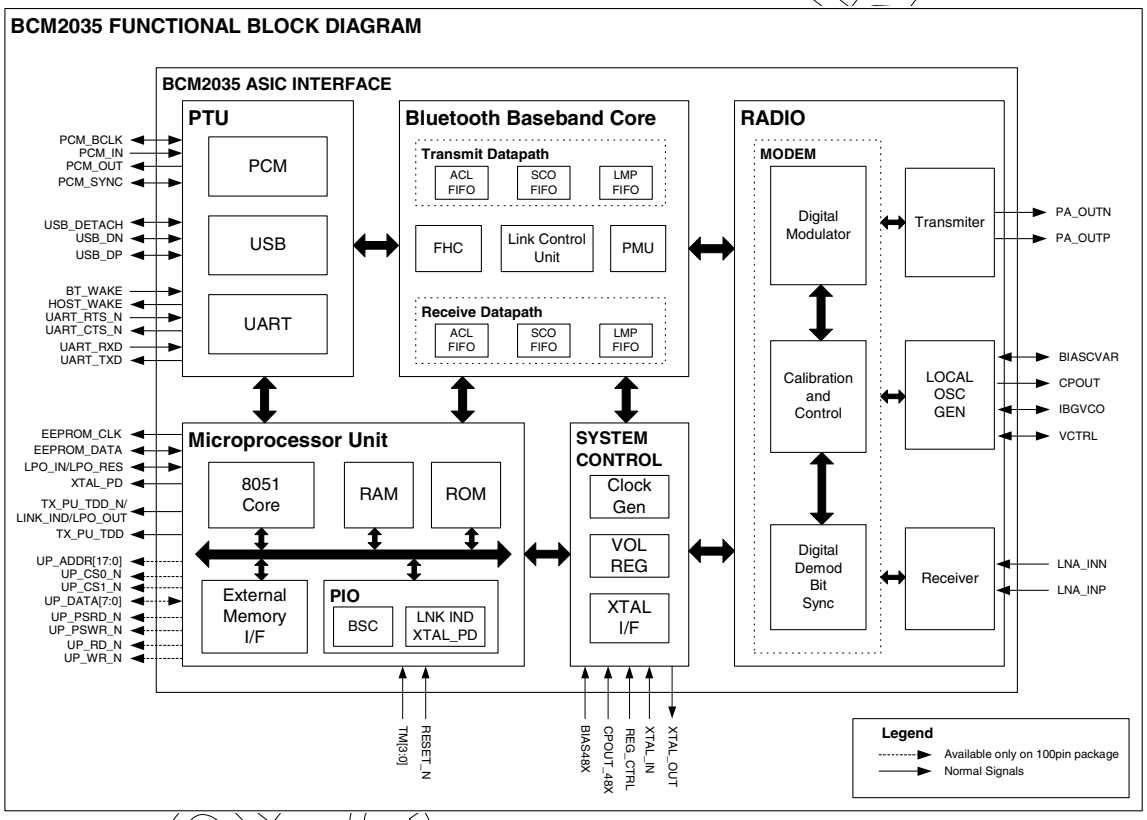

Figure 2: BCM2035 Hardware Functional Block Diagram

BCM2035 RADIO TRANSCEIVER

The BCM2035 has an integrated radio transceiver that has been optimized for use in 2.4 GHz Bluetooth wireless systems. It has been designed to provide low-power, low-cost, robust communications for applications operating in the globally available 2.4 GHz unlicensed ISM band. It is fully compliant with the Bluetooth Radio Specification v1.1 and meets or exceeds the requirements to provide the highest communication link quality of service.

MODULATOR AND DEMODULATOR (MODEM)

The BCM2035 Modem interfaces with the BBC and prepares the serial data from the baseband suitable for transmitting. The main functions of the modem block include digital modulation, digital demodulation, bit synchronization and calibration and control.

Digital Modulator

The Digital Modulator performs the data modulation and Gaussian filtering required for the GFSK signal. The fully-digital modulator minimizes any frequency drift or anomalies in the modulation characteristics of the transmitted signal and is much more stable than direct VCO modulation schemes.

Digital Demodulator and Bit Synchronizer

The Digital Demodulator and Bit Synchronizer takes the low IF received signal and performs an optimal frequency tracking and bit synchronization algorithm.

Calibration and Control

The calibration and control section provides the radio timing, command, and control functions. These have been optimized for power savings.

The BCM2035 radio transceiver features an automated calibration scheme that is fully self-contained in the radio and is fully automated. No user interaction is required during normal operation or during manufacturing to provide the optimal performance. Calibration optimizes the performance of all the major blocks within the radio to within 2% of optimal conditions, including gain and phase characteristics of filters, matching between key components, and key gain blocks. This takes into account process variation and temperature variation. Calibration occurs transparently during normal operation during the settling time of the hops and calibrates for temperature variations as the device cools and heats during normal operation in its environment.

TRANSMITTER PATH

The BCM2035 features a fully-integrated zero IF transmitter. The baseband transmit data is digitally GFSK modulated in the modem block and up-converted to the 2.4 GHz ISM band in the Transmitter Path. The Transmitter Path consists of signal filtering, I/Q up-conversion, output power amplifier (PA), and RF filtering.

Power Amplifier (PA)

The fully-integrated PA provides a maximum output signal level of +7 dBm using a highly linearized, temperature compensated design. This gives the user greater flexibility and options in the type of front-end matching and filtering to use with the BCM2035. Due to the linear nature of the PA, combined with the fully-integrated filtering, no external filters are required for meeting Bluetooth and regulatory harmonic and spurious requirements. For integrated mobile handset applications where the Bluetooth is integrated next to the cellular radio, minimal external filtering can be applied to achieve near thermal noise levels for spurious and radiated noise emissions.

The PA supports a power offset that can be set to one of four maximum output power levels in 2 dB increments. In addition, 24 dB of power control in 2 dB steps is supported for power control functions.

RECEIVER PATH

The Receiver Path uses a 2 MHz low IF scheme to down-convert the received signal for demodulation in the digital demodulator and bit synchronizer. The Receiver Path provides a high degree of linearity, an extended dynamic range, and high order on-chip channel filtering to ensure reliable operation in the noisy 2.4 GHz ISM band. The on-chip filtering enables the BCM2035 to be used in most applications with no off-chip filtering. For integrated handset operation where the Bluetooth function is integrated close to the cellular transmitter, minimal filtering is required to eliminate the desensitization of the receiver by the cellular transmit waveform.

Receiver Signal Strength Indicator (RSSI)

The radio portion of the BCM2035 provides an RSSI signal to the baseband so that the controller can take part in a Bluetooth power-controlled link by providing a metric of its own receiver signal strength to determine whether the transmitter should increase or decrease its output power.

LOCAL OSCILLATOR GENERATION

Local Oscillator (LO) generation provides fast frequency hopping (1600 hops/second) across the 79 maximum available channels. The LO generation sub-block employs a proprietary architecture for high immunity to LO pulling during PA operation.

The BCM2035 uses an external RF and IF loop filter. The RF loop filter uses the different values depending on the reference frequency or crystal being used. The BCM2035 Data Sheet has recommendations for the external RF and IF loop filter topology and component values.

BLUETOOTH BASEBAND CORE

The Bluetooth Baseband Core (BBC) implements all of the time critical functions required for high performance Bluetooth operation. The BBC manages the buffering, segmentation, and routing of data for all connections. It also buffers data that passes through it, handles data flow control, schedules SCO/ACL TX/RX transactions, monitors Bluetooth slot usage, optimally segments and packages data into baseband packets, manages connection status indicators, and composes and decodes HCI packets. In addition to these functions, it independently handles HCI event types, and HCI command types.

Several transmit and receive functions are also implemented in the BBC hardware to increase reliability and security of the Tx/Rx data before sending over the air.

In the receive datapath, the BBC performs symbol timing recovery, data de-framing, forward error correction (FEC), header error control (HEC), cyclic redundancy check (CRC), data decryption, and data de-whitening.

In the transmit datapath, the BBC performs data framing, FEC generation, HEC generation, CRC generation, key generation, data encryption, and data whitening.

FREQUENCY HOPPING GENERATOR

The frequency hopping sequence generator selects the correct hopping channel number depending on the Link Controller state, Bluetooth clock, and the device address.

LINK CONTROL LAYER

The Link Control Layer is part of the Bluetooth link control functions that are implemented in dedicated logic in the Link Control Unit (LCU). This layer consists of the Command Controller that takes commands from the software, and other controllers that are activated or configured by the Command Controller to perform the Link Control tasks. Each task performs a different state in the Bluetooth Link Controller. There are two major states: STANDBY and CONNECTION. In addition, there are seven sub-states: PAGE, PAGE SCAN, INQUIRY, INQUIRY SCAN, PARK, SNIFF, and HOLD.

TEST MODE SUPPORT

The BCM2035 fully supports Bluetooth Test Mode as described in Part I:1 of the Specification of the Bluetooth System Version 1.1. This includes the Transmitter Tests, Normal and Delayed Loopback Tests, and Reduced Hopping Sequence.

In addition to the standard Bluetooth Test Mode, the BCM2035 also supports enhanced testing features to simplify RF debugging and qualification and type approval testing. These features include:

- Fixed frequency carrier wave (unmodulated) transmission
  - Simplifies some type approval measurements (Japan)
  - Aids in transmitter performance analysis
- Fixed frequency constant receiver mode
  - Receiver output directed to I/O pin
  - Allows for direct BER measurements using standard RF test equipment
  - Facilitates spurious emissions testing for receive mode

- Fixed frequency constant transmission
  - 8-bit fixed pattern or PRBS-9
  - Enables modulated signal measurements with standard RF test equipment
- Fixed frequency connectionless Bluetooth packet transmission
  - All packet types supported
  - 8-bit fixed pattern or PRBS-9 payload

POWER MANAGEMENT UNIT

The Power Management Unit (PMU) provides power management features that can be invoked by either software through power management registers, or packet handling in the baseband core. The following sections are power management functions provided by the BCM2035.

RF Power Management

The BBC generates power down control signals for the transmit path, receive path, PLL, and power amplifier to the 2.4-GHz transceiver. The transceiver then processes the power down functions accordingly.

BBC Power Management

There are several low power operations for the BBC:

Physical layer packet handling turns RF on and off dynamically within packet Tx and Rx.

Bluetooth specified low-power connection modes: Sniff, Hold, and Park. While in these low-power connection modes, the BCM2035 runs on the Low Power Oscillator and wakes up after a pre-defined time period.

Lower Power Shutdown

The BCM2035 provides a Low Power Shutdown feature that allows the device to be turned off while the host, and any other devices in the system, remain operational. When the BCM2035 is not needed in the system, VDD_RF and VDD_CORE are shut down while VDD_IO is left powered. This allows the BCM2035 to be effectively off while keeping the I/O pins powered so that they do not draw extra current from any other devices connected to the I/O.

During low power shutdown state, as long as VDD_IO remained applied to the BCM2035, all outputs are tri-stated and most inputs signals are disabled. Input voltages must remain within the limits defined for normal operation. This is done to prevent current paths or create loading on any digital signals in the system, and enables the BCM2035 to be fully integrated in an embedded device and take full advantage of the lowest power savings modes.

Two signals on BCM2035, the frequency reference input (XTAL_IN) and external LPO input (LPO_IN), are designed to be high impedance inputs that will not load down the driving signal regardless of if the chip has VDD_IO power applied to it or not.

When the BCM2035 is powered on from this state, it is the same as a normal power-up and the device does not contain any information about its state before being powered-down.

Host Controller Power Management

When running in USB mode, the BCM2035 supports the USB version 1.1 specification, suspend/resume signaling as well as remote wake-up signaling for power control.

When running in UART mode, the BCM2035 may be configured such that dedicated signals are used for power management hand-shaking between the BCM2035 and the host. The basic power saving functions supported by those hand-shaking signals include the standard Bluetooth defined power savings modes as well as standby modes of operation.

An alternative to using the BT_WAKE and HOST_WAKE signaling uses the CTS and RTS as a combination of UART hardware handshake signals during normal operation and as BT_WAKE and HOST_WAKE when the device has been placed into a power savings mode, as described in Table 1.

Table 1: Power Savings Mode Descriptions

Pin	*Direction*	*Description*
BT_WAKE	Host Output BT Input	Bluetooth Device wake-up: Signals from the host to the Bluetooth device that the host requires attention. Asserted = Wake up or remain awake (depending on context) De-asserted = Bluetooth may sleep when sleep criteria are met. This signal can be configured to be ASSERTED HIGH or ASSERTED LOW via vendor specific command (HCI_Write_Sleep_Mode) or via configuration data.
HOST_WAKE	BT Output Host Input	Host Wake-up: Signals from the Bluetooth device to the host that the Bluetooth device requires attention. Asserted = Wake up or remain awake (depending on context) De-asserted = Host may sleep when sleep criteria are met. This signal can be configured to be ASSERTED HIGH or ASSERTED LOW via vendor specific command (HCI_Write_Sleep_Mode) or via configuration data.
XTAL_PD	BT Output	Crystal Power-Down signal is ASSERTED by the BCM2035 to indicate that the system clock can be shut-down (this feature no longer requires the device be in External LPO mode). Crystal Power-Down mode: Asserted = external frequency reference may be powered down De-asserted = external frequency reference required Crystal power down mode is set through the mode strap pins (TM[3:0]). The Crystal Power-Down signal can be configured via the TM[3:0] bits to be ASSERTED HIGH or ASSERTED LOW.

ADAPTIVE FREQUENCY HOPPING (AFH)

The BCM2035 gathers link quality statistics on a channel-by-channel basis to facilitate channel assessment and channel map selection. The BCM2035 also integrates a programmable hardware acceleration engine and frequency hop mapping tables to enable future support of AFH.

FAST CONNECTION

The BCM2035 supports page scan and inquiry scan modes that significantly reduce the average inquiry response and connection times. These scanning modes are compatible with the Bluetooth version 1.1 page and inquiry procedures and are designed to be forward compatible to Bluetooth version 1.x extension fast connection mode.

MICROPROCESSOR UNIT

The Microprocessor Unit (uPU) runs software from the Link Control (LC) layer, up to the Host Controller Interface (HCI). The microprocessor is an enhanced performance 8051 microcontroller. The uPU also consists of the peripheral input/output, external memory interface (only available in the 100-pin and 104-pin package for debug), 144 kbytes of ROM memory for program storage and boot ROM, 20 kbytes of RAM for data scratchpad and patch RAM code, and interface to the PTU.

The internal boot ROM allows for flexibility during power-on reset to enable the same device to be used in various configurations, including USB or UART and with or without an external serial EEPROM. At power-up, the lower layer protocol stack is executed from the internal ROM memory.

External patches may be applied to the ROM based firmware to provide flexibility for bug fixes or features additions. These patches may be downloaded from the host to the BCM2035 through the USB or UART transports, or by using an external serial EEPROM memory. The mechanism for downloading via UART or USB is identical to the proven interface of the BCM2035.

Optionally, for code development, a 100-pin or 104-pin version of the component is available which allows for the interface to an external flash memory.

The 8051 core is object code compatible with the industry standard 8051 microcontroller.

PROGRAMMABLE I/O (PIO) PORT

The PIO or General Purpose I/O in the BCM2035 have been assigned to specific functions. Table 2 lists the signal names and provides a brief description of the BCM2035 GPIO's.

Table 2: PIO/GPIO Assignment for the BCM2035

Pin	*Direction*	*Description*
GPIO0	BT_WAKE	BT_WAKE and HOST_WAKE are side band signals used for low power operation while using the UART transport.
GPIO1	XTAL_PD	XTAL_PD is an output signal that can be used to shutdown system clock for low power operation. This can be configured by the user as an active high or an active low signal.
GPIO2	ADDR_17	This signal is used as an address pin to enable the 8051 processor to access a 2M Flash. This option is only available on the 100-pin and 104-pin package.
GPIO3	HOST_WAKE	BT_WAKE and HOST_WAKE are side band signals used for low power operation while using the UART transport.

Table 2: PIO/GPIO Assignment for the BCM2035 (Cont.)

Pin	Direction	Description
GPIO4	TX_PU_TDD_N/ LINK_IND/ LPO_OUT	This pin has multiple functions. In normal mode, the default configuration is TX_PU_TDD_N (see "External TDD Switch Control" on page 14). In RF mode, the signal functions as the LPO_OUT indicator. The signal can be configured by the user for Link indication (LINK_IND).
GPIO5	EEPROM_CLK	EEPROM_CLK and EEPROM_DATA signals are used to connect with an external EEPROM with I2C interface.
GPIO6	EEPROM_DATA	EEPROM_CLK and EEPROM_DATA signals are used to connect with an external EEPROM with I2C interface.
GPIO7	USB_DETACH	This signal is used by the BCM2035 firmware to signal USB device detachment and attachment to the HOST.

EEPROM Interface

The BCM2035 provides the BSC (Broadcom Serial Control) master interface; the BSC is programmed by the CPU to generate four different types of BSC transfers on the bus -- read-only, write-only, combined read/write, and combined write-read. BSC supports both low-speed and fast mode devices. The BSC is compatible with I2C slave devices, except that multiple I2C masters are not supported, and flexible wait state insertion by either the master interface or slave devices are not support.

The EEPROM may contain configuration information concerning the customer application, including the following: Fractional-N information, BD_ADDR, baud rate, USB enumeration information, SDP service record, and file system information used for code, code patches, or data.

External Memory Interface (100-Pin and 104-Pin Package Only)

The memory interface (available only on the 100-pin and 104-pin package) allows 8051 microcontroller accesses to two types of 8-bit wide external memory: Flash memory and SRAM. The interface can access 128 KB of external Flash memory, or 128 KB of external SRAM with no bank switching required.

RAM, ROM, and Patch Memory

The BCM2035 has 20 kbytes of internal RAM, which is mapped between general-purpose scratch pad memory and patch memory and 144 kbytes of ROM used for the lower layer protocol stack, test mode software, and boot ROM. The patch memory capability enables the addition of code changes for purposes of feature additions and bug fixes to the ROM memory.

SYSTEM CONTROL

SYSTEM CLOCK

The BCM2035 uses two different clock references for normal and low-power operational modes. For normal operation, the BCM2035 uses an external crystal or external frequency reference driven by a temperature compensated crystal oscillator (TCXO) to generate all radio frequencies and system timing references. For low power mode operation, the BCM2035 uses either an external 32.768 kHz or fully integrated internal low-power oscillator (LPO) for system timing reference.

EXTERNAL RESET

The BCM2035 has an integrated power-on reset circuit, which will which will completely reset all circuits to a known power on state. This action can also be driven by an external reset signal, which can be used to externally control the device, forcing it into a power-on reset state. The signal input, RESET_N is an active low signal that has an internal pull-up resistor and is not required to be connected in most applications. No external pull-up resistor is required.

EXTERNAL TDD SWITCH CONTROL

The BCM2035 provides two signals to control an external time division duplex (TDD) or transmit and receiver (T/R) switch. A differential output, TX_PU_TDD and TX_PU_TDD_N, active high and active low signals are provided to simplify interfacing to any switch or pin-diode type switch.

PERIPHERAL TRANSPORT UNIT

PCM INTERFACE

The PCM Interface on the BCM2035 can connect to linear PCM Codec devices in master or slave mode. In master mode, the BCM2035 generates the PCM_BCLK and PCM_SYNC signals, and in slave mode, these signals are provided by another master on the PCM interface and are inputs to the BCM2035. The PCM interface may also be configured in a hybrid mode such that the PCM_BCLK signal comes from the master, while the PCM_SYNC signal comes from the slave, or vice-versa.

The BCM2035 supports up to three SCO channels through the PCM Interface and each channel can be independently mapped to any of the available slots in a frame.

The configuration of the PCM interface may be adjusted by the host through the use of Vendor Specific HCI Commands.

SYSTEM DIAGRAM

Figure 3 shows three options for connecting a BCM2035 to a PCM codec device as either a master, slave, or hybrid type connection.

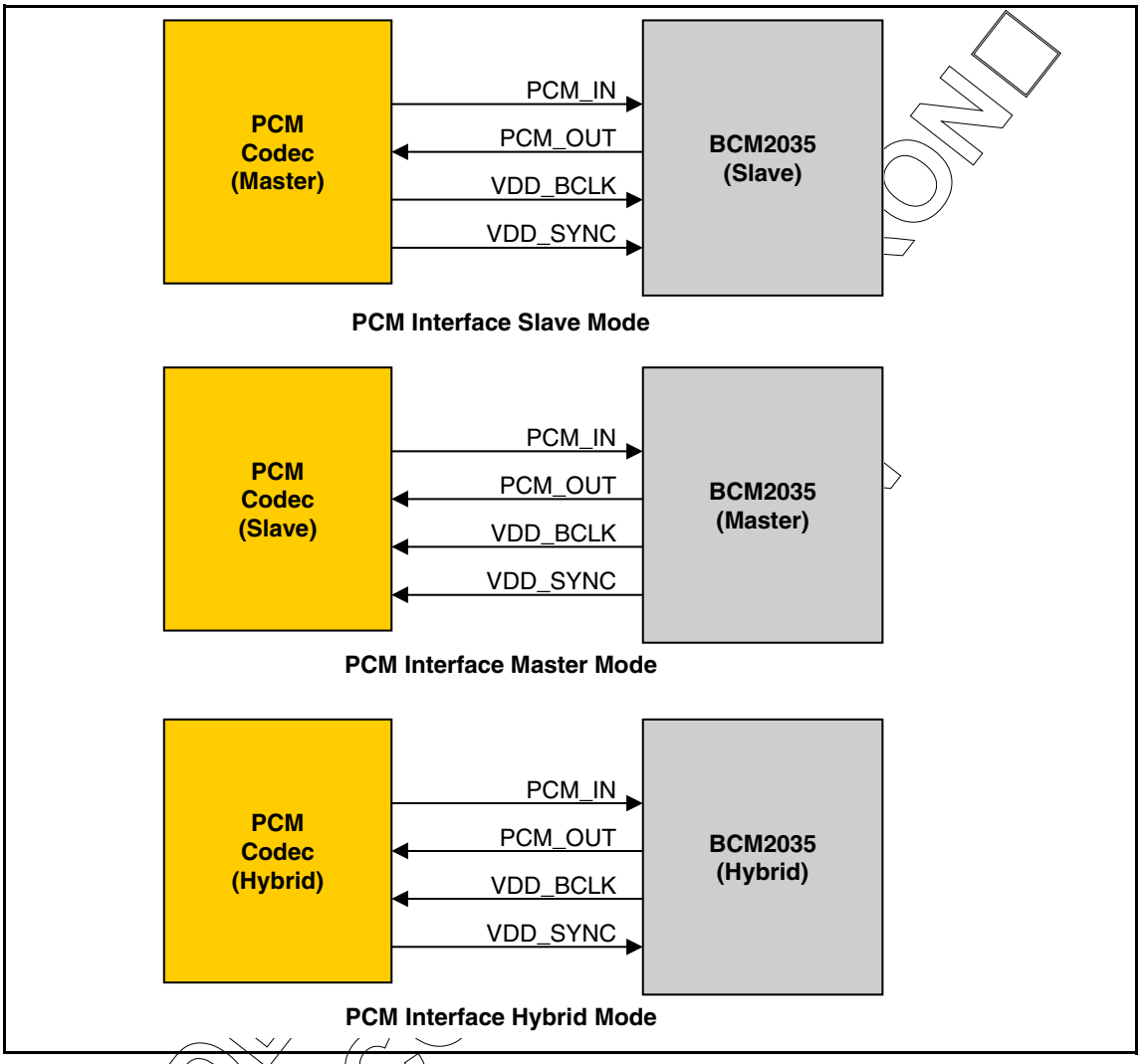

Figure 3: BCM2035 PCM Interface with Linear PCM Codec

SLOT MAPPING

The BCM2035 supports up to 3 simultaneous full duplex SCO channels. These 3 channels are time multiplexed onto the single PCM interface by using a time slotting scheme where the 8 kHz audio sample interval is divided into up to 16 slots. The number of slots is dependant on the selected interface rate of 128, 256, 512, 1024, or 2048 kHz. The corresponding number of slots for these interface rates is 1, 2, 4, 8 and 16 respectively. Transmit and receive PCM data from an SCO channel is always mapped to the same slot. The PCM data output driver tri-states its output on unused slots to allow other devices to share the same PCM interface signals. The data output driver tri-states its output after the falling edge of the PCM clock during the last bit of the slot.

FRAME SYNC

The BCM2035 supports both short and long frame sync types in both master and slave configurations. In the short frame sync mode, the frame sync signal is an active high pulse at the 8 kHz audio frame rate that is a single bit period in width and synchronized to the rising edge of the bit clock. The PCM slave will look for a high on the falling edge of the bit clock and expect the first bit of the first slot to start at the next rising edge of the clock. In the long frame sync mode, the frame sync signal is again an active high pulse at the 8 kHz audio frame rate; however, the duration is 3 bit periods and the pulse starts coincident with the first bit of the first slot.

DATA FORMATTING

The BCM2035 may be configured to generate and accept several different data formats. The BCM2035 uses 13 of the 16 bits in each PCM frame. The location and order of these 13 bits is configurable to support various data formats on the PCM interface. The remaining three bits will be ignored on the input, and may be filled with 0's, 1's, sign bit, or a programmed value on the output. The default format is 13-bit 2's complement data, left justified and clocked most significant bit first.

USB INTERFACE

The USB port is a USB 1.1 high-speed compliant slave interface, with on-chip USB transceiver. The USB interface on the BCM2035 complies with the HCI USB specification as described in the Specification of the Bluetooth System Version 1.1.

The BCM2035 detects automatically detects activity on the USB interface when connected, so no special configuration is needed to select HCI as the transport. The USB interface cannot be used simultaneously with HCI UART.

The USB port contains three logical interfaces. Interface 0 contains a Control Endpoint (Endpoint 0x00) for HCI commands, a Bulk In Endpoint (Endpoint 0x82) for receiving ACL data, a Bulk Out Endpoint (Endpoint 0x02) for transmitting ACL data, and an Interrupt Endpoint (Endpoint 0x81) for HCI events.

Interface 1 contains Isochronous In and Out Endpoints (Endpoints 0x83 and 0x03) for SCO traffic. Several alternate settings of Interface 1 are available to reserve the proper bandwidth for the isochronous data depending on the application.

Interface 2 contains Bulk In and Bulk Out Endpoints (Endpoints 0x84 and 0x04) which are used for proprietary testing and debugging support and may be ignored during normal operation.

The USB Device Descriptor, Configuration Descriptor, and String Descriptor are fully programmable if the default settings are not desired. This allows the end manufacturer to customize the descriptors that the BCM2035 uses to identify itself on the USB port, including the Vendor ID and Product ID. Up to two sets of custom descriptors can be supplied so that the BCM2035 may boot using one set of descriptors and then later re-enumerate using the second set of descriptors. The custom USB descriptor information may be stored in external EEPROM so that it is available at boot time.

The USB Descriptor Tables on the following pages contain detailed default descriptor information for the USB interface.

UART

The UART physical interface is a standard, 4-wire interface (RX,TX, RTS, CTS) with adjustable baud rates from 9600 bps to 1.5 Mbps. The interface features an automatic baud rate detection capability that will return a baud rate selection from Table 3. Alternatively, the baud rate may be selected via a vendor specific UART HCI command. It has a 240-byte receive FIFO and a 240-byte transmit FIFO. The interface supports the Bluetooth 1.1 UART HCI specification.

Table 3: Common Baud Rate Examples

Desired Baud Rate (bps)	*UART Clock Divisor (256-Desired Divisor)*	*Baud Rate Adjustment*	*Actual BCM2035 Baud Rate (bps)*	*% Error*
1500000	0XFF	0x00	1500000	0.00%
921600	0XFF	0x55	923077	0.16%
460800	0xFD	0x22	461538	0.16%
230400	0xFA	0x44	230796	0.17%
115200	0xF3	0x00	115385	0.16%
57600	0xE6	0x00	57692	0.16%
38400	0xD9	0x01	38400	0.00%
28800	0xCC	0x00	28846	0.16%
19200	0xB2	0x11	19200	0.00%
14400	0x98	0x00	14423	0.16%
9600	0x64	0x22	9600	0.00%

The baud rate of the BCM2035 UART is controlled by two values. The first is a UART Clock Divisor (also called the DLBR register) that divides the 24 MHz reference clock by an integer multiple of 16. The second is a Baud Rate Adjustment (also called the DHBR register) that is used to specify a number of 24 MHz clock cycles to stuff in the first or second half of each bit time. Up to eight 24 MHz clock cycles can be inserted into the first half of each bit time, and up to eight 24 MHz clock cycles can be inserted into the end of each bit time.

When setting the baud rate manually, the UART Clock Divisor is an 8-bit value that is stored as 256 - Desired Divisor. For example, a desired divisor of 13 is stored as 256-13=243=0xF3.

The Baud Rate Adjustment is also an 8-bit value, of which the 4 most significant bits are the number of additional clock cycles to insert in the first half of each bit time, and the 4 least significant bits are the number of clock cycles to insert in the second half of each bit time. If either of these two values is over 8, it is rounded to 8.

The baud rate of the BCM2035 UART is expressed as:

24 MHz/((16xUART Clock Divisor) + Total Inserted 24 MHz Clock Cycles)

Normally, the UART baud rate will be set by the parameters stored in the optional serial EEPROM, or by automatic baud rate detection and the host will not need to adjust the baud rate. Support for changing the baud rate during normal HCI UART operation is included through a Vendor Specific command that allows the host to adjust the contents of the baud rate registers.

The BCM2035 UART will operate correctly with the Host UART as long as the combined baud rate error of the two devices is within ± 5%.

Auto-Baud Rate Detection

If no EEPROM is connected to the BCM2035 containing UART configuration information, the BCM2035 may be put into a state where it will attempt to automatically detect the baud rate. A 'U' character (Hex 0x55) is sent from the Host to train the BCM2035 UART when this feature is used.

Section 3: BCM2035 Usage Models

MOBILE PHONE USAGE MODEL

The BCM2035 is designed to directly interface with new and existing handset designs as shown in Figure 4. The BCM2035 has a very flexible PCM and UART interfaces, enabling it to transparently connect with the existing circuits. In addition, the TCXO and external LPO inputs allow the use of existing features of the handset to further minimizing the size, power, and cost of the integration.

The BCM2035 incorporates a number of unique features to accommodate the integration into mobile phone platforms.

1. The PCM interface and UART interface have separate power domains allowing the BCM2035 to interface with two separate devices with different interface voltage requirements.
2. The PCM interface provides multiple modes of operation to support both master and slave as well as hybrid interfacing to single or multiple external codec devices.
3. The UART interface supports both hardware and software flow based flow control and with tight integration with power control side band signaling to support the lowest power operation.
4. The TCXO interface accommodates any of the typical reference frequencies used by cell phones.
5. A programmable TCXO power-up or power-down signal (active high or active low) allows the device to interface with TCXO devices with different start-up times.
6. Both the TCXO and external LPO inputs are high impedance inputs that have minimal loading on the driving source regardless of if the BCM2035 has power applied to it or power has been removed.
7. The internal power control on the transceiver power amplifier can be offset to a lower output power, saving 25% over the normal transmit power consumption.
8. The highly linear design of the radio transceiver ensures that the device has the lowest output spurious emissions regardless of the state of operation and has been fully characterized in the global cellular bands.
9. The transceiver design has excellent blocking (eliminating desensitization of the Bluetooth receiver) and intermodulation performance (distortion of the transmitted signal caused by the mixing of the cellular and Bluetooth transmissions) in the presence of any cellular transmission (GSM, GPRS, CDMA, WCDMA, or iDEN). Minimal external filtering is required for integration inside the handset.
10. Minimal external components are required for integration and very compact packaging is available, eliminating the need for modules

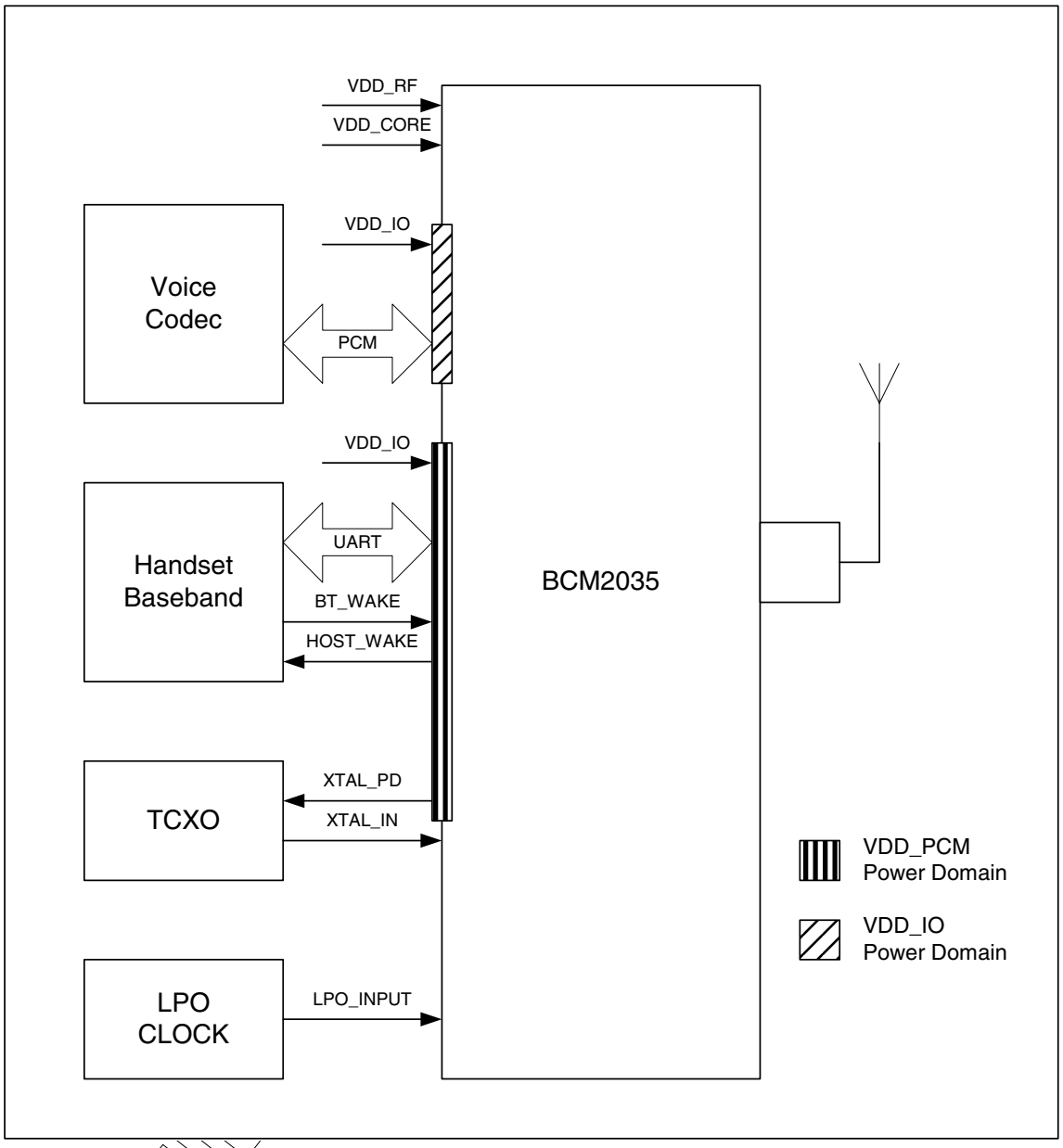

Figure 4: Mobile Phone Usage Model

PC PRODUCT USAGE MODEL

The BCM2035 can be directly interfaced using the HCI USB interface and fully supports embedded USB applications such as laptop PC and PC motherboard integration, or as an external USB dongle peripheral device. Figure 5 shows an example of a PC product usage model.

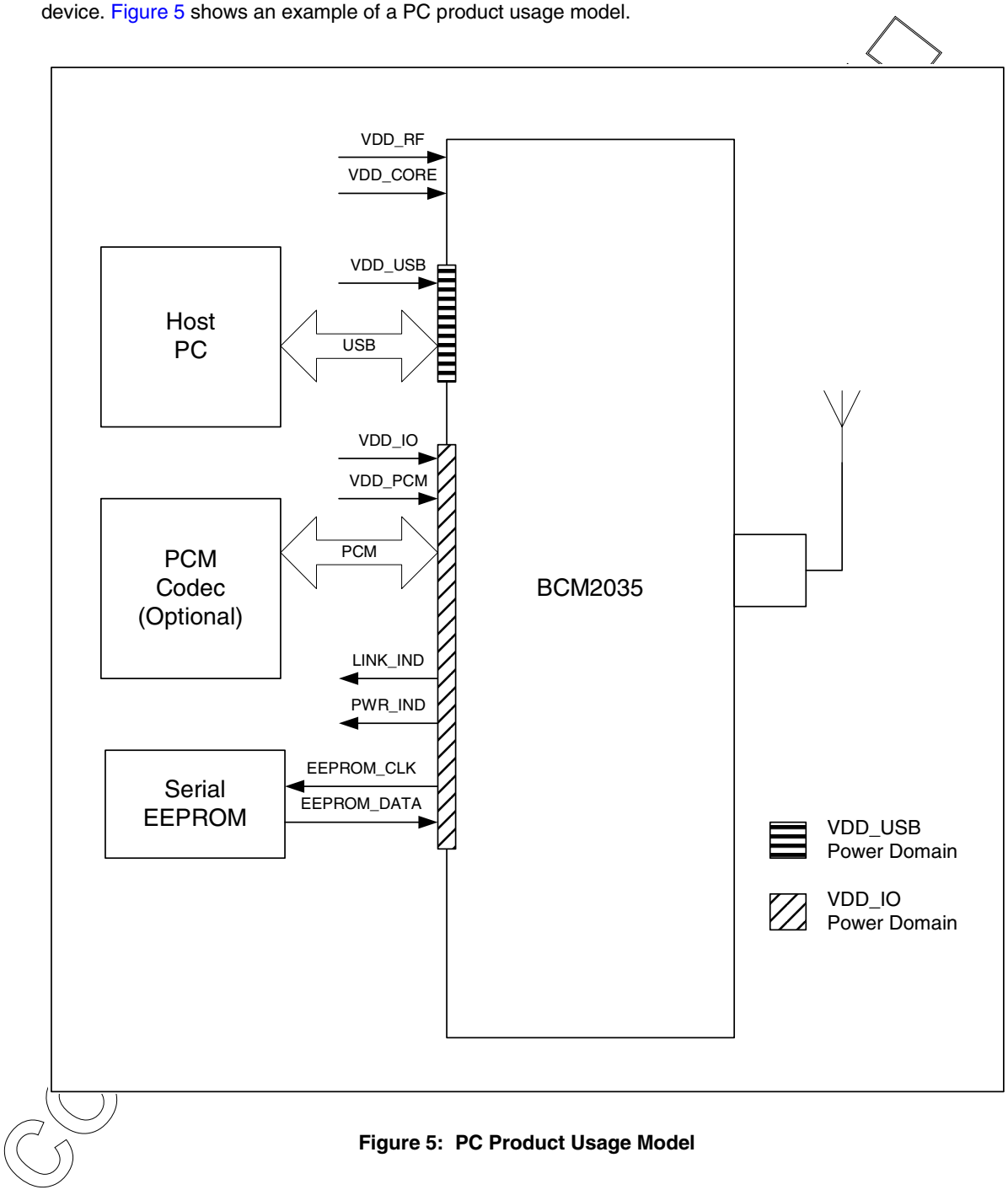

Figure 5: PC Product Usage Model

Section 4: Hardware Configuration

The BCM2035 is a single chip, HCI compliant Bluetooth transceiver. The Link Manager and Host Controller Interface layers reside in the BCM2035 ROM, so no external Flash is required to operate the BCM2035. UART and USB interfaces are supported for connection to the Host system.

The BCM2035 employs several methods of configuration to provide support for specific applications. Various items can be configured, including the reference crystal frequency, the transport to be used (USB or UART), the use of internal or external Low Power Oscillator, option to use internal voltage regulator, USB descriptors, UART Baud Rate and Bluetooth Device Address.

Mode Pins

The hardware configuration of the BCM2035 is done through a combination of strapping the Mode Pins (TM[3:0]), and correctly connecting the desired transport (USB or UART).

The Mode Pins, TM[3:0] are used to set configuration of specific hardware blocks inside the BCM2035. Specifically, the Mode Pins control the selection of internal or external Low Power Oscillator (LPO), and the polarity of the XTAL_PD signal.

Table 4 defines the configurations selectable by the Mode Pins.

Table 4: Mode Pin Settings

TM_3	*TM_2*	*TM_1*	*TM_0*	*Description*
0	0	0	0	Internal LPO Enabled, XTAL_PD is active HIGH (Crystal may be powered down when XTAL_PD is HIGH).
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	Reserved
0	1	1	1	External LPO Required, XTAL_PD is active HIGH.
1	0	0	0	Internal LPO Enabled, XTAL_PD is active LOW.
1	0	0	1	External LPO Required, XTAL_PD is active LOW.
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

The Internal/External LPO selection allows a design to take advantage of a precise external 32.768 kHz oscillator if one is present in the design. The internal LPO may be used in a design in which no external 32.768 kHz oscillator is available. The advantage of using an external LPO when possible is that the higher precision will allow longer intervals to be used in the power savings modes (Park, Sniff, Hold).

See "Low Power Oscillator" on page 24 regarding LPO connections for more details about how the BCM2035 I/O should be connected for internal or external LPO designs.

The XTAL_PD pin is an output that is used by the BCM2035 to indicate when the main clock may be shut off in a design that is feeding a clock signal to the BCM2035 rather than using the build in crystal oscillator circuit with an external crystal. The mode pins are used to select the polarity of the XTAL_PD pin.

The XTAL_PD signal is not needed in a design using the BCM2035 oscillator circuit with an external crystal, because the BCM2035 shuts down the oscillator circuit internally when it is not needed.

SYSTEM CLOCK CONFIGURATION

The BCM2035 uses a fractional-N synthesizer to generate the radio frequencies, clocks, and data/packet timing. This enables it to operate from any of a multitude of frequency sources. This may either be an external source such as a TCXO or a crystal interfaced directly to the BCM2035.

The default frequency reference setting is 15.36 MHz crystal or TCXO, a widely used frequency reference for 3G handset platforms.

CRYSTAL OSCILLATOR

The BCM2035 can use an external crystal to provide a frequency reference. The recommended configuration for the crystal oscillator including all components is shown in Figure 6.

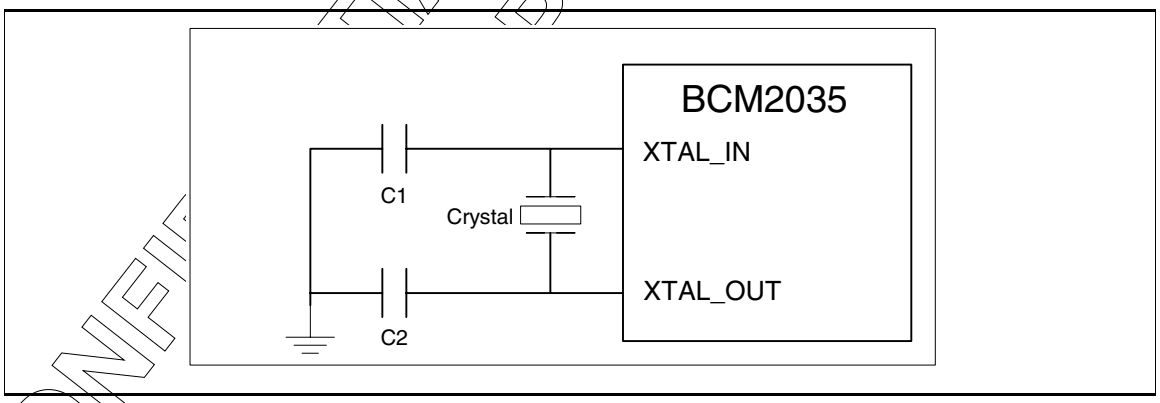

Figure 6: BCM2035 Crystal Oscillator Configuration

External Frequency Reference

An external frequency reference such as those generated by a TCXO signal may be directly connected to the XTAL_IN pin on the BCM2035 as shown in Figure 7.The external frequency reference input is designed to not change the loading on the TCXO when the BCM2035 is powered up or powered down.

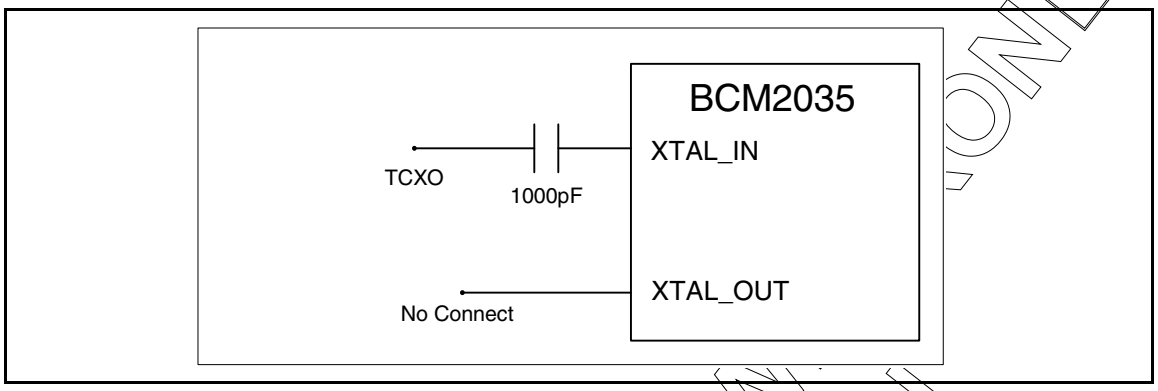

Figure 7: BCM2035 TCXO Configuration

Low Power Oscillator

The BCM2035 uses the LPO clock to provide low power mode timing for park, hold, and sniff modes of operation. The LPO clock can either be provided externally to the device from a 32.768 kHz source or the BCM2035 can operate using the internal LPO clock.

The accuracy of the internal LPO will limit the maximum park, hold, and sniff intervals.

An optional external resistor (120k, 5%) can be used with the BCM2035 to increase the accuracy of the internal LPO.

LPO Clock Interface

The BCM2035 can be configured for use with an external 32.768 kHz oscillator for low power operation. The 32.768 kHz clock should be connected to the LPO_IN/LPO_RES pin that can accept an analog or digital LPO clock. The LPO_IN/LPO_RES provides a high impedance input and will not load the oscillator even when the BCM2035 is powered down. Figure 8 shows the connectivity of BCM2035 to a 32.768 kHz oscillator in external LPO mode.

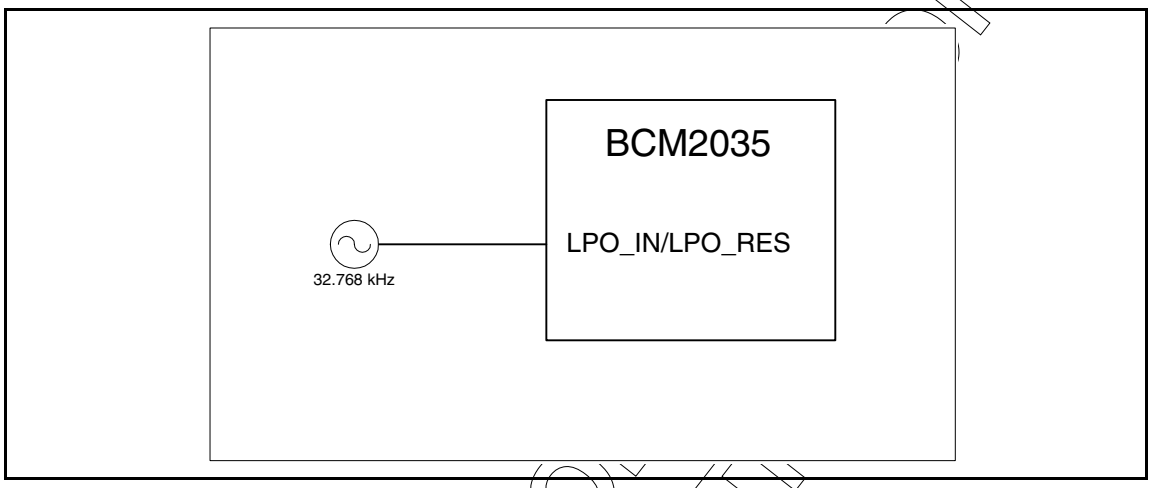

Figure 8: BCM2035 Connectivity to 32.768 kHz Oscillator in External LPO Mode

The BCM2035 can be configured to use the internal LPO for low power mode operation.

Figure 9 shows the proper connection for internal LPO mode.

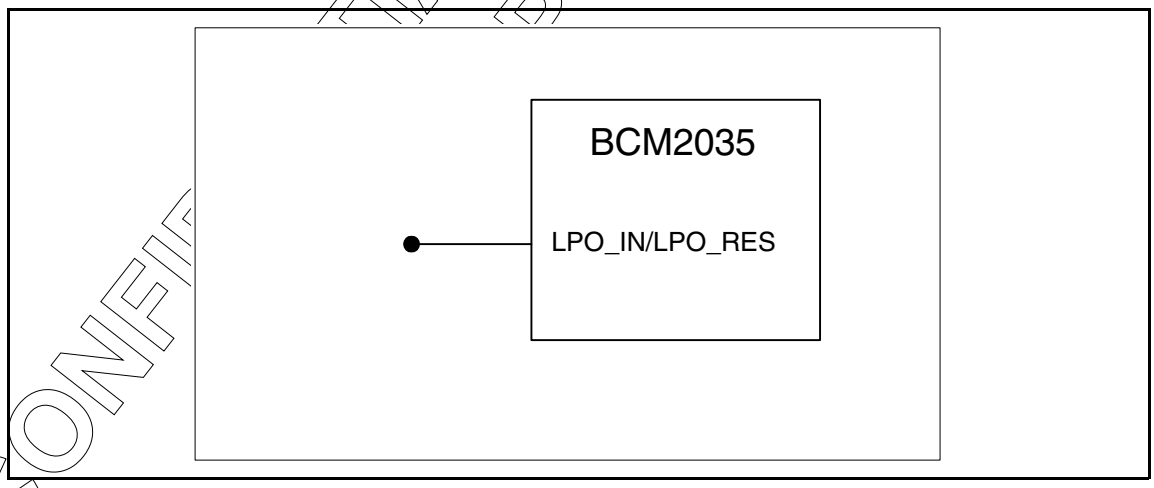

Figure 9: BCM2035 Connectivity For Internal LPO Mode

TRANSPORT CONFIGURATION

The BCM2035 supports the HCI UART and HCI USB transports, and either one may be selected in a particular design. The transport selection is mutually exclusive, so it is not possible to use both USB and UART in a single application. The selection of USB or UART as the HCI transport is sensed by the BCM2035 Boot ROM after a hardware reset has occurred. No special part number, HCI command, or configuration setting is needed to select the transport.

To select UART as the transport, the BCM2035 USB and UART signals should be connected as shown in Figure 10. Connecting the USB signals to GND completely disables and powers off the USB transceiver, eliminating excess current draw from the USB transceiver in UART designs. UP_RX must also be connected to GND. The BCM2035 will detect UART as the transport when UART_CTS_N is driven LOW by the Host.

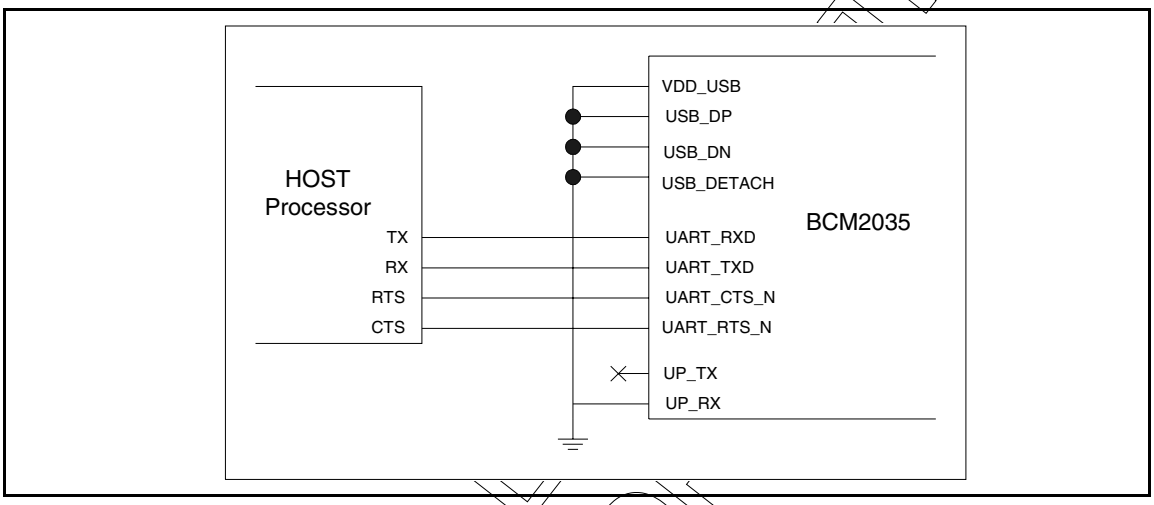

Figure 10: BCM2035 UART Transport Configuration

Figure 11 shows the proper connection for using the BCM2035 with the USB transport. UART_CTS_N must be strapped to VDD to prevent the BCM2035 from detecting activity on the CTS signal. The UART_RXD pin should be connected to GND to prevent it from floating. The UP_RX signal must also be connected to GND. The BCM2035 will automatically detect activity on the USB transceiver when the device is connected to the USB signals on the Host.

The USB_DETACH signal and the resistor network allow the BCM2035 to control USB enumeration through firmware. This is not necessary in all applications.

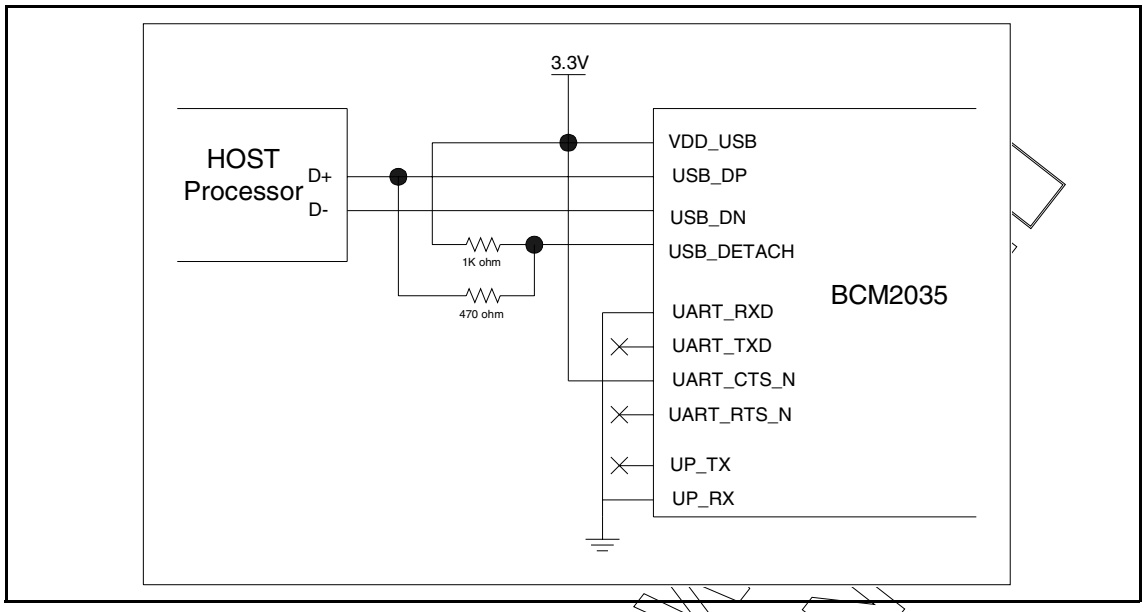

Figure 11: BCM2035 USB Transport Configuration

When using the LCSP package, there is a separate VDD_UART power pin. If UART is not being used, then VDD_UART should be connected to the 1.8V supply to reduce leakage current.

PCM CONFIGURATION

PCM HARDWARE SIGNALS

There are four signals used for the PCM interface; PCM_BCLK, PCM_SYNC, PCM_IN, and PCM_OUT. The PCM_BCLK signal is the bit clock that controls the timing for the other three signals. PCM_SYNC is an 8kHz sync pulse that marks the start of a new PCM frame. PCM_IN and PCM_OUT are the serial PCM data input and output.

The hardware connection between the PCM interface and a CODEC are the same regardless of the configuration of the PCM interface. The connections are shown in Figure 12.

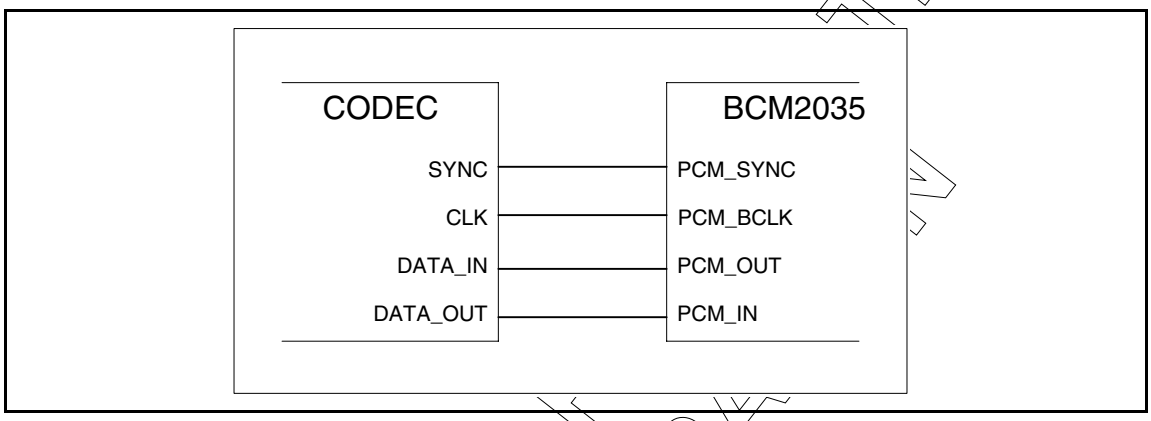

Figure 12: PCM Hardware Connectivity

The PCM interface on the BCM2035 may be configured to operate in Master or Slave mode. In Master Mode, the PCM_SYNC and PCM_BCLK signals are outputs from the BCM2035. In Slave Mode, the PCM_SYNC and PCM_BCLK signals are inputs to the BCM2035 and should to be generated by the external CODEC.

The PCM_BCLK signal supports several clock rates as Master or Slave. Each clock rate allows for a different number of available time slots for multiplexing audio CODEC connections. Table 5 lists the supported clock rates and the number of time slots that are provided by each.

Table 5: Supported PCM Clock Rates

Supported Clock Rates	*Number of Time Slots Provided*
128 kHz	1
256 kHz	2
512 kHz	4
1024 kHz	8
2048 kHz	16

The PCM_SYNC signal supports two different sync formats - Long Sync, and Short Sync. The Long Sync format generates or expects (depending on Master or Slave operation) a rising edge on the PCM_SYNC signal at the beginning of the first bit of the first time slot, and generates or expects the falling edge of the PCM_SYNC signal after the third bit of the first time slot. The Short Sync format generates or expects a one bit wide pulse during the last bit of the last time slot (i.e. the bit just before the first bit of the first time slot).

Detailed timing diagrams for the PCM interface modes are provided in the BCM2035 data sheet.

EEPROM CONFIGURATION

The BCM2035 may store configuration information in an external EEPROM. This is particularly useful in a USB PC application in which it may not be desirable for the HOST to download configuration information each time the BCM2035 is powered up.

Two I/O pins, EEPROM_CLK and EEPROM_DATA, are provided for the connection to an EEPROM that supports I2C serial interface. The EEPROM_DATA signal is open drain and requires a pull-up resistor, whether or not an EEPROM is present. The EEPROM_CLK signal is an output from the BCM2035 that controls the clock for the serial EEPROM.

Figure 13 and Figure 14 on page 30 show the connectivity of BCM2035 pins (EEPROM_CLK and EEPROM_DATA) for system with and without EEPROM respectively.

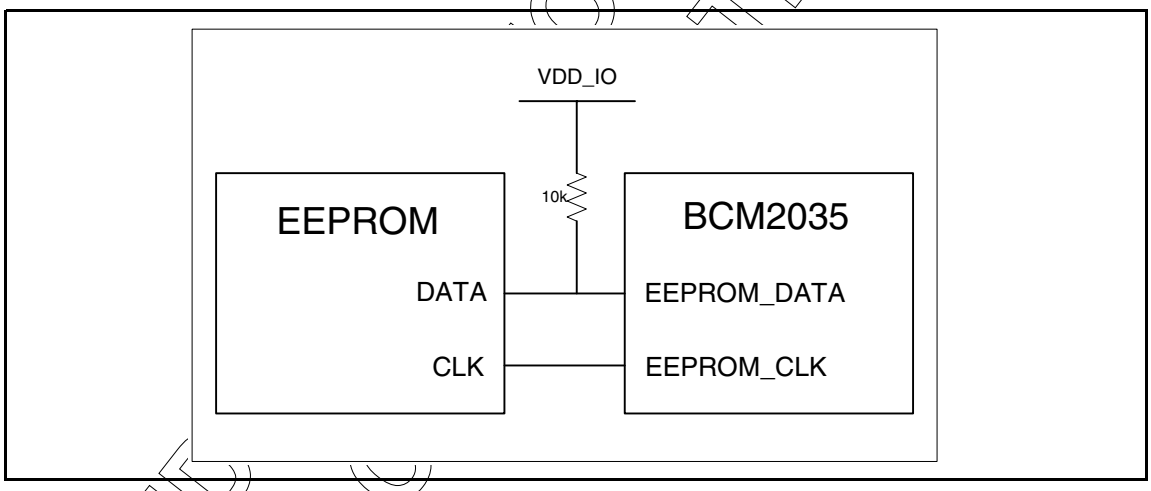

Figure 13: BCM2035 Connection to EEPROM

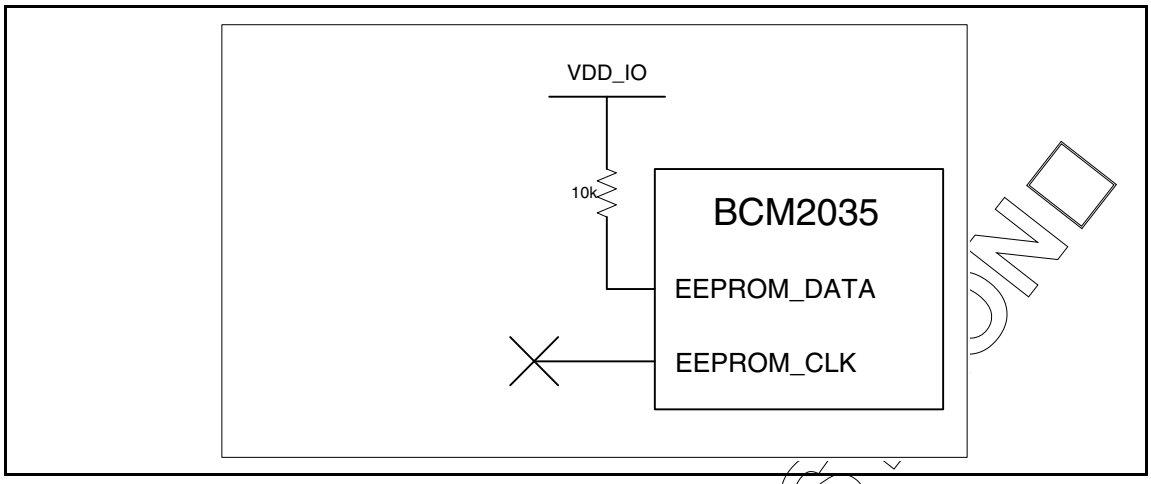

Figure 14: BCM2035 Without EEPROM

ON-CHIP VOLTAGE REGULATOR

The BCM2035 has a full integrated on-chip voltage regulator. The on-chip regulator operates from a 3.3V supply and can generate a 1.8V regulated supply. The on chip regulator can be selectively enabled or disabled via the REG_CTRL pin. Figure 15 shows the configuration with the internal regulator enabled.

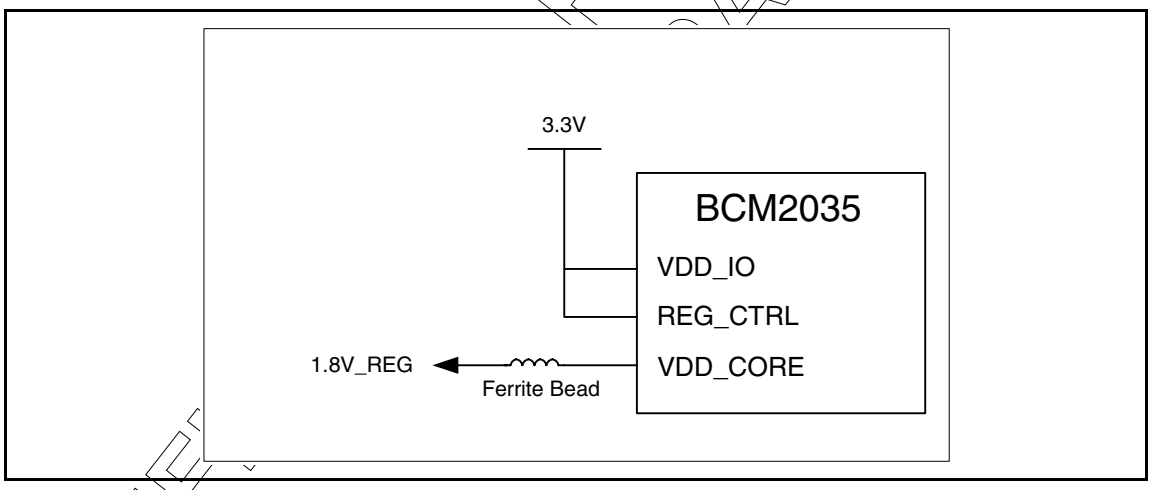

Figure 15: BCM2035 With On-Chip Regulator Enabled

Figure 16 shows the configuration with the internal regulator disabled.

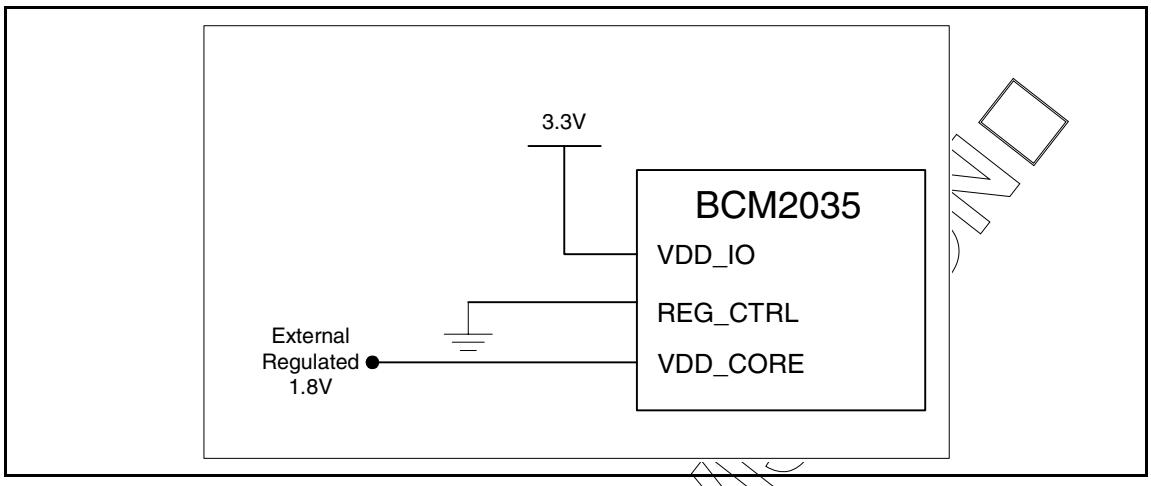

Figure 16: BCM2035 With On-Chip Regulator Disabled

Section 5: Software Configuration

INTRODUCTION

Figure 17 shows the high level power-up sequence for BCM2035. On power-up, the BCM2035 initializes the hardware and begins executing the BOOT code.

The boot code performs RF programming to complete hardware initialization and sets the execution context to the firmware. In the absence of valid RF configuration data (either via hard-wired setting, EEPROM or Flash configuration) the boot code downloads a mini-driver to complete RF initialization.

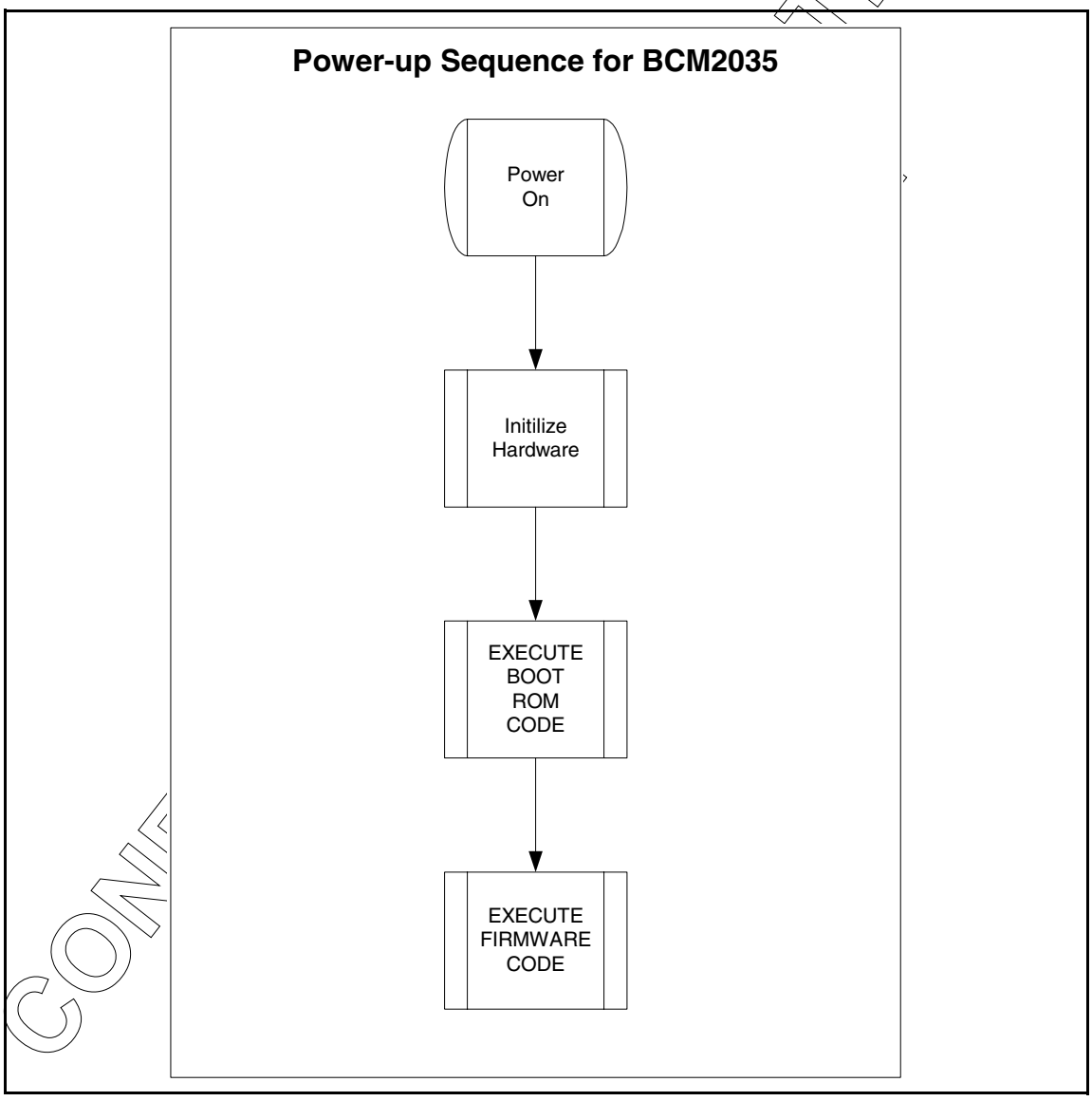

Figure 17: BCM2035 High-Level Power-Up Sequence

HARDWARE POWER-UP SEQUENCE

The BCM2035 contains a Power On Reset circuit. The desired voltage levels are checked for both CORE and I/O power. An LPO clock is required (external or internal) for proper H/W initialization. Without the presence of the LPO clock, the BCM2035 will remain in the state of reset (i.e crystal warm-up period) forever.

The hardware power-up sequence is shown as the Initialize Hardware block in Figure 17 on page 32.

The Power-On Reset occurs when power is applied to the BCM2035, and RESET_N goes from Low to High. The RESET_N signal has an internal Pull-Up, so it may be left unconnected such that the Power-On Reset occurs automatically when power is applied to the chip, or the RESET_N signal may be controlled externally.

The following criteria must be met for the POR to occur successfully:

- If external LPO mode is selected, the LPO must be 32.768 kHz, enabled, and stable.
- If a TCXO or other clock is being used for the main clock, the clock must be enabled. If a crystal is being used with the BCM2035 crystal oscillator circuit, it will start automatically on power-up.
- Correct power levels must be applied to VDD_IO, VDD_CORE, and all RF VDD pins.
- RESET_N must transition from Low to High after power has been applied, or during power-up.

The POR lasts for 192 cycles of the Low Power Oscillator (approximately 32kHz), or approximately 6ms. If the BCM2035 is being used in external LPO mode, and the main clock frequency is to be auto-detected, the main clock must be stable before 192 cycles of the LPO complete.

Toggling the RESET_N signal Low and High at any time after the BCM2035 is powered up will cause the chip to reboot and go through this Power-On Reset sequence.

BOOT ROM POWER-UP SEQUENCE

Figure 18 on page 34 shows the BOOT ROM power-up sequence. The BOOT ROM code is responsible for correctly initializing the system by programming the encoded crystal frequency, downloading mini-drivers, applying boot code patches and launching the firmware or mini-driver (either from FLASH or program ROM).

"Crystal Programming" on page 35 through "Read Flash" on page 37 discuss the three phases of the boot sequence:

1. Crystal Programming
2. Reading EEPROM or Flash for patches and code updates
3. Launching firmware from either FLASH or program-ROM

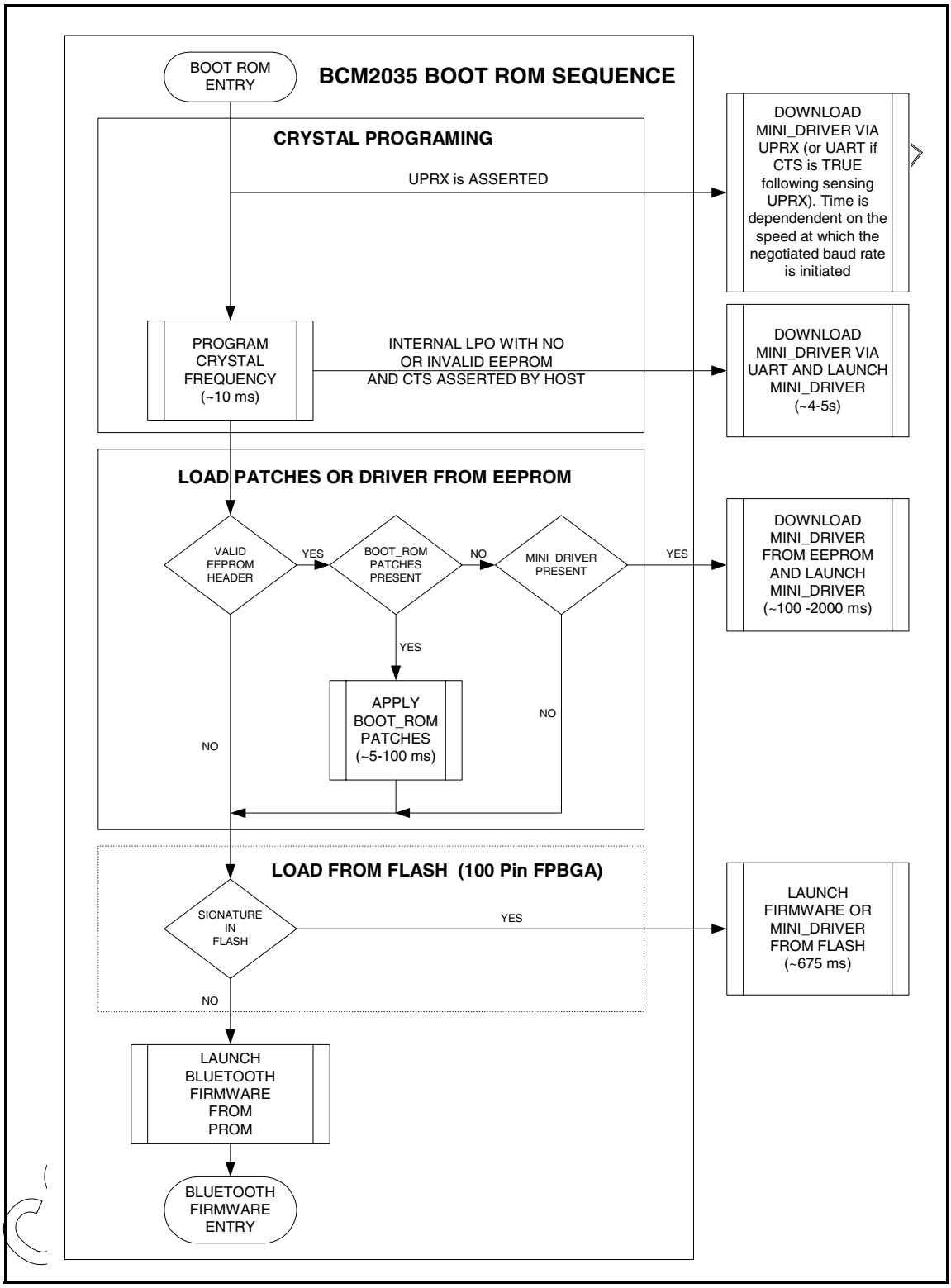

Figure 18: BCM2035 Boot ROM Power-Up Sequence

CRYSTAL PROGRAMMING

The BCM2035 supports several crystal frequency selection modes. Auto crystal frequency detection is supported in the presence of an external LPO clock. List of supported crystal frequencies are shown in Table 6. Crystal frequency settings can also be read from the EEPROM or programmed using a mini driver.

Table 6: 4-bit PLL_SEL Field of CLK_CTL Byte

Binary	*Decimal*	*Crystal Frequency (Mhz)*
0000	0	19.2
0001	1	19.68
0010	2	19.8
0011	3	12
0100	4	13
0101	5	7.68
0110	6	14.4
0111	7	16.8
1000	8	26
1001	9	20
1010	10	Variable/Wildcard
1011	11	18
1100	12	19.44
1101	13	16.2
1110	14	38.4
1111	15	15.36

There are six options for programming the crystal frequency.

1. All modes and UPRX pin is ASSERTED When the boot-rom starts if UPRX is ASSERTED a baud rate of up to 57600bps can be negotiated (using the character 0x55) for the purpose of downloading an encoded-crystal-frequency-programming or *non-volatile-memory-programming-mini-driver. Downloading over UPRX is the same as is described in 4 below without the use of any flow control.

Note Non-volatile-memory refers to external serial-EEPROM or FLASH memories.

2. All modes and UPRX pin is ASSERTED and CTS is ASSERTED.

   If the boot rom firmware senses CTS TRUE following sensing UPRX TRUE a baud rate of up to 1.5Mbps (at certain crystal frequencies) can be negotiated (using the 0x55 character) for the purpose of downloading a non-volatile-memory-programming mini-driver. Important: BCM2035 will not transmit RTS until the 0x55 character is sampled and measured.

3. External LPO

   a. In this mode, an external LPO clock is used by the BCM2035 to determine the frequency of the external crystal and complete the crystal programming phase. The external crystal must be one of the supported frequencies show in Table 6 on page 35.

4. Internal LPO with valid serial EEPROM

   a. The Configuration Data Header (stored/found in non-volatile-memory) header contains the encoded crystal-frequency programming information and is used to complete crystal programming. See Section 6: “Configuration Data File System” on page 63 for the 12-Byte Fractional-N information field of EEPROM. The boot code will read the PLL_SEL[3:0] field to determine the crystal frequency as per Table 6 on page 35. If the PLL_SEL = 1010 (Variable/Wildcard) it will use the information in the 12-Byte Fractional-N field for crystal programming information.

5. Internal LPO with Invalid non-volatile-memory using Mini-Driver

   a. In this mode, either the non-volatile-memory is absent, blank or does not have a valid content (header checksum failed). The HOST would download a mini driver via the UART port that could be used to program the non-volatile-memory (driver named eemd, flmd) or program the encoded crystal frequency (rfmd).

   b.The HOST should assert CTS to indicate a baud rate negotiation request should take place and followed by mini-driver download. Negotiation of baud-rate starts when the HOST transmits the character 0x55. (physically 57600ps is the limit over the UPRX port, 1.5mbps is the limit for UART when combined with certain crystal frequencies, logically the boot-rom firmware is currently limited to approximately 330000bps). **Important: The BCM2035 will not transmit RTS until the 0x55 character is sampled, measured and found to be within the limits noted above.** Following reception of the negotiation character from the HOST, the BCM2035 will acknowledge with RTS and two ASCII characters 41 (0x34 & 0x31) at the negotiated baud rate. The host can use this acknowledgement to begin downloading the mini-driver, or an optional micro-driver. Currently, the boot ROM of the BCM2035 only supports downloading using the ASCII Intel hex format. The download time for a 2k mini-driver or a 200 byte micro-driver at the negotiated baud rate ASCII Intel hex format is around 2.7 seconds and 325 milliseconds, respectively. The boot ROM acknowledges every HEX record from the host with either an '.' (0x2E) character for successful reception or 'x' (0x78) character for negative acknowledgement. The ACK's and NACK's are sent to the host at the negotiated baud rate baud. The mini-driver should be used to properly program the crystal frequency and/or update non-volatile memory to correctly complete encoded crystal frequency programming on subsequent boot up.

   Following reception of a nack the host should consider the BCM2035 will reset (max time = 1second) and the process of negotiating for baud rate and subsequent mini-driver download should be restarted.

6. Internal LPO with Invalid non-volatile-memory forcing crystal frequency at 15.36Mhz.

   a. In this mode, either the non-volatile-memory is absent, blank or does not have a valid content (header checksum failed). This option is useful to program the non-volatile-memory during manufacturing.

   b. An external frequency of 15.36 Mhz should be applied at the crystal input.

   c. The HOST is required to de-assert CTS during the BOOT sequence allowing the firmware to launch normally.

   d. Vendor-specific command HCIDownLoadMiniDriver (0x2E) can be used to download mini driver either via the UART or USB I/F.

READ EEPROM

An BSC EEPROM is required for storage of BlueRF encoded crystal frequency data. The EEPROM will be connected to the 8051 microprocessor unit via GPIO5 and GPIO6. GPIO6 will provide the 8051's serial data interface to the EEPROM (requires 10K Ω pull-up no longer necessary as of BCM2035 with the 0-detecting boot and bluetooth ROM-firmware yet remains to be tested for BCM2035). GPIO5 will provide the 8051's serial clock interface to the EEPROM. The EEPROM should be hardware configured for slave address 0 (VCC=3.3V) on the serial clock and data bus and should have a minimum capacity of seventy-two bits. A larger EEPROM will be required to store boot code patches and configuration data.

The EEPROM header is read by the boot code and the header checksum is validated. A failed header checksum or a blank EEPROM will cause the boot code to bypass reading the EEPROM for boot code patches or mini driver code.

Figure 18: "BCM2035 Boot ROM Power-Up Sequence," on page 34 shows the EEPROM read sequence used to apply the boot code patches or for downloading the mini driver. The patches or mini driver should be stored in the EEPROM using the File system format specified in the EEPROM layout section.

READ FLASH

A 1 Mbit FLASH device is required for storage of BlueRF fractional-N data. The flash will be connected to BCM2035 via the external memory bus. The flash should be selected via the BCM2035's CS0 pin.

After determining the fractional-N values (either by measuring the frequency dynamically or by reading the non-volatile memory, the BCM2035 will determine if it will boot from flash or not. First, it inspects the flash for a 7-byte signature. If the signature is not found, the BCM2035 will not attempt to boot from flash and will boot from internal program ROM (PROM). Otherwise, if the signature is found, the BCM2035 will attempt to boot from flash.

FIRMWARE POWER-UP SEQUENCE

The firmware is responsible for initializing the processors, BT baseband, RF and HCI subsystems. It will also read the serial EEPROM or FLASH for configuration data and to load any firmware patches. Once the configuration data is successfully read, it determines the transport type (UART or USB) and goes to normal operation. A vendor specific command HCIDownLoadMiniDriver (0x2E) is used to download any mini drivers over the UART or USB transport. The firmware power-up sequence is shown in Figure 19 on page 38.

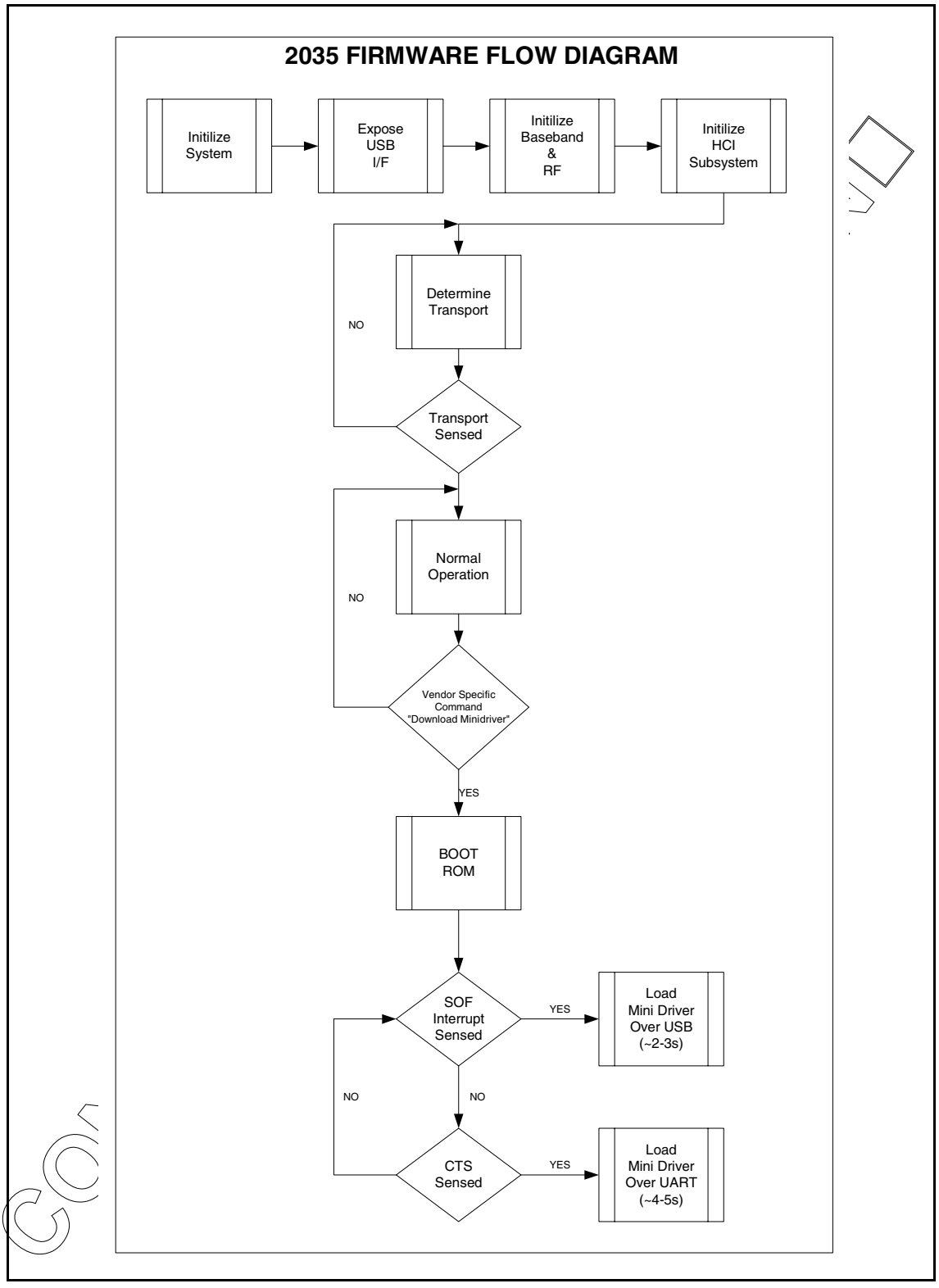

Figure 19: BCM2035 Firmware Power-Up Sequence

BCM2035 MINI-DRIVERS

Mini-drivers serve several purposes such as updating EEPROM, FLASH, changing baud rates, updating run-time configuration data, etc. Depending on the purpose, the mini-driver is downloaded during different stages of the device operation. The device should be setup to accept a mini-driver download. The procedure to setup the device to accept a mini-driver download depends on the device configuration (External or Internal LPO, XTAL Frequency, etc.), execution context of the device (Boot ROM or Firmware) and the type of transport (UART or USB). The ASCII HEX File download protocol (see "ASCII HEX File Download Protocol" on page 92) is used by the HOST to download a mini-driver image to the Host Controller. Table 7 list all the mini-drivers that exist for BCM2035.

After the mini-driver has been downloaded, the Host communicates with the mini-driver. The exact nature of communication between the Host and the mini-driver running on the Host Controller depends on the purpose of the mini-driver. Please refer to the release notes associated with each mini-driver for more detailed information.

For example, the rmmd_400_usb mini-driver is used with USB transport to update run time memory with configuration data. After the mini-driver has been successfully downloaded, the Host will download the configuration data image using the configuration data download protocol described in "ASCII Configuration Data Image Download Protocol" on page 100.

Table 7: BCM2035 Mini-Drivers

Name	*Part*	*Flash*	*EEPROM*	*RAM*	*RF*	*UART*	*USB*	*UPRX*
Flash								
flmd_400_uprx_autobaud	BCM2035	X						X
flmd_400_uart_autobaud	BCM2035	X				X		
flmd_400_usb	BCM2035	X					X	
EEPROM								
eemd_400_uprxWriter_autobaud	BCM2035		X (write)					X
eemd_400_uprxReader_autobaud	BCM2035		X (write)					X
eemd_400_usbWriter	BCM2035		X (read)				X	
eemd_400_usbReader	BCM2035		X (read)				X	
eemd_400_uartWriter_autobaud	BCM2035		X (write)			X		
eemd_400_uartReader_autobaud	BCM2035		X (read)			X		
Run-Time Memory								
rmmd_400_uart_autobaud	BCM2035			X		X		
rmmd_400_usb	BCM2035			X			X	
Crystal Programming								
rfmd_400_XXXX (where XXXX = crystal frequency…e.g. 1536 = 15.36Mhz)	BCM2035				X	N/A	N/A	N/A

LOADING MINI-DRIVERS FROM USB

Instructions for loading mini-drivers from the USB port are listed below:

1. Download mini-driver

   a. The BCM2035 will enumerate with a VID/PID of 0x0A5C/0x2035.

   b. Transmit the mini-driver to the BCM2035 in Intel Hex format using the Bulk Out Endpoint in Interface 0. Line feeds and carriage returns can be eliminated if desired. The HEX records are sent one at a time. The USB handshake packet that is used to terminate a USB transaction also acts as the BCM2035 acknowledgement for receiving the HEX record. The BCM2035 does not have an acknowledgement scheme to indicate if the HEX record was received correctly or not. The mini-driver HEX records should have valid checksum and are assumed to be correct. If an incorrect checksum is found, the BCM2035 will reset itself. The maximum transfer size for the write operations to the Bulk Out Endpoint is 4096 bytes.

   c. When the mini-driver download has completed, program execution is switched to it.

LOAD MINI-DRIVERS FROM UART

Instructions for loading mini-drivers from UART are listed below:

1. Initialize Serial Port

   a. Set UART baud rate to the HCI system baud-rate. RTS/CTS flow control.

   If optional micro-driver is used continue with Step 2, otherwise skip to step 3.

2. Download micro-driver

   a. Transmit the micro-driver to the BCM2035 in Intel Hex format. Line feeds and carriage returns can be eliminated if desired.

   b. After each complete hex record is received by the BCM2035, it will send a '.' (0x2E) character as an ACK.

   c. When the micro-driver download has completed, program execution is switched to it.

3. Download mini-driver

   a. Set the UART for 921.6 Kbps, keep RTS/CTS flow control.

   b. Transmit the mini-driver to the BCM2035 in Intel Hex format. Line feeds and carriage returns can be eliminated if desired.

   c. After each complete hex record is received by the BCM2035, it will send a '.' (0x2E) character as an ACK.

   d. When the mini-driver download has completed, program execution is switched to it.

The micro-driver, if used, determines the UART baud rate that is going to be used for the subsequent mini-driver download. Typically the baud rates set by the micro- and mini-drivers will be the same. The max baud rate that the BCM2035 can support is 1.5M. The downloading of actual firmware using 1.5M baud rate based on ASCII hex format is around 4.5 seconds.

To ensure that all the different stages of the download procedure can be achieved with efficiency; many acknowledgements are sent from the BCM2035 to the host. The host uses these returned characters as indications to transition from one stage to the next. Figure 20 illustrates the downloading protocol between the host and the BT device for ASCII Intel hex formats.

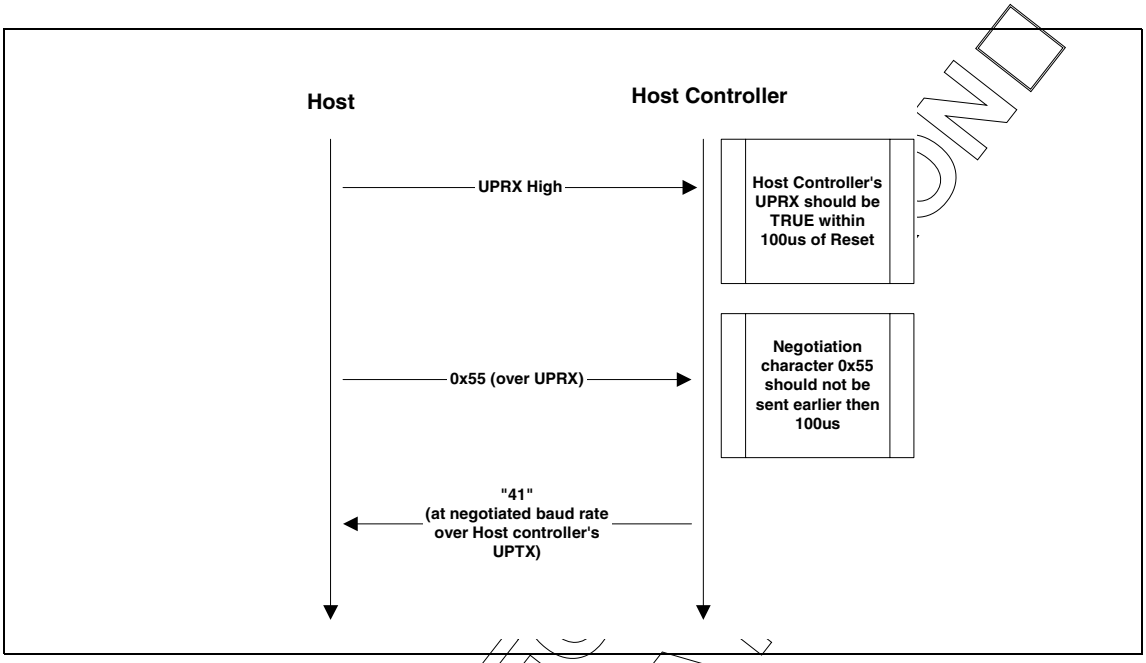

Figure 20: Host and Bluetooth Device Handshaking: Negotiate Baud Rate and Download Mini-Driver Over UPRX

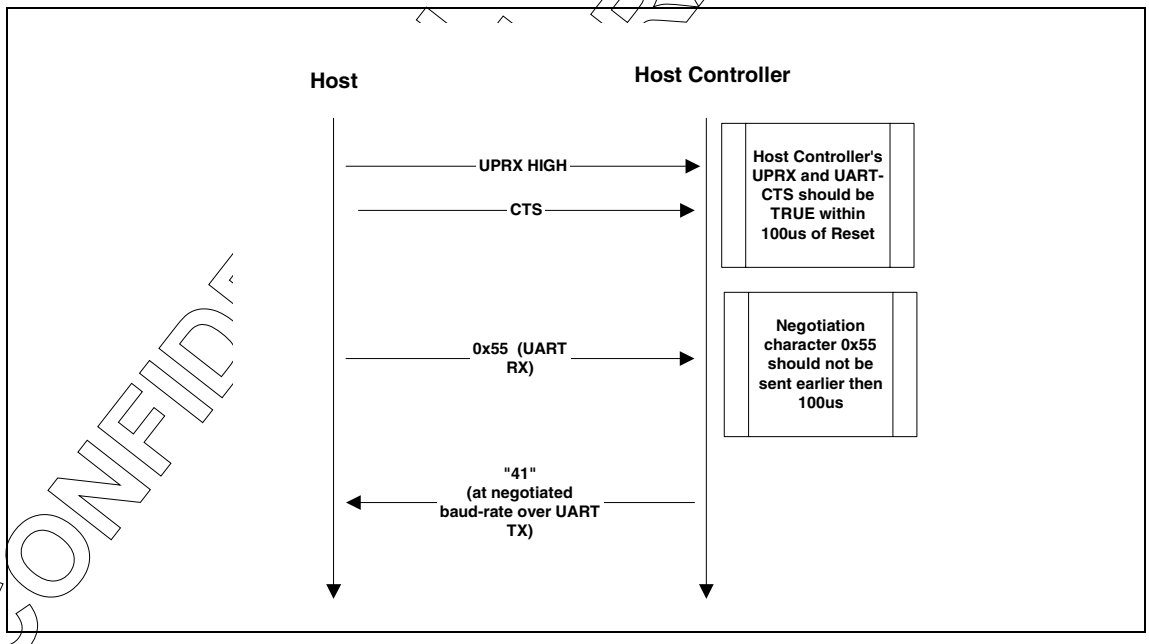

Figure 21: Host and Bluetooth Device Handshaking: Negotiate Baud Rate and Download Mini-Driver Over UART

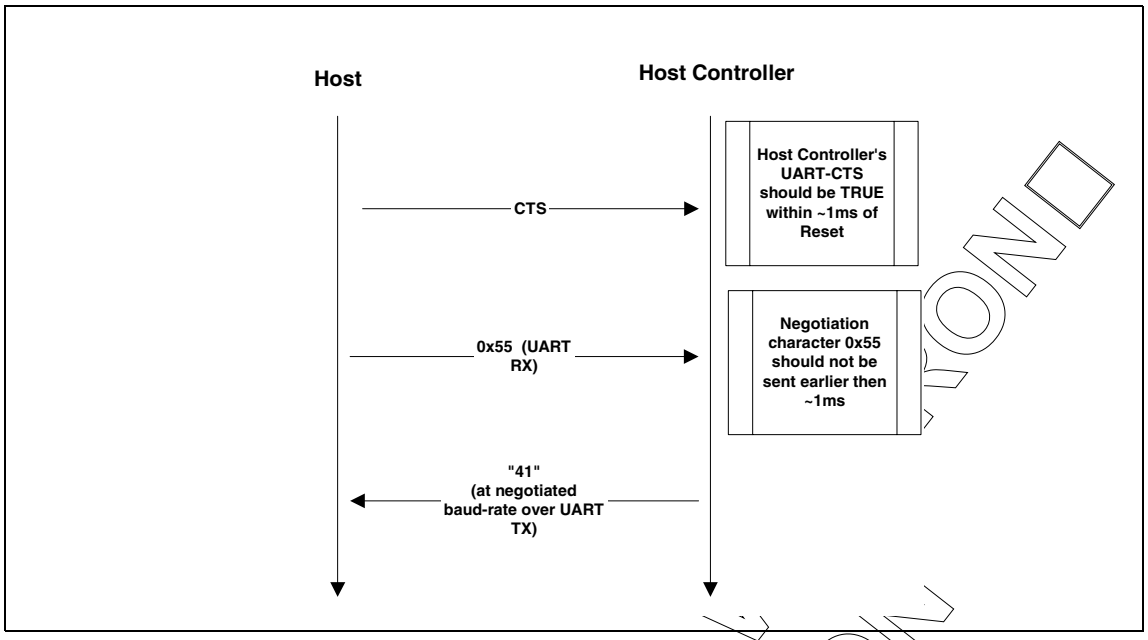

Figure 22: Host and Bluetooth Device Handshaking Prologue: Mode is 0 and Configuration Data is Missing or Invalid. Negotiate Baud Rate and Download Mini-Driver Over UART

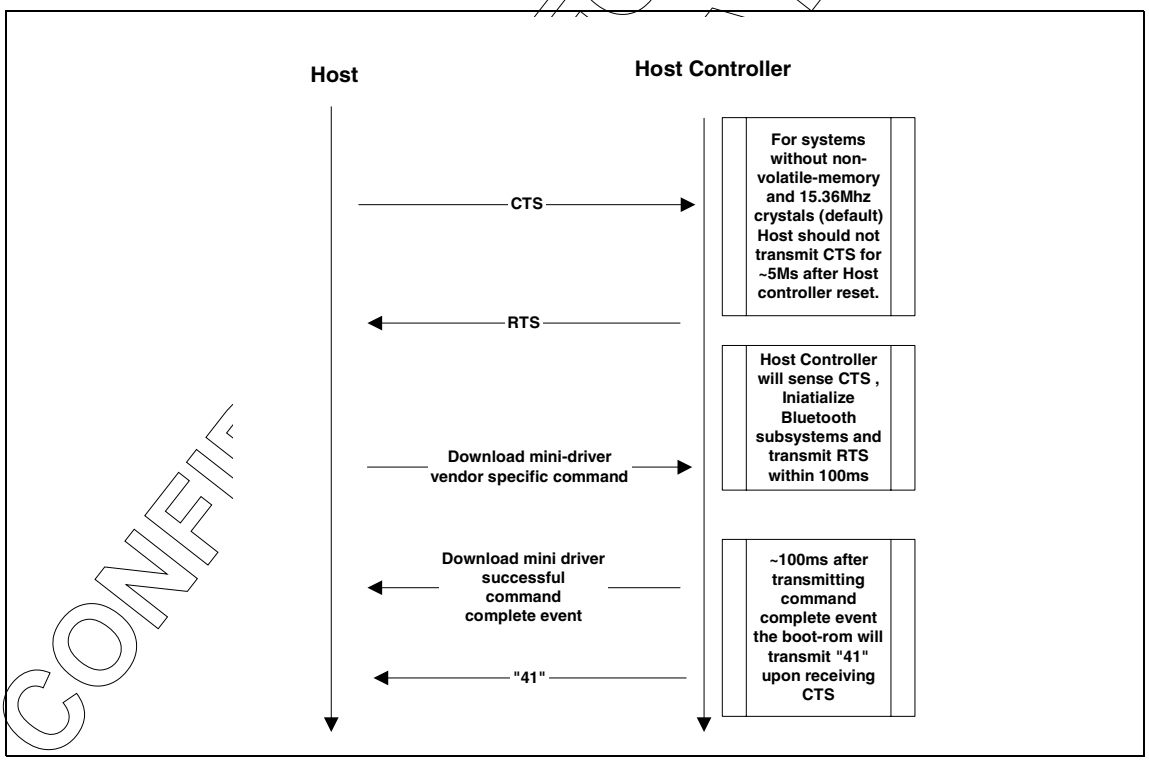

Figure 23: Host and Bluetooth Device Handshaking Prologue: Following Bluetooth Firmware Start-up

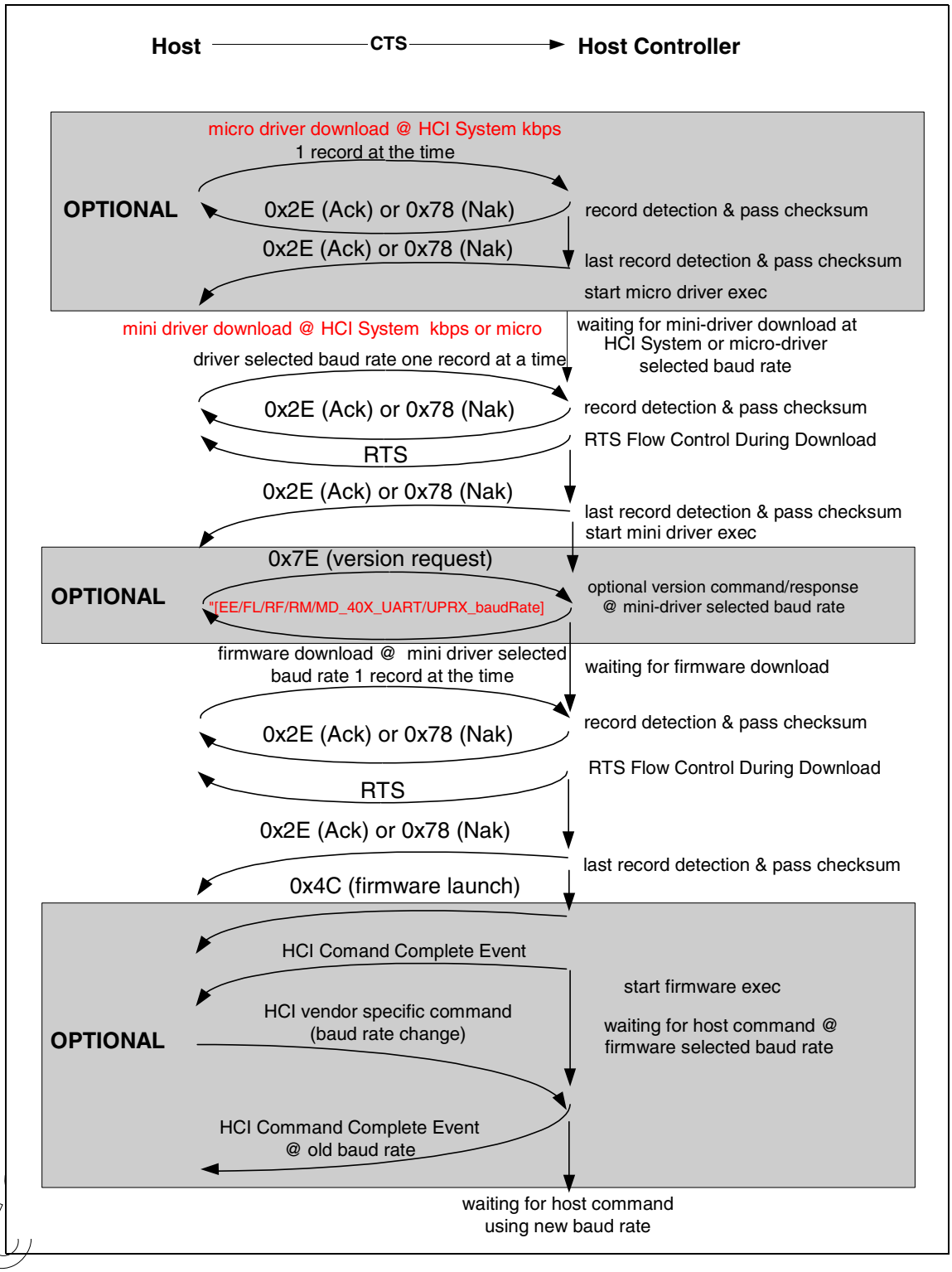

Figure 24: Host and Bluetooth Device Mini-Driver Download Handshaking Epilogue: Flow Diagram Using ASCII Hex Format

The baud rate for downloading the micro- or mini-driver is negotiated at boot-time or read from configuration data. The default firmware baud rate is 115200bps. This setting is selected to ensure that BCM2035 can support all hosts that may or may not have higher baud rate capability. However, the baud rate for downloading the subsequent mini-driver and/or firmware can be modified in both the micro- and/or mini-driver because each application can customize the drivers to allow the fastest transfer rate available between the host and the host controller.

Prior to booting the firmware, the HOST communicates its intent to download a mini-driver by asserting CTS. The BCM2035 will sense CTS assertion by the host and will acknowledge with the ASCII string 41 (0x34 & 0x31). Downloading a micro- or mini-driver before the host controller transmits the acknowledgment characters will cause the boot code to flush the UART FIFOs and re-start.

When each record of the mini-driver is received and passes the checksum, the BCM2035 returns 0x2E (character '.' in ASCII) as an acknowledgement to the host. If a record does not pass the checksum, it returns 0x78 (character 'x' in ASCII) to the host and the boot ROM re-starts. If the host controller receives an ASCII Intel hex record that is not supported, the boot ROM re-starts. The host has to issue an external reset or cycle the power to start the power sequences again. These acknowledgements used for hand shaking between the host and the BCM2035 are fixed in the PROM.

A similar downloading protocol is used by the micro-/mini-drivers as well as when the firmware is downloaded, using either the ASCII Intel hex or binary format. If the binary image format is used, the successful acknowledgment can only be returned after the image is downloaded completely. The hand-shaking protocol can be modified since it resides in the mini-driver.

Figure 25 through Figure 28 on page 48 illustrate the startup, download and reset timing for the ROM and the mini-drivers.

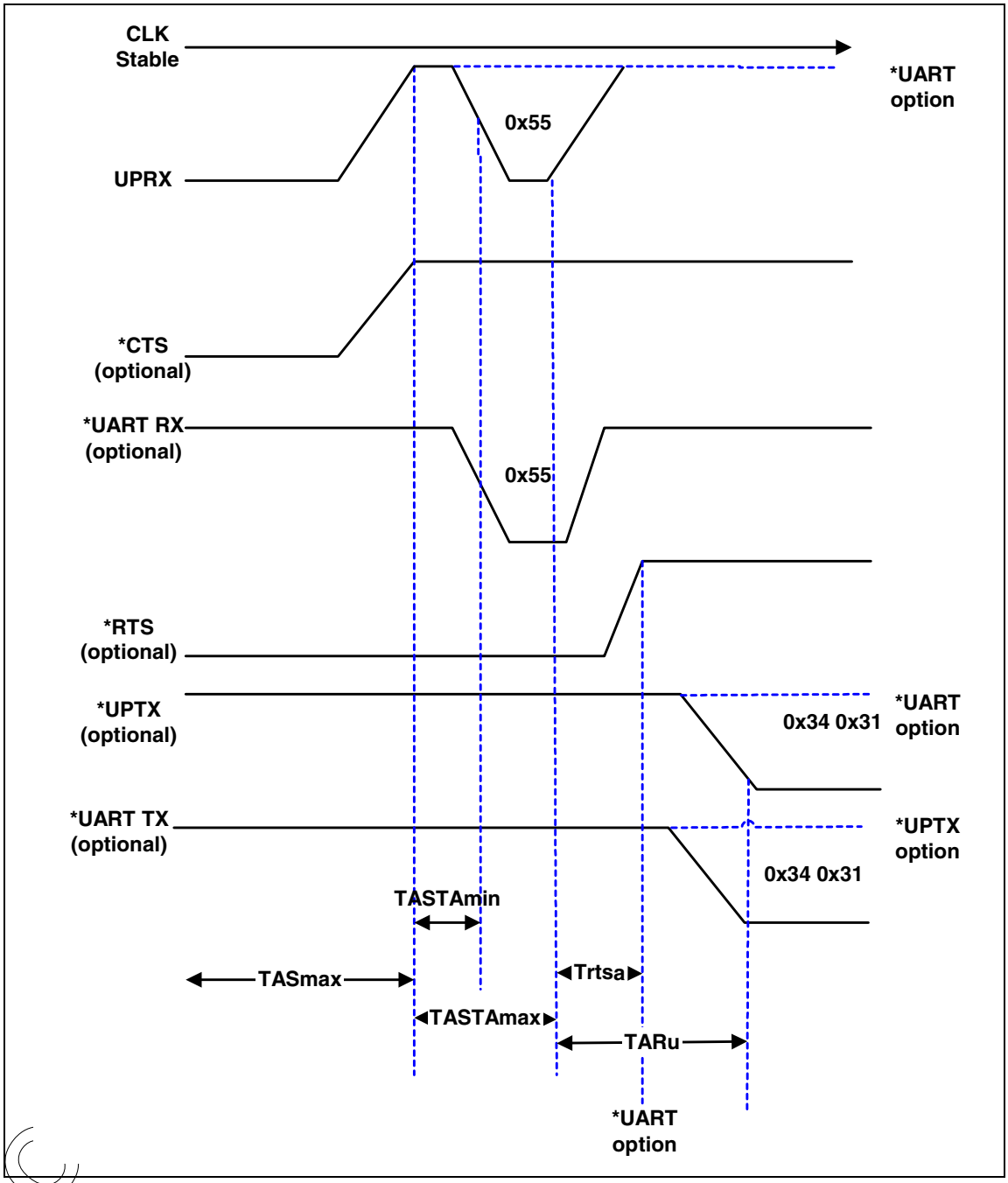

Figure 25: Power-On Clock Stable to Boot ROM (All Modes) Ready for Baud Rate Negotiation and Mini-Driver Download

Table 8: Power-On Clock Stable to Boot ROM (All Modes) Ready for Baud Rate Negotiation and Mini-Driver Download Description Table

Description	Time	Min	Typical	Max
Clock stable to autobaud-signal max	TASmax	-	100 us	-
Autobaud-signal to autobaud-character min	TASTAmin	-	50 us	-
Autobaud-signal to autobaud-character max	Ttac	-	infinity	-
Autobaud character to RTS	Trtsa	-	100 us	-
RTS to autobaud-character response over UART	TARu	-	150 us	-

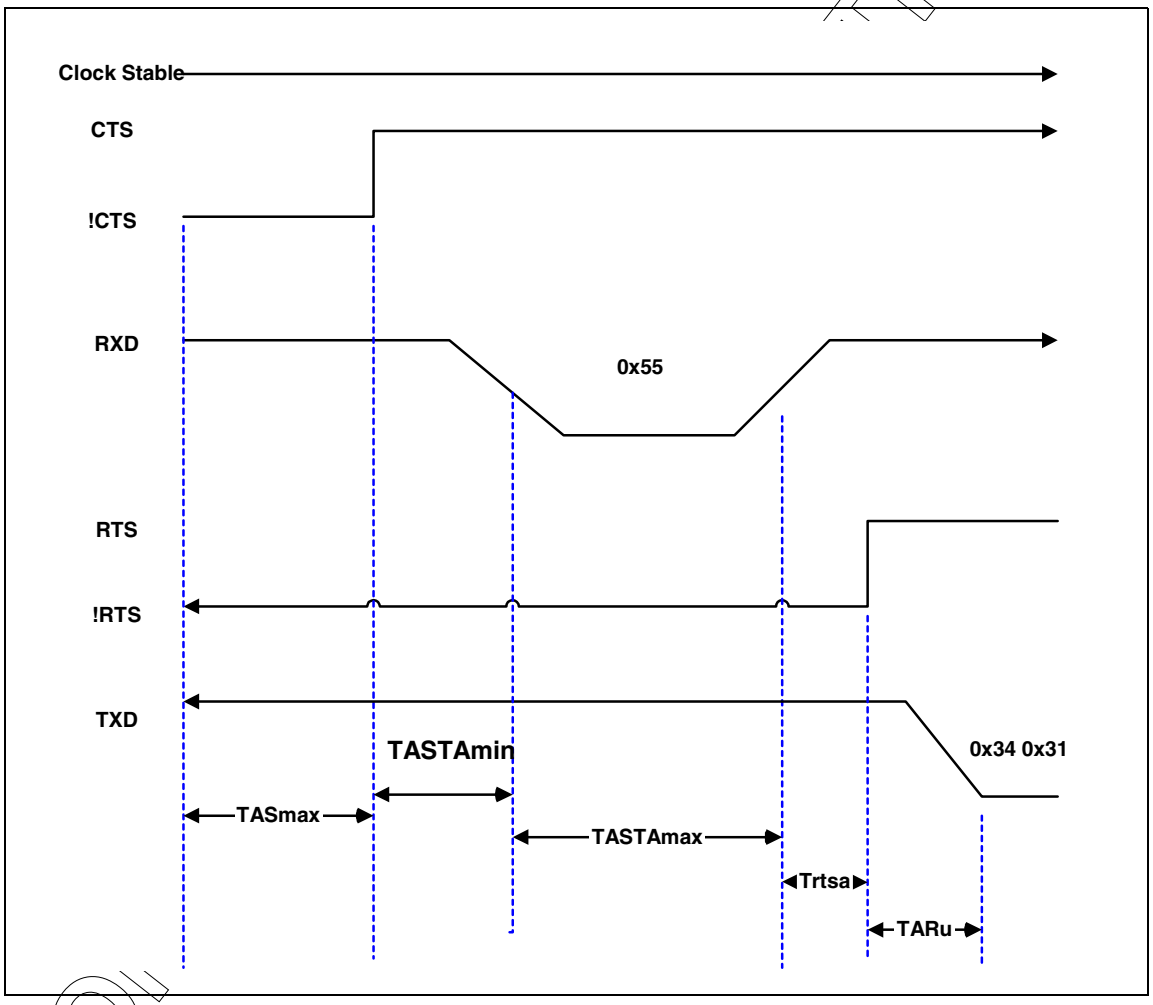

Figure 26: Power-On Clock Stable to Mode-0 Boot ROM Ready for Baud Rate Negotiation and Mini-Driver Download

Table 9: Power-On Clock Stable to Mode-0 Boot ROM Ready for Baud Rate Negotiation and Mini-Driver Download Description Table

Description	Time	Min	Typical	Max
Clock stable to autobaud-signal max	TASmax	-	100 us	-
Autobaud-signal to autobaud-character min	TASTAmin	-	50 us	-
Autobaud-signal to autobaud-character max	Ttac	-	infinity	-
Autobaud character to RTS	Trtsa	-	100 us	-
RTS to autobaud-character response over UART	TARu	-	150 us	-

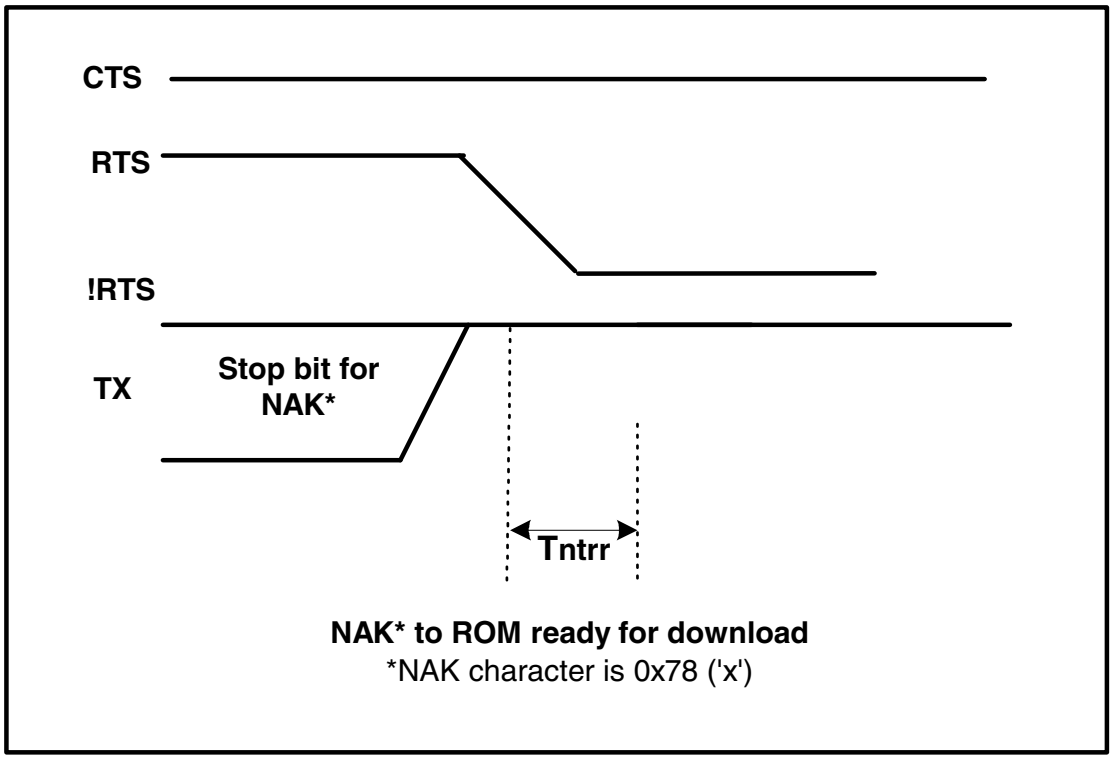

Figure 27: NAK to Boot ROM Read for Baud Rate Negotiation and Mini-Driver Download

Table 10: NAK to Boot ROM Read for Baud Rate Negotiation and Mini-Driver Download Description Table

Description	Time	Min	Typical	Max
NAK to ROM ready for download	Tntrr	-	1.04 s	-

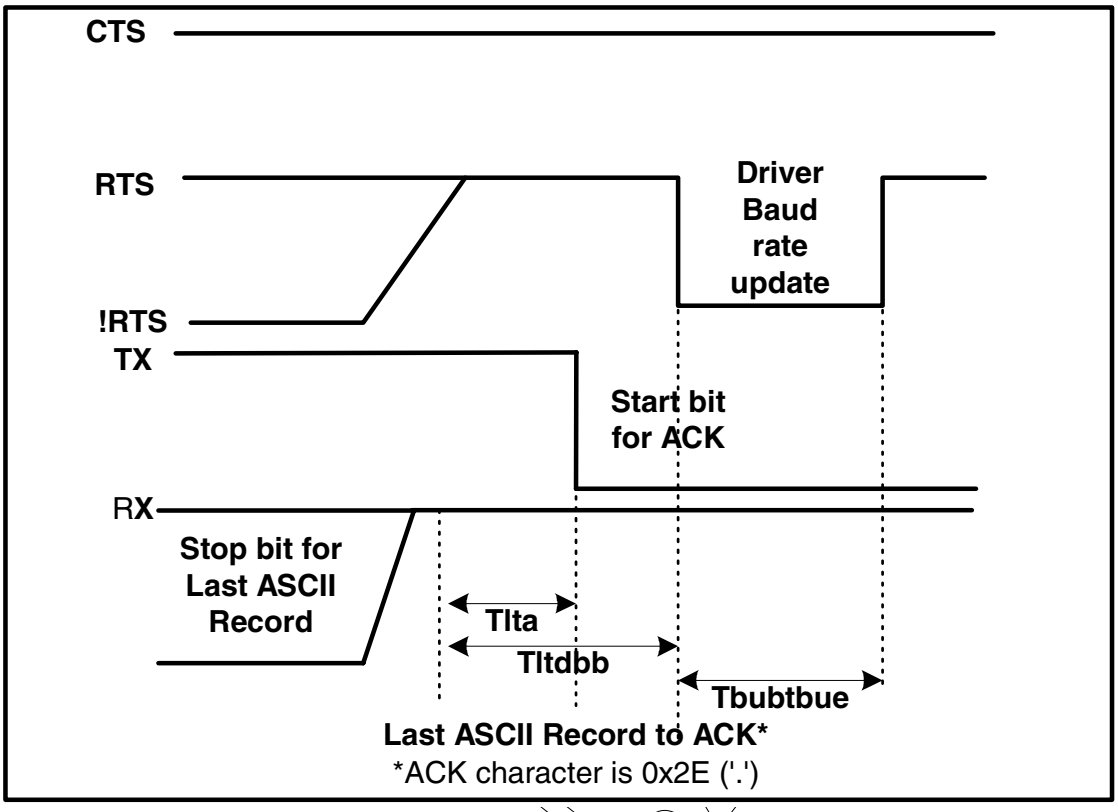

Figure 28: Last ASCII Record to Corresponding ACK

Table 11: Last ASCII Record to Corresponding ACK Description Table

Description	*Time*	*Min*	*Typical*	*Max*
Last ASCII record received to ACK Note: Current ASCII download is limited to ~330 Kbps regardless of baud rate.	Tlta	-	40 us	1.6 ms
Last ASCII record received to driver baud rate change begin.	Tltdbb	-	208 us	-
Driver baud rate update begin to driver baud rate update end.	Tbubtbue	-	800 us	-

SCO CONFIGURATION

INTRODUCTION

The BCM2035 supports Bluetooth SCO connections for phone quality voice applications. HV1, HV2, and HV3 packet types are supported, as are CVSD, a-law, and m-law air coding.

Voice data may be input and output to or from the PCM interface, the UART, or the USB interface using linear PCM, a-law, or m-law audio formats.

The section describe the software requirements needed to support SCO, and any associated Vendor Specific HCI Commands that must be issued to set the correct configuration.

IMPLEMENTING SCO OVER PCM

The two basic requirements to successfully implement SCO connections while using the PCM interface on the BCM2035 are proper hardware connections between the PCM interface and the audio CODEC, and proper configuration of the PCM interface through Vendor Specific HCI commands. The hardware connections between the PCM interface and audio CODEC are described in Section 4.5.1.

Once the configuration is correct, an SCO connection that is added through the use of the standard Add_SCO_Connection HCI command will automatically route audio data to and from the PCM interface.

PCM Data Format

The data formats supported by the PCM Interface are Linear PCM, a-law, and m-law. Linear PCM is a signed 2's complement 16-bit format, and a-law and m-law are 8-bit formats defined in ITU recommendation G.711.

When using a-law or m-law, the 8-bit data may be right or left justified within a 16-bit PCM time slot, and may be shifted Least Significant Bit first or Most Significant Bit first.

When using Linear PCM audio coding with the PCM interface, the data format may be adjusted to allow flexibility interfacing with various CODECs. Each PCM time slot is 16-bits long, but the meaningful PCM data supported by the BCM2035 is always 13-bits. The data may have fill bits inserted to format it as 14, 15, or 16-bits, and the right/left justification may be modified. The fill bits are programmable and may each be set to 1, 0, or a copy of the sign bit. The position of the fill bits depends on the number of fill bits enabled, and the justification selected.

The Linear PCM data may be shifted out of the PCM interface Least Significant Bit first, or Most Significant Bit first. These settings are controlled through Vendor Specific HCI commands discussed throughout this document, and in Appendix 1.

Figure 29 shows the various Linear PCM data formats supported.

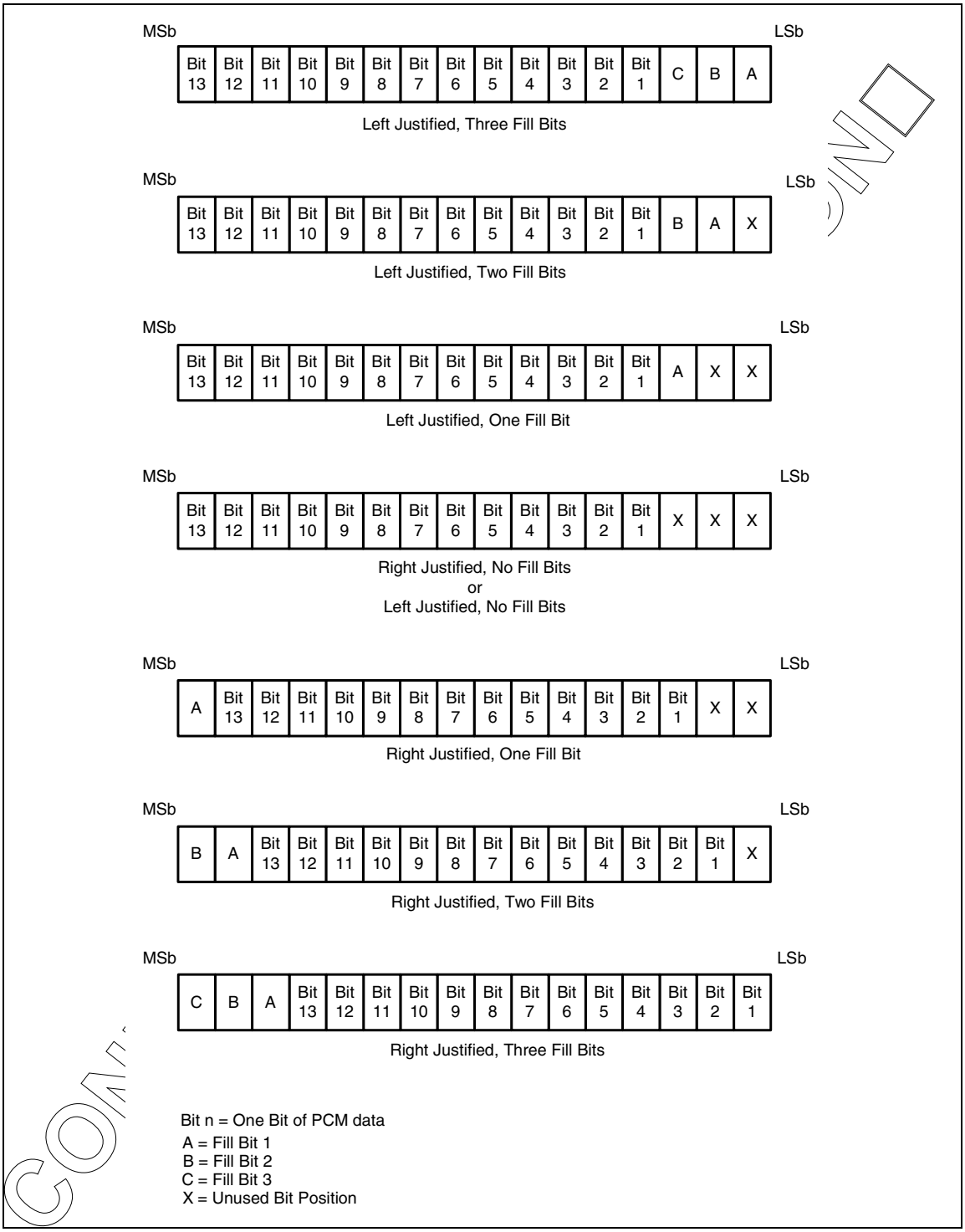

Figure 29: Linear PCM Data Formats

PCM Time Slots Example

Depending on the clock rate that the BCM2035 PCM interface is operating at, there will be multiple PCM time slots available between sync pulses. Figure 30 is an example showing the timing of 4 PCM time slots when using a clock rate of 512kHz in Master Mode with Short Frame Sync.

In this example, two SCO connections are being used, one in time slot 0, and one in time slot 2. During time slots 1 and 3, the BCM2035 PCM_OUT signal is not driving, allowing other devices to share the CODEC and use these time slots.

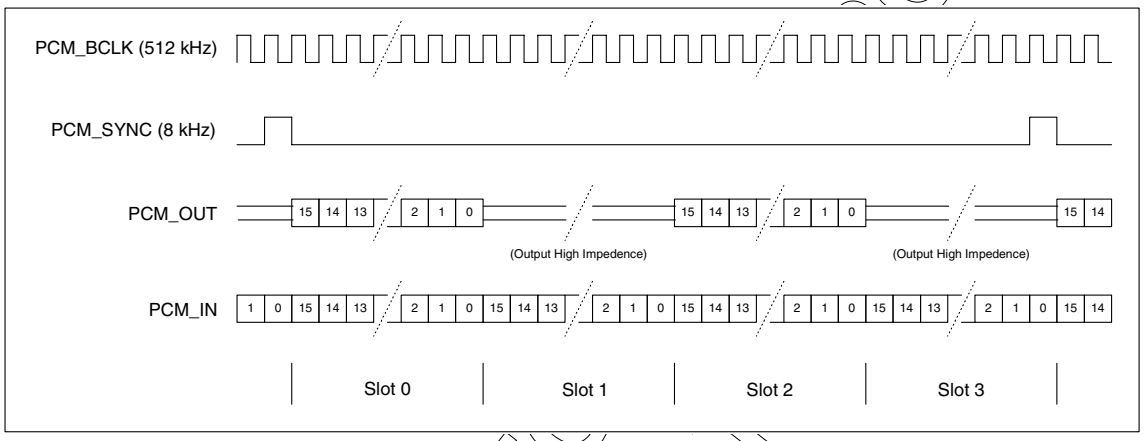

Figure 30: PCM Time Slots Example

Software Configuration of the PCM Interface

The hardware configuration and PCM data formats are configured in the BCM2035 through Standard and Vendor Specific HCI commands. The following HCI commands need to be sent from the Host to properly configure the BCM2035 before adding an SCO connection.

HCI_Write_Voice_Settings (to select the Input and Air Coding formats for all SCO connections.)

HCI_Write_SCO_PCM_Int_Parameters (to configure the PCM interface Master/Slave operation, clock rate, and Frame Sync type.)

HCI_Write_PCM_Data_Format_Parameters (to configure the fill bits and justification of the audio data, if Linear PCM is selected as the input format. Not needed if a-law or u-law are selected as the input format.)

HCI_Write_Comfort_Noise_Parameters (to configure the handling of SCO samples lost over-the-air)

HCI_Write_SCO_TS (to select the PCM time slot to be used for the next SCO connection that is added)

After issuing these commands with the desired parameters, the BCM2035 will be configured for SCO connections. The format of each of these HCI commands is described in detail in Chapter 6.