Freescale Semiconductor
Application Note

AN3361
Rev.5, 01/2015

MPC8321E/MPC8323E PowerQUICC™ II Pro Design Checklist

Freescale Semiconductor, Inc.

This application note describes the generally recommended connections for new designs based on the Freescale PowerQUICC™ II Pro MPC8323E processor family.These devices include the following:

- MPC8321/MPC8321E
- MPC8323/MPC8323E

The design checklist may also apply to bus- or footprint-compatible processors introduced in the future. In addition, it can serve as a useful guide to debugging a newly-designed system by highlighting those areas of a design that merit special attention during initial system startup.

To locate any published errata or updates for this document, refer to the Freescale web site listed on the back cover of this document.

Contents

1 Introduction

This section outlines recommendations to simplify the first phase of design. Before designing a system with a PowerQUICC II Pro device, it is recommended that the designer be familiar with the available documentation, software, models, and tools.

1.1 References

Some of the following reference documents may be available only under a nondisclosure agreement (NDA). Contact your local field applications engineer or sales representative to obtain a copy:

- Collateral
  - *MPC8323E PowerQUICC™ II Pro Integrated Host processor Family Reference Manual* (MPC8323ERM)
  - *MPC8323E PowerQUICC™ II Pro Family Device Errata* (MPC8323ECE)
  - *MPC8323E Integrated Host Processor Hardware Specifications* (MPC8323EEC)
- Available Tools
  - QE Utility Tool
  - UPM Programming Tool
- Models
  - IBIS
  - BSDL

1.2 Device Errata

The device errata documents (MPC8323ECE) describe the latest fixes and workarounds for the MPC8323E family of devices. Before you start a design, be sure to research the most recent errata documents thoroughly.

1.3 QUICC Engine Utility Tool

The QE Utility Tool includes the Pin Mux Tool, QE Driver API Tool, and QE Performance Calculator (license required).

1.4 UPM Programming Tool

The UPM Programming Tool features a GUI for a user-friendly programming interface for programming all three PowerQUICC II Pro UPM machines. The GUI consists of a wave editor, table editor, and report generator. The user can directly edit the waveform or the RAM array. The report generator prints out the UPM RAM array for use in a C program after programming is complete. The UPM Programming Tool is available on the MPC8323E and MPC8321E device product pages of the Freescale web site.

1.5 Available Training

Our third-party partners are part of an extensive Design Alliance Program. Our current training partners are listed on our web site under the Design Alliance Program. Also available are training materials from past Smart Network Developer's Forums and Freescale Technology Forums. These training documents are a valuable resource in understanding the PowerQUICC II Pro.

1.6 Product Revisions

Table 1 lists the product revisions.

Table 1. Product Revisions

Device	Package	SVR (Rev 1.1, Rev 1.3)	PVR (Rev 1.1, Rev 1.3)
MPC8323E	PBGA	0x8062_0011	0x8084_0010
MPC8323	PBGA	0x8063_0011	
MPC8321E	PBGA	0x8066_0011	
MPC8321	PBGA	0x8067_0011	

2 Power

This section provides design considerations for the MPC8323E power supplies, as well as power sequencing. For information on MPC8323E AC and DC electrical specifications and thermal characteristics, refer to the MPC8323E hardware specification (MPC8323EEC). For power sequencing recommendations, refer to Section 2.3, "Power Sequencing."

2.1 Power Supply

The MPC8323EEC lists the recommended and maximum range for each power supply listed in Table 2. OV_{DD} has a noise margin of 300 mV. All other power supplies have a 5 percent margin. No external signals on the MPC8323E are 5-V-tolerant. Note that absolute maximum ratings are stress ratings only. The functional operation at the maximums is not guaranteed. Stresses beyond those listed may affect device reliability or permanently damage the device.

Table 2. Clock Signal Pin Listing

Type	Name	Block
Core	V_{DD}	e300 core voltage
PLL	AV_{DDn}	QUICC Engine PLL power supply e300 PLL power supply System PLL power supply
I/O	GV_{DD}	DDR DRAM I/O supply voltage DDR2 DRAM I/O supply voltage
I/O	OV_{DD0}	PCI, local bus, DUART, system control and power management, I2C, SPI, and JTAG I/O voltage

2.2 Power Dissipation

The MPC8323E hardware specification provides the power dissipation of V_{DD} for various configurations of the coherent system bus (CSB), QUICC Engine block, and the e300 core frequencies. The hardware specification also estimates power dissipation for all the I/O power rails. I/O power highly depends on the application and is an estimate. A full analysis of your board implementation is required to define your I/O power supply needs. The typical V_{DD} power plus I/O power should be used for the thermal solution design.

The junction temperature must not exceed the maximum specified value. The maximum V_{DD} power is the worst case power consumption and should be used for the core power supply design.

2.3 Power Sequencing

One consequence of multiple power supplies is that when power is initially applied, the voltage rails ramp up at different rates. These rates depend on the power supply, the type of load on each power supply, and the way different voltages are derived.

From a system standpoint, if the I/O power supplies ramp before the V_{DD} core supply stabilizes, there may be a period of time before all voltages stabilize when all functional pins, including pure input pins, are driven to a random logic-one or logic-zero state, with possible adverse effects on externally-connected logic. However, after the power is stable, and as long as $\overline{\text{PORESET}}$ is asserted, most I/O pins are tri-stated. To minimize the time that I/O pins are actively driven, apply core voltage before I/O voltage and assert $\overline{\text{PORESET}}$ before the power supplies fully ramp up. The core voltage (Vdd) should reach its nominal level before the I/O voltage (OVdd and GVdd) reaches 0.7 V. In addition, the IO voltage should reach its nominal level within 100 microseconds after the core voltage reaches its nominal level. Excessive current consumption may be observed if these recommendations are not followed.

The hardware specification shows the current maximum ratings for the power supplies. Supplies must not exceed these absolute maximum ratings. However, during normal operation, use the recommended operating conditions tables in the hardware specification.

2.4 Power Planes

Each V_{DD} pin should be provided with a low-impedance path to the board power supply. Similarly, each ground pin should be provided with a low-impedance path to ground. The power supply pins drive distinct groups of logic on-chip. The capacitor leads and associated printed circuit traces connecting to chip V_{DD} and ground should be kept to less than half an inch per capacitor lead.

2.5 Decoupling

Due to large address and data buses and high operating frequencies, the PowerQUICC II Pro can generate transient power surges and high-frequency noise in its power supply, especially while driving large capacitive loads. This noise must be prevented from reaching other components in the PowerQUICC II Pro system, and the PowerQUICC II Pro itself requires a clean, tightly regulated source of power. Therefore, it is recommended that the system designer place at least one decoupling capacitor at each V_{DD}, GV_{DD}, OV_{DD} pin of the PowerQUICC II Pro. These decoupling capacitors should receive their power from separate V_{DD}, GV_{DD}, OV_{DD}, and GND power planes in the PCB, using short traces to minimize inductance. Capacitors can be placed directly under the device on the other side of the board using a standard escape pattern. Others may surround the part on both top and bottom sides of the board.

These capacitors should have a value of 0.01 or 0.1 µF. Only ceramic surface mount technology (SMT) capacitors should be used to minimize lead inductance, preferably 0402 or 0603 sizes. In addition, several bulk storage capacitors should be distributed around the PCB, feeding the V_{DD}, GV_{DD}, and OV_{DD} planes, to enable quick recharging of the smaller chip capacitors. These bulk capacitors should have a low equivalent series resistance (ESR) rating to ensure the quick response time necessary. They should also be connected to the power and ground planes through two vias to minimize inductance. Suggested bulk capacitors: 100–300 µF (AVX TPS tantalum or Sanyo OSCON). Simulation is strongly recommended to minimize noise on the power supplies before proceeding into the PCB design and manufacturing stage of development.

2.6 PLL Power Supply Filtering

Each PowerQUICC II Pro PLL is provided with power through independent power supply pins (AV_{DDn}). The AV_{DD} level should always be equivalent to V_{DD}, and preferably these voltages are derived directly from V_{DD} through a low frequency filter scheme.

There are several ways to provide power reliably to the PLLs, but the recommended solution is four independent filter circuits as illustrated in Figure 1, one to each of the four AV_{DD} pins. Providing independent filters to each PLL reduces the opportunity to cause noise injection from one PLL to the other. This circuit is intended to filter noise in the PLLs resonant frequency range from a 500 kHz to 10 MHz range. It should be built with surface mount capacitors with minimum effective series inductance (ESL). Consistent with the recommendations of Dr. Howard Johnson in *High Speed Digital Design: A Handbook of Black Magic* (Prentice Hall, 1993), multiple small capacitors of equal value are recommended over a single large value capacitor.

To minimize noise coupled from nearby circuits, each circuit should be placed as closely as possible to the specific AV_{DD} pin being supplied. It should be possible to route directly from the capacitors to the AV_{DD} pin, which is on the periphery of package, without the inductance of vias. Figure 1 shows the PLL power supply filter circuit.

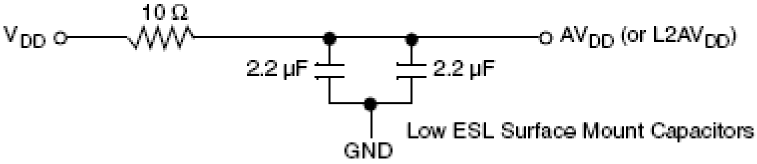


Figure 1. PLL Power Supply Filter Circuit

2.7 Pin Listing and Connections

Table 3 lists the power signal pin connections.

Table 3. Power Signal Pin Listing

Signal	Connection	Notes
AV_{DD1}	1.0 V ± 50 mV	Power for the QUICC Engine PLL
AV_{DD2}	1.0 V ± 50 mV	Power for the System PLL
AV_{DD3}	1.0 V ± 50 mV	Power for the e300 PLL
AV_{DD4}	1.0 V ± 50 mV	—
GV_{DD}	2.5 V ± 125 mV 1.8 ± 90 mV	Power for DDR and DDR2 I/O voltage

Table 3. Power Signal Pin Listing (continued)

Signal	Connection	Notes
V_{DD}	1.0 V ± 50 mV	Power for the core
OV_{DD}	3.3 V ± 330 mV	Power for PCI, local bus, DUART, system control, I2C, QUICC Engine, and JTAG I/O voltage
MVREF[1:2]	0.49 x GV_{DD} to 0.51 x GV_{DD}	DDR I/O reference voltage

3 Clocking

Figure 2 shows the internal distribution of clocks within the MPC8323E. The primary clock source for the MPC8323E can be one of two inputs, CLKIN or PCI_CLK, depending on whether the device is configured in PCI host, PCI agent, or PCI disabled mode.

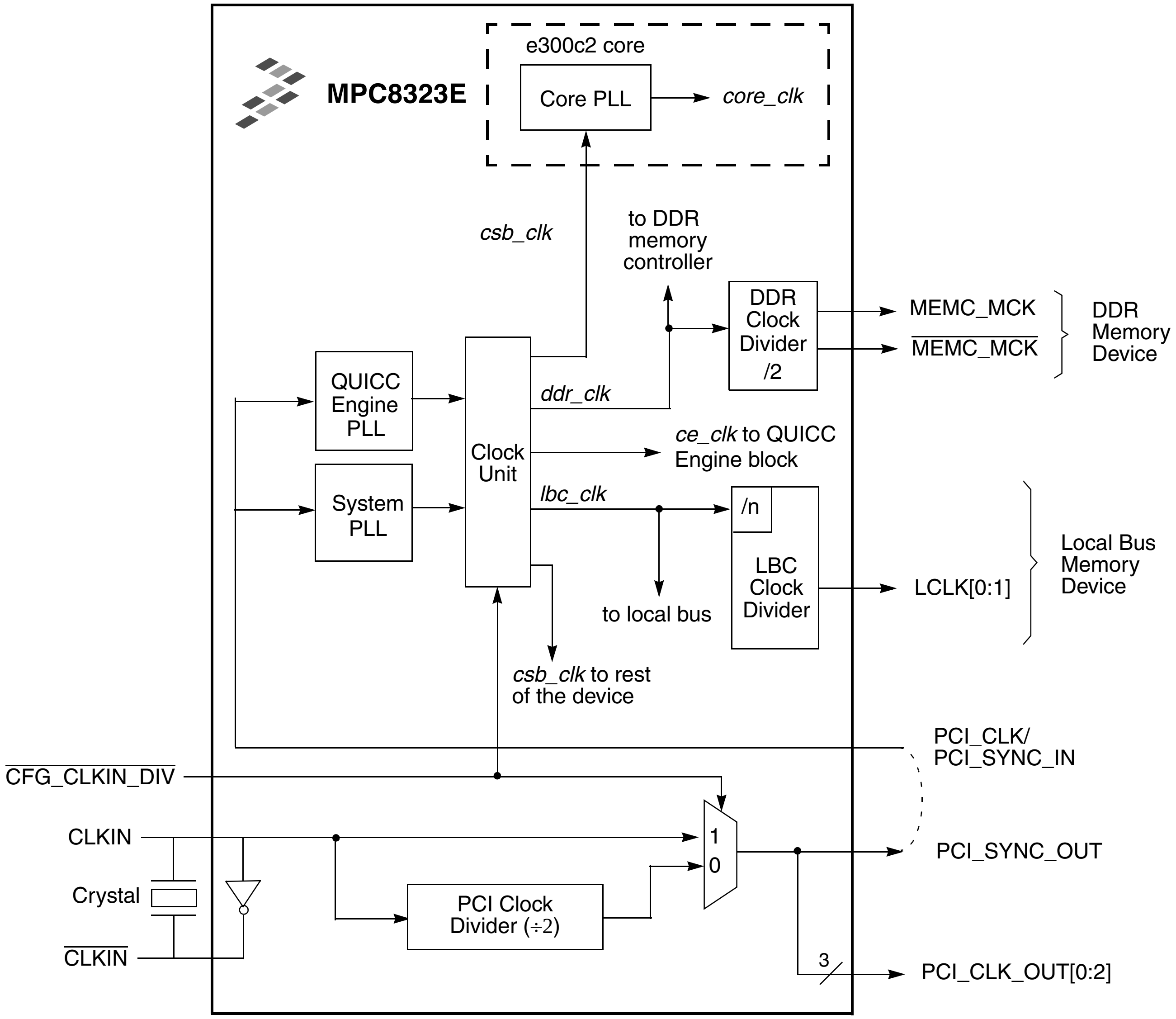


Figure 2. MPC8323E Clock Subsystem

3.1 System Clock in PCI Host Mode

When the MPC8323E is configured as a PCI host device (RCWH[PCIHOST] = 1), CLKIN is its primary input clock. CLKIN feeds the PCI clock divider (÷2) and the PCI_SYNC_OUT and PCI_CLK_OUT multiplexers. PCI_SYNC_OUT is connected externally to PCI_SYNC_IN to allow the internal PCI controller clock to synchronize with the external PCI agent clocks. PCI_SYNC_OUT must be connected properly to PCI_SYNC_IN, with equal delay to all PCI agent devices in the system.

The individual PCI_CLK_OUT[0:2] can be disabled by clearing the OCCR[PCICOEn] bit. For example, if only one PCI clock is needed, then configure OCCR[0:2] = 3’b100. $\overline{\text{CFG\_CLKIN\_DIV}}$ selects whether CLKIN or CLKIN ÷ 2 is driven out on the PCI_SYNC_OUT and PCI_CLK_OUT[0:2] signals. If $\overline{\text{CFG\_CLKIN\_DIV}}$ = 0, then the PCI interface runs at half the CLKIN speed.

3.2 System Clock in PCI Agent Mode

When the MPC8323E is configured as a PCI agent device, PCI_CLK is the primary input clock. In agent mode, the CLKIN signal should be pulled down with a 1 – 4.7 k Ω resistor. PCI_CLK_OUT*n* and PCI_SYNC_OUT are not used. In agent mode, the $\overline{\text{CFG\_CLKIN\_DIV}}$ configuration input can be used to double the internal clock frequencies.

$$CSB\ clock = PCI\_CLK \times (1 + \sim\overline{CFG\_CLKIN\_DIV}) \times RCWH[SPMF]$$

This feature is useful if a fixed internal frequency is desired, regardless of whether the PCI clock runs at 33 or 66 MHz. PCI specifications requires the PCI clock frequency information to be provided by the M66EN signal. If $\overline{\text{CFG\_CLKIN\_DIV}}$ = M66EN and PCI_CLK is 33 MHz, then CSB speed is doubled.

3.3 System Clock if PCI is Disabled

If the PCI interface is not used, PCI_CLK is the primary input clock. CLKIN should be tied to GND, and $\overline{\text{CFG\_CLKIN\_DIV}}$ should be tied high. Table 4 summarizes the clock signal pins.

Table 4. Clock Signal Pin Listing

Signal	Pin Type	Connection		Notes
		If Used	If Not Used	
PCI_CLK_OUT[0:2]	O	As needed[1]	Open	• **Device as PCI Host:** Functions as PCI output clock banks. Program the Output Clock Control Register (OCCR) to enable only the output clocks needed and disable unused clocks in order to reduce EMI. • **Device as PCI Agent or PCI Disabled:** These signals are not used.
PCI_SYNC_IN/ PCI_CLK	I	Connect to PCI_SYNC_OUT or 25–66 MHz clock	**Not applicable.** This pin should always be connected, see Notes.	• **Device as PCI Host (PCI_CLK_OUTx signals driven):** Functions as PCI_SYNC_IN. Connect externally to PCI_SYNC_OUT • **Device as PCI Host (PCI_CLK_OUTx signals not driven), PCI Agent, or PCI Disabled:** Functions as PCI_CLK (primary input clock to the device). A valid 25-66MHz clock signal (at OV_{DD} level) must be applied to this signal.

Table 4. Clock Signal Pin Listing (continued)

Signal	Pin Type	Connection		Notes
		If Used	If Not Used	
PCI_SYNC_OUT	O	Connect to PCI_SYNC_IN	Open	• **Device as PCI Host (PCI_CLK_OUTx signals driven):** Connect externally to PCI_SYNC_IN signal for de-skewing of external PCI clock routing. Loop trace should match with PCI_CLK_OUTx signal traces. • **Device as PCI Host (PCI_CLK_OUTx signals not driven), PCI Agent, or PCI Disabled:** This signal is not used.
CLKIN	I	Connect to 25–66 MHz clock signal	1 k-4.7 k Ω to GND	• **Device as PCI Host:** Functions as primary input clock. A valid 25-66 MHz clock signal (at OV_{DD} level) must be applied to this signal. • **Device as PCI Host (PCI_CLK_OUTx signals not driven), Device as PCI Agent or PCI Disabled:** This signal is not used.

Note:

1. “As needed” terminology is used throughout this document to indicate that the pin should be connected to other devices on the board as required for the application.

4 Power-On Reset and Reset Configurations

A detailed power-on reset flow is described in the “Reset, Clocking, and Initialization” chapter of the *MPC8323E PowerQUICC™ II Pro Integrated Communications Processor Family Reference Manual*.

4.1 Reset Configuration Signals

Various device functions of the MPC8323E are initialized by sampling certain signals during the assertion of the $\overline{\text{PORESET}}$ signal after a stable clock is supplied. These inputs are either pulled high or low during this period. While these pins are generally output pins during normal operation, they are treated as inputs while $\overline{\text{PORESET}}$ is asserted. See Table 5 for termination recommendations for the reset configuration pins.

Table 5. Reset Configuration Pin Listing

Signal	Pin Type	Termination
$\overline{\text{PORESET}}$	I	Driven actively by the external reset logic
$\overline{\text{HRESET}}$	I/O	Pullup with 1 kΩ to OV_{DD}
$\overline{\text{SRESET}}$	I/O	Pullup with 2 - 10 kΩ to OV_{DD}
LGPL0/ CFG_RESET_SOURCE[0]	I/O	Pull-up with 4.7kΩ to OV_{DD} or pulldown with 1kΩ to GND as desired, see Table 6 OR Driven by FPGA as needed during HRESET assertion and tri-state after HRESET negation
LGPL1/ CFG_RESET_SOURCE[1]	I/O	
LGPL3/ CFG_RESET_SOURCE[2]	I/O	
$\overline{\text{CFG\_CLKIN\_DIV}}$	I	Pull-up with 4.7 kΩ to OV_{DD} or pull down with 1kΩ to GND as desired **Note:** For more information, see Table 7.

The CFG_RESET_SOURCE[0:2] input signals are sampled during the assertion of $\overline{\text{PORESET}}$ to specify the interface from which the device is to load the reset configurations words. The words can be loaded from the I^2C interface, from a device on the local bus (CPLD, EEPROM, or FLASH), or from an internally defined word value. Table 6 describes the RCW options. For details, see the *MPC8323E PowerQUICC™ II Pro Integrated Communications Processor Family Reference Manual* (MPC8323ERM).

Table 6. Reset Configuration Word Source

CFG_RESET_SOURCE[0:2] Value (Binary)	Description
000	Reset configuration words are loaded from a device on the local bus
001	Reserved configuration, should not be used.
010	Reset configuration word is loaded from a device on I^2C. PCI_CLK/PCI_SYNC_IN is in the range of 24-66.67MHz.
011 - 111	Hard coded options 0-4.

The $\overline{\text{CFG\_CLKIN\_DIV}}$ input signal is also sampled during the assertion of $\overline{\text{PORESET}}$ to determine the relationship between CLKIN and PCI_SYNC_IN. See Table 7. Notice that the configuration pins are multiplexed with the local bus GPLx signals.

Table 7. CLKIN Divisor Recommendations

$\overline{\text{CFG\_CLKIN\_DIV}}$ Value (Binary)	Description
1	**In PCI host mode:** • PCI runs at CLKIN speed • *csb_clk* = CLKIN × SPMF **In PCI agent or PCI Disabled mode:** • *csb_clk* = PCI_CLK × SPMF
0	**In PCI host mode:** • PCI interface runs at half of CLKIN speed • PCI_CLK_OUT = CLKIN/2 • *csb_clk* = (PCI_SYNC_IN × 2 × SPMF) = CLKIN × SPMF **In PCI agent or PCI Disabled mode:** • *csb_clk* = (PCI_CLK × 2 × SPMF)

4.2 Reset Configuration Words

The reset configuration words control the clock ratios and other basic device functions such as PCI host or agent mode, boot location, and endian mode. The reset configuration words are loaded from the local bus or the I^{2}C interface during the power-on or hard reset flows. If the reset configuration word is from the flash memory, it should reside at the beginning of the flash memory. That is should start from address 0. A total of two 32-bit-words are read. The first byte is read from address 0x0, the second byte from address 0x8, the third byte from address 0x10, and so on until all 8 bytes are read. Bytes b0–b3 form a word, and this is the reset configuration word low register (RCWLR). Bytes b4–b7 form the reset configuration word high register (RCWHR) as follows:

- RCWLR
  0x0000: b0xxxxxx xxxxxxxx
  0x0008: b1xxxxxx xxxxxxxx
  0x0010: b2xxxxxx xxxxxxxx
  0x0018: b3xxxxxx xxxxxxxx
- RCWHR
  0x0020: b4xxxxxx xxxxxxxx
  0x0028: b5xxxxxx xxxxxxxx
  0x0030: b6xxxxxx xxxxxxxx
  0x0038: b7xxxxxx xxxxxxxx

If the reset configuration word is from an I^{2}C device, the I^{2}C setup must comply with the following requirements:

- I^{2}C EEPROM must be connected to I^{2}C.
- EEPROM of extended address type must be used.
- EEPROM must respond to the calling address 0x101_0000.

- Use the special data format as described in the reference manual.

4.2.1 Blank Flash Memory or I^{2}C EEPROM

After power-on reset, the MPC8323E first loads the hard reset configuration word from either the local bus or I^{2}C device depending on the value of the CFG_RESET_SOURCE[0:2] pins.

4.2.1.1 Blank Flash Memory

If the HRCW is in the local bus flash memory, and the flash memory is blank, the MPC8323E loads an invalid HRCW and the PLL cannot lock. When connecting to the processor using an emulator, it will not be possible to communicate with the device through the JTAG port. To overcome this problem, many third-party tools implement a procedure to override the HRCW through JTAG. Contact your tool vendor for further information. To override the HRCW through JTAG with CodeWarrior™, use a JTAG configuration file to change the board reset configuration settings, regardless of the current switches or the values in flash memory. The JTAG configuration file should contain the desired values for the RCWLR and RCWHR registers and specify them in the CCS remote connection settings (Debug Version Settings → Remote Debugging). You can also use the following example file:

```
<CW8.7_Folder>\PowerPC_EABI_Support\Initialization_Files\jtag_chains\832x_HRCW_jtag.txt
```

This file has one line:

```
        E300 (1 1) (2 0x62040083) (3 0x84600000)
```

The meanings of the parameters inside the JTAG configuration file are:

- Config template index 1: Boolean value that turns the overriding on and off
- Config template index 2: Value for Reset Configuration Word Low in the format of RCWLR
- Config template index 3: Value for Reset Configuration Word High in the format of RCWHR

NOTE

> If index 1 is set, the next time the reset_to_user or reset_to_debug commands are issued, the core should come up with the HRCW overridden. It is not enough to clear index 1 to turn this feature off; a power-on reset should be issued.

4.2.1.2 Blank I^{2}C EEPROM

If you are loading the HRCW from the I^{2}C bus, the override procedure described in the previous section will not work. To avoid this situation, be sure to have an alternate means of programming the EEPROM on the I^{2}C bus or implement an option on the board to use the default built-in HRCW options at power-on reset. The three possible workarounds are:

- Temporarily tie the SDA to the SCL pin rendering the I^{2}C bus non-functional.
- Pull up the reset configuration pins to set the HRCW to load from a different interface such as the local bus, and then override the HRCW word.
- Pull up the reset configuration pins to use one of the default preloaded HRCW so the JTAG tool can gain control. Then program the I^{2}C EEPROM with the appropriate HRCW. This is particularly

useful when the local bus flash memory is programmed for the first time or when reflashing is needed after the flash memory is accidentally erased. Figure 3 shows an example that fetches the HRCW either from the local bus (CFG_RESET_SOURCE[0:2] = 0b000) or chooses a hardcoded HRCW (CFG_RESET_SOURCE[0:2] = 0b100) through a jumper setting.

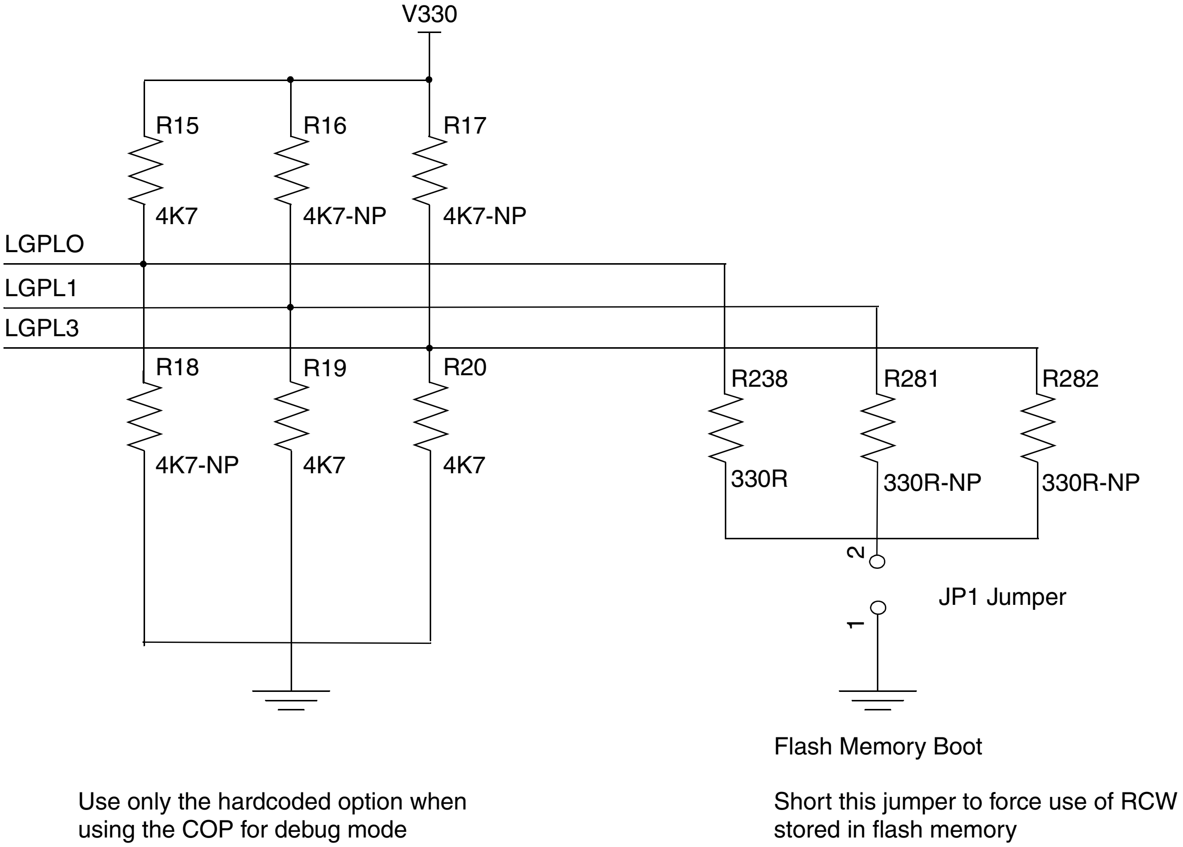

V330
R15
4K7
R16
4K7-NP
R17
4K7-NP
LGPLO
LGPL1
LGPL3
R18
4K7-NP
R19
4K7
R20
4K7
R238
330R
R281
330R-NP
R282
330R-NP
2
1
JP1 Jumper
Flash Memory Boot
Use only the hardcoded option when using the COP for debug mode
Short this jumper to force use of RCW stored in flash memory

Figure 3. Enabling Hardcoded HRCW Option Via Jumper on CFG_RESET_SOURCE[0:2]

4.3 Boot sequencer

The boot sequencer is for loading the hardware reset configuration word and configuring any memory-mapped register before the boot-up code runs. Reset configuration load mode is selected based on the settings of the CFG_RESET_SOURCE[0:2] pins during the power-on reset sequence. The I^{2}C interface loads the reset configuration words from an EEPROM at a specific calling address while the rest of the device is in the reset state. After the reset configuration words are latched inside the device, I^{2}C is reset until $\overline{\text{HRESET}}$ is negated. Then the device can be initialized using boot sequencer mode.

Boot sequencer mode is selected at power-on reset by the RCWHR[BOOTSEQ] field. The I^{2}C module communicates with one or more EEPROMs through the I^{2}C interface to initialize one or more configuration registers of the PowerQUICC II Pro. For example, this code can be used to configure the port interface registers if the device is booting from PCI. Refer to the reference manual for the complete data format for programming the I^{2}C EEPROM. The boot sequencer contains a basic level of error detection. If the I^{2}C boot sequencer fails while loading the reset configuration words, the RSR[BSF] bit is set. If a preamble or CRC fail is detected in boot sequencer mode, there is no internal or external indication that the boot sequencer operation has completed unsuccessfully.

4.4 $\overline{\text{HRESET}}$ and $\overline{\text{SRESET}}$

The $\overline{\text{HRESET}}$ and $\overline{\text{SRESET}}$ signals are not pure input signals. They are open-drain signals that the MPC8323E processor can drive low. The connection on the left side of Figure 4 causes signal contention and must not be used.

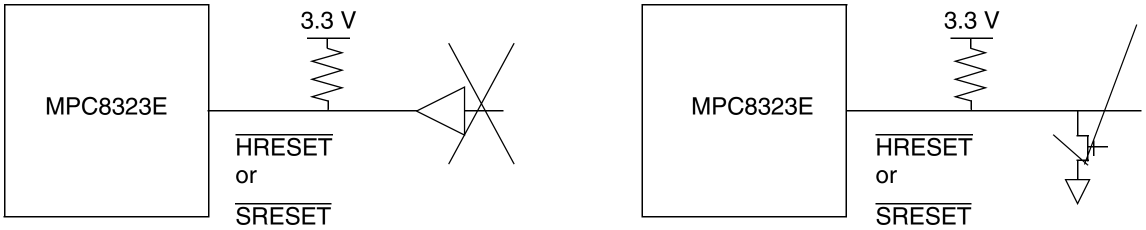

MPC8323E
3.3 V
$\overline{\text{HRESET}}$
or
$\overline{\text{SRESET}}$
MPC8323E
3.3 V
$\overline{\text{HRESET}}$
or
$\overline{\text{SRESET}}$

Figure 4. $\overline{\text{HRESET}}$ and $\overline{\text{SRESET}}$ Connection

4.5 Useful System POR Debug Registers

The hardware reset configuration settings can be read in the RCWLR, the RCWHR, the reset status register (RSR), and the system PLL mode register (SPMR). See the reference manual for details on these registers. Note that all of these resisters are read-only registers.

5 JTAG and Debug

Boundary-scan testing is enabled through the JTAG interface signals. The $\overline{\text{TRST}}$ signal is optional in the IEEE Std 1149.1™ specification, but it is provided on all processors that implement the Power Architecture™. The device requires $\overline{\text{TRST}}$ to be asserted during power-on reset flow to ensure that the JTAG boundary logic does not interfere with normal chip operation. While the TAP controller can be forced to the reset state using only the TCK and TMS signals, generally systems assert $\overline{\text{TRST}}$ during the power-on reset flow. Simply tying $\overline{\text{TRST}}$ to $\overline{\text{PORESET}}$ is not practical because the JTAG interface is also used for accessing the common on-chip processor (COP), which implements the debug interface to the chip.

The COP function of these processors allow a remote computer system (typically, a PC with dedicated hardware and debugging software) to access and control the internal operations of the processor. The COP interface connects primarily through the JTAG port of the processor, with some additional status monitoring signals. The COP port requires the ability to assert $\overline{\text{PORESET}}$ or $\overline{\text{TRST}}$ independently to control the processor fully. If the target system has independent reset sources, such as voltage monitors, watchdog timers, power supply failures, or push-button switches, the COP reset signals must be merged into these signals with logic.

In the arrangement shown in Figure 5, the COP port can assert $\overline{\text{PORESET}}$ or $\overline{\text{TRST}}$ independently while ensuring that the target can drive $\overline{\text{PORESET}}$ as well. The COP interface has a standard header, shown in Figure 5, for connecting to the target system and is based on the 0.025" square-post, 0.100" centered header assembly (often called a Berg header). The connector typically has pin 14 removed as a connector key.

The COP header adds many benefits such as breakpoints, watchpoints, register and memory examination/modification, and other standard debugger features. An inexpensive option can be to leave the COP header unpopulated until needed. There is no standardized way to number the COP header; so emulator vendors have issued many different pin numbering schemes. Some COP headers are numbered top-to-bottom then left-to-right, while others use left-to-right then top-to-bottom. Still others number the pins counter- clockwise from pin 1 (as with an IC). Regardless of the numbering scheme, the signal placement recommended in Figure 5 is common to all known emulators.

If the JTAG interface and COP header are not used, Freescale recommends all of the following connections:

- $\overline{\text{TRST}}$ should be tied to $\overline{\text{PORESET}}$ through a 0 kΩ isolation resistor so that it is asserted when the system reset signal ($\overline{\text{PORESET}}$) is asserted, ensuring that the JTAG scan chain is initialized during the power-on reset flow. Freescale recommends that the COP header be designed into the system as shown in Figure 6. If this is not possible, the isolation resistor allows future access to $\overline{\text{TRST}}$ in case a JTAG interface may need to be wired onto the system in future debug situations.
- Tie TCK to OV$_{\text{DD}}$ through a 10 kΩ resistor to prevent TCK from changing state and reading incorrect data into the device.
- No connection is required for TDI, TMS, or TDO.

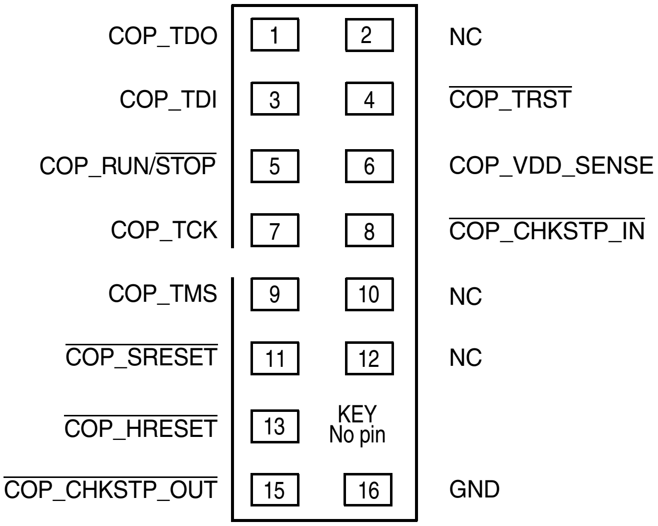
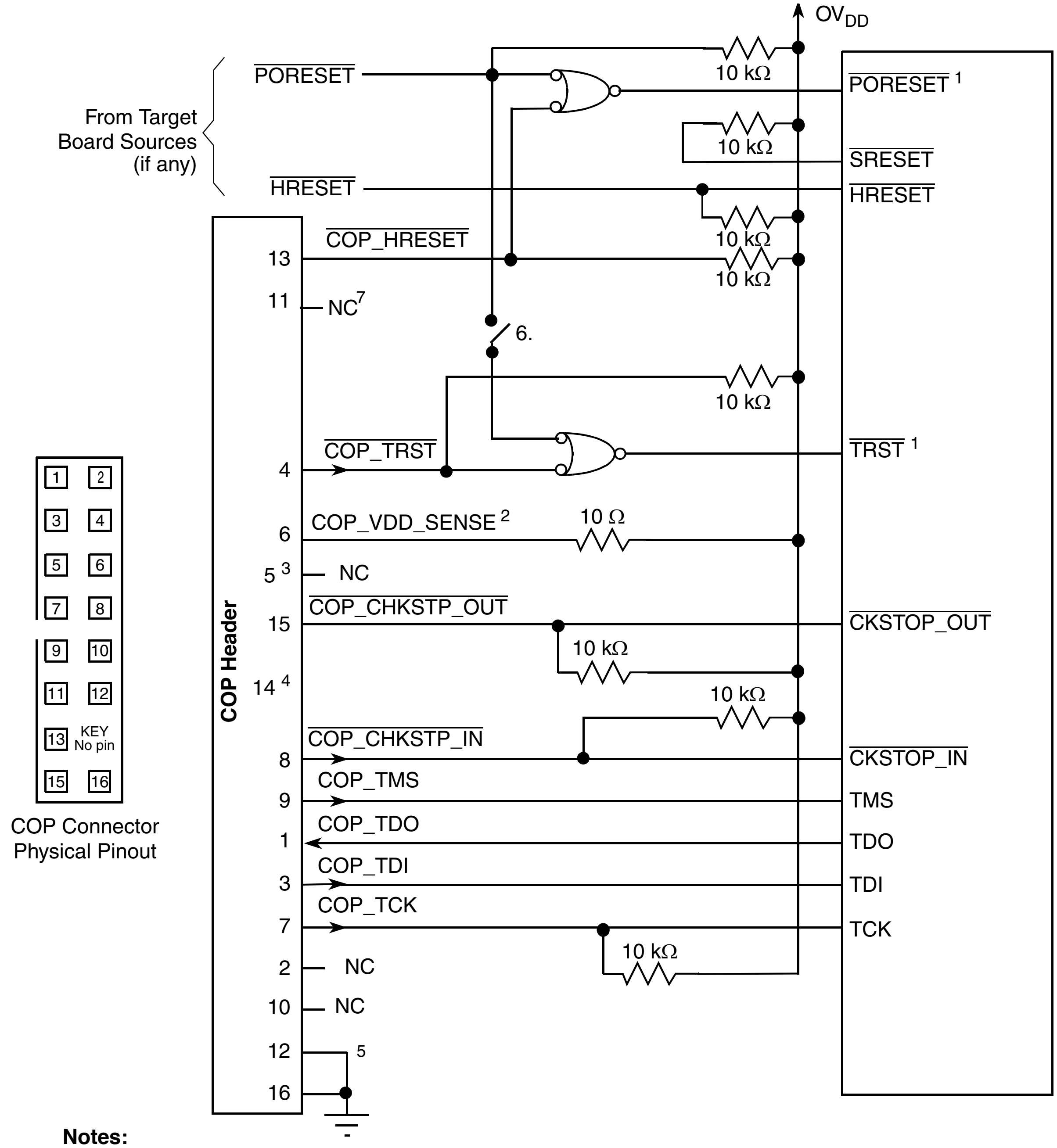

COP_TDO
1
2
NC
COP_TDI
3
4
$\overline{\text{COP\_TRST}}$
COP_RUN/$\overline{\text{STOP}}$
5
6
COP_VDD_SENSE
COP_TCK
7
8
$\overline{\text{COP\_CHKSTP\_IN}}$
COP_TMS
9
10
NC
$\overline{\text{COP\_SRESET}}$
11
12
NC
$\overline{\text{COP\_HRESET}}$
13
KEY
No pin
$\overline{\text{COP\_CHKSTP\_OUT}}$
15
16
GND

Figure 5. COP Connector Physical Pinout



Notes:

1. The COP port and target board should be able to assert PORESET and TRST independently to the processor to control the processor fully as shown here.
2. Populate this with a 10 Ω resistor for short-circuit/current-limiting protection.
3. COP_RUN/STOP, normally found on pin 5 of the COP header, is not implemented on the device. Connect pin 5 of the COP header to OV$_{DD}$ with a 10 kΩ pull-up resistor.
4. The KEY location (pin 14) is not physically present on the COP header.
5. Although pin 12 is defined as a No-Connect, some debug tools may use pin 12 as an additional GND pin for improved signal integrity.
6. This switch is included as a precaution for BSDL testing. The switch should be open during BSDL testing to avoid accidentally asserting the /TRST line. If BSDL testing is not being performed, this switch should be closed or removed.
7. See erratum RESET3 in MPC832xECE.

Figure 6. JTAG Interface Connection

Table 8 details the termination recommendations for the JTAG, TEST, PMC, and thermal management pins.

Table 8. JTAG and TEST Pin Listing

Signal	Pin Type	Connection		Notes
		if used	if not used	
TCK	I	As needed + 10 kΩ to OV_{DD}	10 kΩ to OV_{DD}	Commonly used for boundary scan testing. If this pin is truly not used, it can be tied directly to GND.
TDI	I	As needed	Open	This JTAG pin has a weak internal pull-up P-FET that is always enabled.
TDO	O	As needed	Open	Actively driven during RESET
TMS	I	As needed	Open	This JTAG pin has a weak internal pull-up P-FET that is always enabled.
$\overline{\text{TRST}}$	I	Tie to the output of a Negative OR gate + 10 kΩ to OV_{DD}	Tie to $\overline{\text{PORESET}}$ through a 0 kΩ	This JTAG pin has a weak internal pull-up P-FET that is always enabled. If an In-Circuit Emulator is used in the design, $\overline{\text{TRST}}$ should be tied to the output of a Negative OR gate logic. The inputs to the Negative OR gate logic should be any external $\overline{\text{TRST}}$ source and the $\overline{\text{PORESET}}$ signal
Test				
TEST_MODE	I	Tie directly to GND		—
PMC				
$\overline{\text{QUIESCE}}$	O	As needed	Open	—
Thermal Management				
THERM0	I	As needed	Tie to GND	Thermal sensitive resistor
THERM1	—	As needed	Tie to GND	Thermal sensitive resistor

6 Functional Blocks

This section discusses the recommendations and guidelines for designing with the various functional blocks on the PowerQUICC II Pro processors.

6.1 Local Bus Controller

The local bus runs at up to 66 MHz, The local bus supports four chip selects: $\overline{\text{LCS}}$[0:3]. LGPL4 must be pulled high to OV_{DD} with a 1K-Ω resistor. In the normal case, when MPC8323E boots from the flash memory on the local bus, the local bus uses the GPCM machine. Under GPCM, the LGPL4 functions as $\overline{\text{LGTA}}$, which is an input. Therefore, it must be pulled high. If it is floating and drifts to low, it terminates GPCM access prematurely. The pull-up also avoids the issue described in the local bus signal description table in the reference manual.

The local bus frequency can be adjusted through the LCRR[CLKDIV] register field. After reset, the MPC8323E defaults to a 8:1 ratio. However the MPC8323E only supports Local bus frequencies of 33 and 66 MHz (these are the only frequencies that are currently tested). Therefore, the user must adjust the value of the LCRR[CLKDIV] in order to meet the local bus supported frequencies. Software should not reconfigure the local bus clock while code is executing from the local bus, but rather while code is executing from another interface, such as the DDR controller.

6.1.1 Local Bus Address Signals

To save signals on the local bus, address and data are multiplexed onto the same 16-bit bus. An external latch is needed to demultiplex the 16-bit MSB address and, together with LA[16:25], to reconstruct the original address. No external intelligence is needed, because LALE provides the correct timing to control a standard logic latch. The LAD signals can be directly connected to the data signals of the memory/peripheral. Transactions on the local bus start with an address phase, where the LBC drives the transaction address on the LAD signals and asserts the LALE signal. This can be used to latch the address and then the LBC can continue with the data phase. Figure 7 and Figure 9 illustrate the connection of the local bus in both multiplexed and non-multiplexed address modes defined by the state of the CFG_LBIU_MUX_EN signal. For details on the operation of the local bus controller, refer to the reference manual.

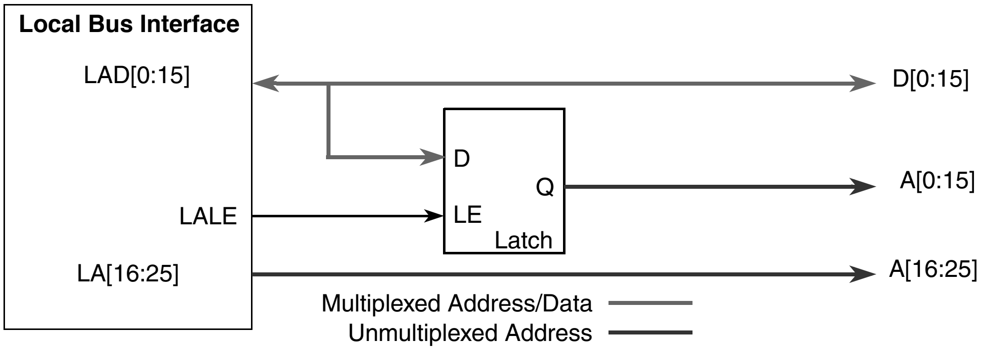


Figure 7. Local Bus Address Connection Example (Multiplexed Mode)

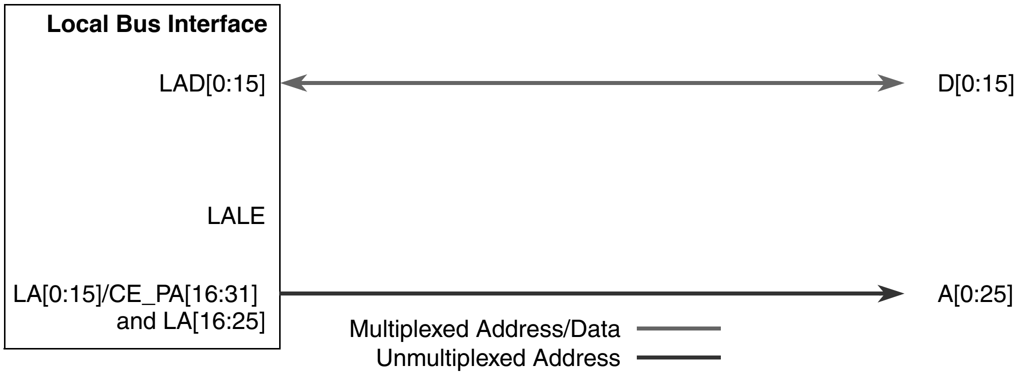


Figure 8. Unmultiplexed Address and Data Bus Connection Example

6.1.2 Connecting Devices to the Local Bus

The MPC8323E local bus features a multiplexed address and data bus, LAD[0:15]. An external latch is required to de-multiplex these signals to the connecting device. Figure 9 shows the timing of LALE. When LALE is high, it indicates LAD[0:15] is in the address phase.

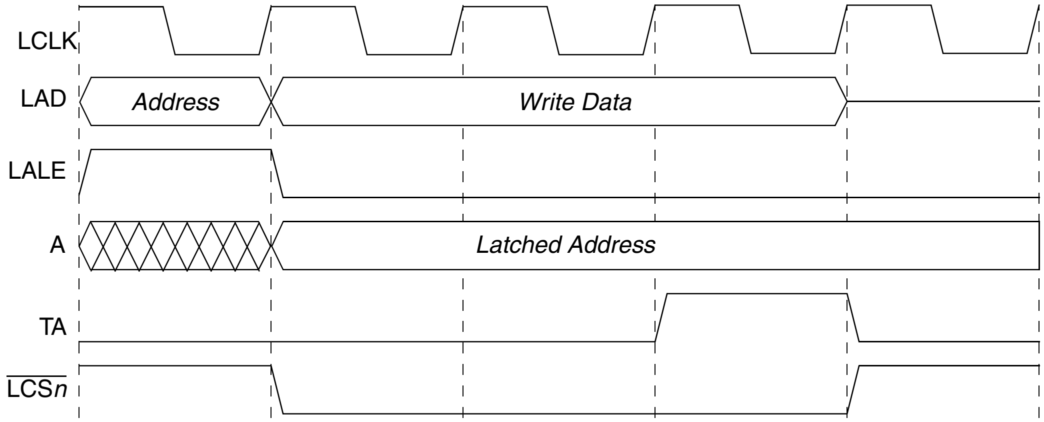


Figure 9. LALE Timing

For every assertion of $\overline{\text{LCS}}$n, LALE is asserted first. While LALE is asserted, all other control signals are negated. The duration of LALE can be programmed to 1–4 cycles in LCRR[EADC]. The default is 4 cycles. The timing of LALE negation is important to ensure the correct latch. If the change of LAD and negation of LALE are too close and the margin for the latch is not sufficient, RCWHR[LALE] can be set. LALE is negated ½ a local bus clock earlier, which should ensure enough margin. Table 9 shows the termination recommendations for the local bus pins.

Table 9. Local Bus Pin Listing

Critical	Signal	Pin Type	Connection		Notes
			If Used	If Not Used	
Local Bus Interface					
—	LAD[0:15]	I/O	As needed	2 k–10 kΩ to OV_{DD}	—
—	LA[0:25]	O	As needed	Open	—
—	$\overline{\text{LCS}}$[0:3]	O	As needed	Open	—
—	$\overline{\text{LWE}}$[0:1]/$\overline{\text{LBS}}$[0:1]	O	As needed	Open	—
—	LBCTL	O	As needed	Open	—
—	LALE	O	As needed	Open	—
X	LGPL0/ **CFG_RESET_SOURCE0**	I/O	As needed	See Table 6	—
X	LGPL1/ **CFG_RESET_SOURCE1**	I/O	As needed	See Table 6	—
—	LGPL2/$\overline{\text{LOE}}$	O	As needed	Open	—
X	LGPL3/ **CFG_RESET_SOURCE2**	I/O	As needed	See Table 6	—

Table 9. Local Bus Pin Listing (continued)

Critical	Signal	Pin Type	Connection		Notes
			If Used	If Not Used	
—	LGPL4/$\overline{\text{LGTA}}$/LUPWAIT	I/O	As needed + 1 k–10 kΩ to OV_{DD}	Open	—
X	LGPL5	O	As needed	Open	—
—	LCLK[0:1]	O	As needed	Open	—

6.2 DDR SDRAM

Refer to the following application notes for details on layout considerations and DDR programming guidelines:

- For signal integrity and layout considerations: *Hardware and Layout Design Considerations for DDR Memory Interfaces* (AN2582).
- *Programming the PowerQUICC III DDR SDRAM Controller* (AN2583).
- For signal integrity and layout considerations: *Hardware and Layout Design Considerations for DDR2 SDRAM Memory Interfaces* (AN2910).

The MPC8323E DDR controller supports both DDR1 and DDR2 SDRAM selectable via the DDRCDR register (please see the Reference Manual for details). The MPC8323E DDR controller can be configured with a 32-bit data bus interface only. Software must set the DDR_SDRAM_CFG[32_BE] field. Also, the burst length should be set to 8 beats in 32-bit mode by properly configuring the DDR_SDRAM_CFG[8_BE]. Refer to the MPC8323E reference manual. Table 10 lists the termination recommendations for the different DDR signals for both controllers.

Table 10. DDR SDRAM Pin Listing

Critical	Signal	Pin Type	Connection		Notes
			If Used	If Not Used	
—	MDQ[0:31]	I/O	As needed	Open	Proper signal integrity analysis must be performed using the respective device IBIS model. • Parallel termination is optional for DDR signals and should be simulated to verify necessity. • Differential termination is included on DIMMs. It is only required for discrete memory applications.
—	MDM[0:3]	O	As needed	Open	—
—	MDQS[0:3]	I/O	As needed	Open	—
—	MBA[0:2]	O	As needed	Open	—
—	MA[0:13]	O	As needed	Open	—
—	$\overline{\text{MWE}}$	O	As needed	Open	—

Table 10. DDR SDRAM Pin Listing (continued)

Critical	Signal	Pin Type	Connection		Notes
			If Used	If Not Used	
—	$\overline{\text{MRAS}}$	O	As needed	Open	—
—	$\overline{\text{MCAS}}$	O	As needed	Open	—
—	$\overline{\text{MCS}}$	O	As needed	Open	—
—	MCKE	O	As needed	Open	This output is actively driven during reset rather than being three-stated during reset.
—	MCK	O	As needed	Open	—
—	$\overline{\text{MCK}}$	O	As needed	Open	—
—	MODT	O	As needed	Open	—

6.3 PCI Bus Interface

If PCI is used, program $\overline{\text{INTA/IRQ\_OUT}}$ to open drain and pull it up with a 10-kΩ resistor.

The RCWHR controls the hardware configuration of the PCI blocks as follows:

- RCWH[PCIHOST]. Host/Agent mode for PCI.
- RCWH[PCIARB]. PCI internal/external arbiter mode select.

When the device is configured as a PCI host, the device provides three clock output signals for external PCI agents, PCI_CLK_OUT[0:2]. Each clock output can be enabled (enable toggling of the clock) by setting its corresponding OCCR[PCICOE*n*] bit. It is recommended to turn off the unused clock outputs by writing to OCCR[PCICOEn] in order to reduce EMI.

The MPC8323E PCI interface is hot swap friendly, meaning that it supports the hardware and software connection processes as defined in the Hot Swap specification. This level of support allows the board and system designers to build full hot swap systems based on the MPC8323E as a PCI target device. It is assumed that the system uses the external arbiter from the PCI host. Therefore, the extra pins needed for the CompactPCI Hot Swap are multiplexed with the PCI arbitration signals. When an external arbiter is selected (RCWH[PCIARB] = 0), the $\overline{\text{CompactPCI}}$ Hot Swap pins function. When an internal arbiter is selected (RCWH[PCIARB] = 1), the $\overline{\text{GNT}}$x/$\overline{\text{REQ}}$x pins function.

Refer to the MPC8323E reference manual for details on the usage of the RCWH settings.

Table 11 lists the PCI bus interface pins.

Table 11. PCI Bus Interface Pin Listing

Critical	Signal	Pin Type	Connection		Notes
			If Used	If Not Used	
PCI Interface					
—	$\overline{\mathrm{PCI\_INTA}}$/$\overline{\mathrm{IRQ\_OUT}}$	O	2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	Open-drain signal. In agent mode, $\overline{\mathrm{INTA}}$ typically connects to a central interrupt controller. In host mode, $\overline{\mathrm{INTA}}$ can be used to assert interrupts to other devices, such as a second processor.
—	$\overline{\mathrm{PCI\_RESET\_OUT}}$	O	As needed	Open	This signal is used only in host mode. It should be left unconnected in agent mode.
—	PCI_AD[31:0]	I/O	As needed	2 k–10 kΩ to OV_{DD} or Open	If the PCI port is not used and the bus is parked, no termination is needed. Software must park the bus as follows: 1. RCWHR[PCIHOST] = 1 2. RCWHR[PCIARB] = 1 3a. PCI Arbiter Control Configuration Register PM bit = 1, or 3b. PCI_GCR[BBR] = 1
—	PCI_C/$\overline{\mathrm{BE}}$[3:0]	I/O	As needed	2 k–10 kΩ to OV_{DD} or Open	If the PCI port is not used and the bus is parked, no termination is needed. Software needs to park the bus as follows: 1. RCWHR[PCIHOST] = 1 2. RCWHR[PCI1ARB] = 1 3a. PCI Arbiter Control Configuration Register PM bit = 1, or 3b. PCI_GCR[BBR] = 1
—	PCI_PAR	I/O	As needed	2 k–10 kΩ to OV_{DD}	If the PCI port is not used, this signal must be pulled up.
—	$\overline{\mathrm{PCI\_FRAME}}$	I/O	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	PCI specification requires a weak pullup.
—	$\overline{\mathrm{PCI\_TRDY}}$	I/O	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	PCI specification requires a weak pullup.
—	$\overline{\mathrm{PCI\_IRDY}}$	I/O	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	PCI specification requires a weak pullup.
—	$\overline{\mathrm{PCI\_STOP}}$	I/O	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	PCI specification requires a weak pullup.
—	$\overline{\mathrm{PCI\_DEVSEL}}$	I/O	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	PCI specification requires a weak pullup.

Table 11. PCI Bus Interface Pin Listing (continued)

Critical	Signal	Pin Type	Connection		Notes
			If Used	If Not Used	
—	PCI_IDSEL	I	**PCI host:** Tie to GND **PCI agent:** One of PCI_AD[31:0]	Tie to GND	IDSEL should be connected to GND for host systems and to one address line for agent systems. If the PCI port is not used, it should be grounded. • PCI host is selected by RCWH[PCIHOST] = 1. • PCI agent is selected by RCWH[PCIHOST] = 0.
—	$\overline{\mathrm{PCI\_SERR}}$	I/O	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	PCI specification requires a weak pullup.
—	$\overline{\mathrm{PCI\_PERR}}$	I/O	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	PCI specification requires a weak pullup.
—	$\overline{\mathrm{PCI\_REQ0}}$	I/O	**External arbiter:** As needed **Internal arbiter:** As needed + 2 k–10 kΩ to OV_{DD}	**External arbiter:** Open **Internal arbiter:** 2 k–10 kΩ to OV_{DD}	If an external arbiter is used, $\overline{\mathrm{REQ0}}$ becomes an *output* signal and does not need to be terminated. • External arbiter selected by RCWH[PCIARB] = 0. • Internal arbiter selected by RCWH[PCIARB] = 1.
—	$\overline{\mathrm{PCI\_REQ1}}$/ CPCI_HS_ES	I	**External arbiter:** As needed **Internal arbiter:** As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	This pin is multiplexed with a CompactPCI Hot Swap function. CompactPCI functionality selected when external arbiter is used. • External arbiter selected by RCWH[PCIARB] = 0. • Internal arbiter selected by RCWH[PCIARB] = 1.
—	$\overline{\mathrm{PCI\_REQ2}}$	I	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	—
—	$\overline{\mathrm{PCI\_GNT0}}$	I/O	**External arbiter:** As needed + 2 k–10 kΩ to OV_{DD} **Internal arbiter:** As needed	**External arbiter:** 2 k–10 kΩ to OV_{DD} **Internal arbiter:** Open	If an external arbiter is used, $\overline{\mathrm{GNT0}}$ becomes an *input* signal and should be pulled up with 2 k–10 kΩ to OV_{DD}. • External arbiter selected by RCWH[PCIARB] = 0. • Internal arbiter selected by RCWH[PCIARB] = 1.

Table 11. PCI Bus Interface Pin Listing (continued)

Critical	Signal	Pin Type	Connection		Notes
			If Used	If Not Used	
—	PCI_GNT1/ CPCI_HS_LED	O	As needed	Open	This pin is multiplexed with a CompactPCI Hot Swap function. CompactPCI functionality selected when external arbiter is used. • External arbiter selected by RCWH[PCIARB] = 0. • Internal arbiter selected by RCWH[PCIARB] = 1.
—	PCI_GNT2/ CPCI_HS_ENUM	O	**External arbiter:** As needed + 2 k–10 kΩ to OV_{DD} **Internal arbiter:** As needed	**External arbiter:** Open **Internal arbiter:** Open	This pin is multiplexed with a CompactPCI Hot Swap function. CompactPCI functionality selected when external arbiter is used. • If **CompactPCI** Hot Swap function is used, a weak pullup is required (2 k–10 kΩ to OV_{DD}). • External arbiter selected by RCWH[PCIARB] = 0. • Internal arbiter selected by RCWH[PCIARB] = 1.
—	M66EN	I	As needed	5 kΩ to OV_{DD} or 1 kΩ to GND	No role if PCI is not used.

6.4 Integrated Programmable Interrupt Controller (IPIC)

The integrated programmable interrupt controller (IPIC) provides interrupt management for receiving hardware-generated interrupts from different sources (both internal and external). It also prioritizes and delivers the interrupts to the CPU for servicing. MCP_OUT is an open-drain signals that must be pulled up with a 4.7-kΩ resistor. IRQ0/MCP_IN is an input when the e300 core is enabled. When the e300 core is disabled, IRQ0/MCP_IN is an output that must be connected to the host processor. IRQ[6:7] are multiplexed with other functions and are controlled by the SICRL register. Unused IRQ signals must be pulled up to OV_{DD}. The IIC_SCL/CKSTOP_IN and IRQ[7]/CKSTOP_IN] signals must be actively driven or pulled up to OV_{DD} if they are programmed as CKSTOP_IN.

Table 12 lists the PIC pins.

Table 12. Programmable Interrupt Controller Pin Listing

Critical	Signal	Pin Type	Connection		Notes
			If Used	If Not Used	
—	MCP_OUT	O	As needed + 4.7 kΩ to OV_{DD}	4.7 kΩ to OV_{DD}	Open-drain signal

Table 12. Programmable Interrupt Controller Pin Listing (continued)

Critical	Signal	Pin Type	Connection		Notes
			If Used	**If Not Used**	
—	IRQ[0]/MCP_IN	I	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	—
—	IRQ[1:5]	I	As needed + 2 k–10 kΩ to OV_{DD}	2 k–10 kΩ to OV_{DD}	—
—	IRQ[6]/CKSTOP_OUT	I/O	**CKSTOP_OUT:** As needed + 10 kΩ to OV_{DD} **Others:** As needed + 2k-10 kΩ to OV_{DD}	**CKSTOP_OUT:** Open **Others:** 2 k–10 kΩ to OV_{DD}	Pin functionality determined by SICRL[IRQ_CKS] bit settings.
—	IRQ[7]/CKSTOP_IN	I/O	**CKSTOP_IN:** As needed + 10 kΩ to OV_{DD} **Others:** As needed + 2 k-10 kΩ to OV_{DD}	**CKSTOP_IN:** 4.7 kΩ to OV_{DD} **Others:** 2 k–10 kΩ to OV_{DD}	Pin functionality determined by SICRL[IRQ_CKS] bit settings.

6.5 DUART

The MPC8323E DUART module provides two standard UART interfaces, and any UCC from the QUICC Engine block can be programmed to function as UART. Refer to Section 6.7.3, “QUICC Engine UART.” The DUART pins are multiplexed with the DDR controller and local bus controller debug functions. The functions are programmed with SICRL[URT_CTPR]. If only UARTn_SOUT and UARTn_SIN are used, UARTn_CTS must be pulled down.

Table 13 lists the DUART pins.

Table 13. Dual UART Pin Listing

Critical	Signal	Pin Type	Connection		Notes
			If Used	**If Not Used**	
—	UART1_SOUT/ M1SRCID[0]/ LSRCID[0]	O	As needed	Open	Pin functionality determined by SICRL[URT_CTPR] bit setting.

Table 13. Dual UART Pin Listing (continued)

Critical	Signal	Pin Type	Connection		Notes
			If Used	If Not Used	
—	UART1_SIN/ M1SRCID[1]/ LSRCID[1]	I/O	As needed	2 k–10 kΩ to GND or program pin to function as a debug pin	Pin functionality determined by SICRL[URT_CTPR] bit setting.
—	UART1_CTS/ M1SRCID[2]/ LSRCID[2]	I/O	As needed	2 k–10 kΩ to GND or program pin to function as a debug pin	Pin functionality determined by SICRL[URT_CTPR] bit setting If DUART1 is used, but CTS is not used, pull it down.
—	UART1_RTS/ M1SRCID[3]/ LSRCID[3]	O	As needed	Open	Pin functionality determined by SICRL[URT_CTPR] bit setting. If DUART1 is used but RTS is not used, leave open.
—	UART2_SOUT/ M1SRCID[4]/ LSRCID[4]	O	As needed	Open	Pin functionality determined by SICRL[URT_CTPR] bit setting.
—	UART2_SIN/MDVAL/ LDVAL	I/O	As needed	2 k–10 kΩ to GND or program pin to function as a debug pin	Pin functionality determined by SICRL[URT_CTPR] bit setting.
—	UART2_CTS	I	As needed	2 k–10 kΩ to GND or program pin to function as a debug pin	If DUART2 is used, but CTS is not used, pull it down.
—	UART2_RTS	O	As needed	Open	If DUART2 is used but RTS is not used, leave open.

6.6 I^{2}C Interface

Table 14 lists the I^{2}C pins.

Table 14. I^{2}C Pin Listing

Signal	Pin Type	Termination		Notes
		If Used	If Not Used	
IIC1_SCL/CKSTOP_IN	I/O	**$\overline{\text{CKSTOP\_IN}}$:** As needed + 10 kΩ to OV_{DD} IIC1_SCL: As needed + 2 k-10 kΩ to OV_{DD}	**$\overline{\text{CKSTOP\_IN}}$:** 4.7 kΩ to OV_{DD} IIC1_SCL: 2 k–10 kΩ to OV_{DD}	Pin functionality determined by SICRL[LDP_LCS_A] bit settings.
IIC1_SDA/CKSTOP_OUT	I/O	**$\overline{\text{CKSTOP\_OUT}}$:** As needed + 10 kΩ to OV_{DD} **Others:** As needed + 2k-10 kΩ to OV_{DD}	**$\overline{\text{CKSTOP\_OUT}}$:** Open **Others:** 2 k–10 kΩ to OV_{DD}	Pin functionality determined by SICRL[LDP_LCS_A] bit settings.

6.7 QUICC Engine Communication Interfaces

The QUICC Engine communication interfaces include the Ethernet controller, UTOPIA/POS, universal asynchronous receiver/transmitter (UART), universal serial bus (USB) controller, and serial peripheral interface (SPI).

6.7.1 Ethernet Controller

The interfaces of the QUICC Engine are as follows:

- Media-independent interface (MII)
- Reduced media-independent interface (RMII)

6.7.1.1 Media-Independent Interface (MII)

The MPC8323E supports three Ethernet interfaces. UCC2, UCC3, and UCC4 can be programmed to be an Ethernet controller. We denote UCC*n* Ethernet as Ethernet *n*. All UCC Ethernet interfaces have dedicated NMSI pins that support MII interfaces, excluding the clocking signals. The NMSI signals include:

- Enet-TXD[0:3]
- Enet-TX_EN
- Enet-TX_ER
- Enet-RXD[0:3]

- Enet-COL
- Enet-CRS
- Enet-RX_DV
- Enet-RX_ER

The MPC8323E MII interface clocking pins (RX_CLK and TX_CLK) are inputs. They are driven by the PHY device. The RX_CLK and TX_CLK are routed through CLKx pins that are multiplexed with other Parallel I/O Ports pin signals. See the MPC8323E reference manual, Section 3.4. For each Ethernet *n*, the RX_CLK and TX_CLK multiplexing is controlled by CMXUCRx.

Because the Ethernet pins are multiplexed with other I/O port pins, all three Ethernet interfaces can render other interfaces unusable due to pin multiplexing limitations. Therefore, you should carefully review the port tables in Section 3.4.7 of the MPC8323E reference manual before starting a new design.

6.7.1.2 RMII Interface Connection

All UCC Ethernet interfaces (UCC2, UCC3, and UCC4) support the RMII interface, which uses only a subset of the MII signals. The MII signals should be connected as shown in Table 15. For RMII, the TX and RX share one clock called REF_CLK. The REF_CLK from the PHY must be connected to TX_CLK of the MPC8323E. RX_CLK is not used.

Table 15. RMII Connection

MPC8323E Signals	PHY Signals
Enet-TXD[0:1]	TXD[0:1]
Enet-TX_EN	TX_EN
Enet-RXD[0:1]	RXD[0:1]
Enet-RX_ER	RX_ER
Enet-RX_DV	CRS_DV
TX_CLK	REF_CLK

6.7.1.3 Ethernet Management Interface

The Ethernet management interface can be controlled by a UCC or the dedicated SPI2 interface. Section 6.7.5, “Serial Peripheral Interface (SPI),” describes how to configure SPI2. Each UCC has its own built-in Ethernet management logic. CMXGCR[MEM] determines which UCC masters the serial management interface (SMI). SPI2 or UCC is selected in the QUICC Engine port (see Table 16). To use SPI2, program PD4 and PD5 to SPI2:MDIO and SPI2:MDC, respectively. To use the UCC management interface, program PD4 and PD5 to CE MUX:MDIO and CE MUX:MDC, respectively.

6.7.2 Utopia

The MPC8323E supports one Utopia L2 interface. If you are familiar with the Utopia interface of the CPM in MPC82xx and MPC85xx, note that the external signal naming convention of the QUICC Engine block follows the Utopia standard. Therefore, there is different naming in master and slave modes. The naming conventions in the CPM retain the master mode signal naming for slave mode. For example, the QUICC Engine block transmit TXSOC in slave mode is named RXSOC, but the CPM transmit SOC in slave mode is named TXSOC. In the QUICC Engine block, you should connect signals between master and slave by name. In the example here, we connect the external master TXSOC with the QUICC Engine TXSOC.

6.7.3 QUICC Engine UART

Each UCC in the MPC8323E QUICC Engine block can be programmed to function as a UART controller. The QUICC Engine UART programming model is compatible with that of the CPM SCC UART. Therefore, the user may prefer to use the QUICC Engine UART to reuse existing CPM SCC UART software drivers. However, using the QUICC Engine UART consumes I/O port pins and also acts as a load for the RISC controller inside the QUICC Engine block. If you face either pin multiplexing limitation issues or QUICC Engine performance issues, you may instead decide to use the DUART interface discussed in Section 6.5, “DUART.”

6.7.3.1 UART Configuration

The pins of the QUICC Engine UART are on each UCC NMSI interface, and they are programmed through the following registers:

- CPODRx: Determines the open-drain configuration, one bit per pin.
- CPDIR1x, CPDIR2x: Determines the in/out characteristics of the pins, two bits per pin.
- CPPAR1x, CPPAR2x: Determines the functionality of each pin, two bits per pin

Refer to the parallel I/O port table in Section 3.4 of the MPC8323E reference manual for details on the pin multiplexing of each UART pin. In general, the mapping for the QUICC Engine UART pins is as described in Table 16.

NOTE

The MPC8323E rev 1.1 must load a RAM microcode package to use the QUICC Engine UART protocol.

Table 16. QUICC Engine UART Pin Listing

UCC No.	Signal	QE Port	Termination
UCCn (where n = 1,2,3,4, or 5)	UARTn_SOUT	SERn_TXD[0]	If QE UARTn is not used, all the pins can be programmed for other functions. If $\overline{\text{CTS}}$ is programmed for UART function but is not connected, it must be pulled low. Programming $\overline{\text{CTS}}$ for non-UART use automatically terminates the pin to low. No pulldown resistor is needed.
	UARTn_SIN	SERn_RXD[0]	
	UARTn_CTS	$\overline{\text{SERn\_CTS}}$	
	UARTn_RTS	$\overline{\text{SERn\_RTS}}$	

6.7.4 USB controller

The USB controller interfaces to the USB bus through a differential line driver and differential line receiver. The $\overline{\text{OE}}$ (output enable) signal enables the line driver when the USB controller transmits on the bus. See Figure 10.

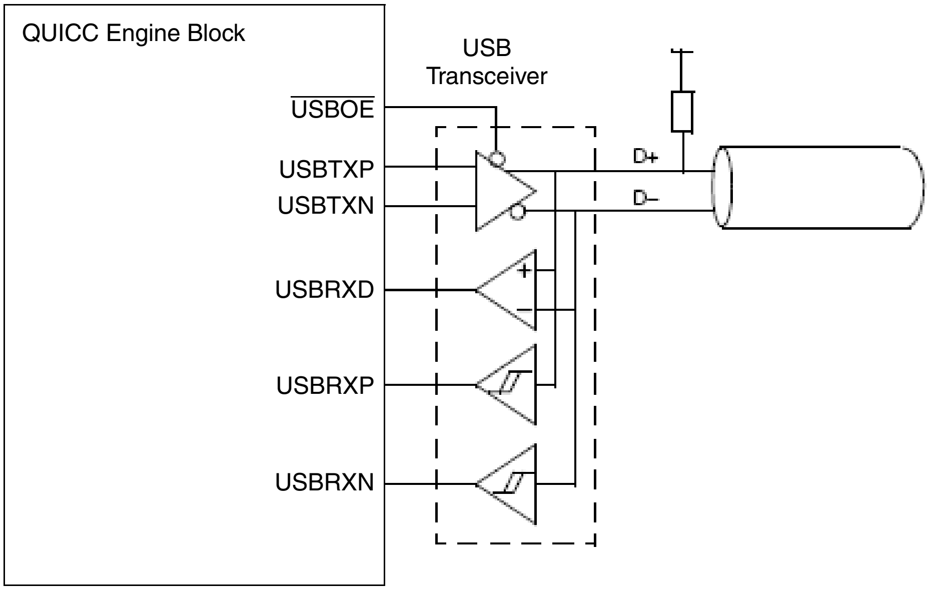


Figure 10. USB Interface

In addition, a reference clock must be provided. CMXGCR[USBCS] determines the source of USB clock. The possible clock sources are CLK3, CLK5, CLK7, CLK9, CLK13, CLK17, CLK19, BRG9, and BRG10. The USB reference clock must be four times of the USB bit rate. Thus, it must be 48 Mhz for a 12-Mbps full-speed transfer or 6 Mhz for a 1.5-Mbps low-speed transfer.

Table 17. USB pins and connections

Signal	QE Port	Termination
USB_OE	PA8	If USB is not used, program these signals for general-purpose IO or other QE functions.
USB_TP	PA1	
USB_TN	PA0	
USB_RP	PA4	
USB_RN	PA5	
USB_RXD	PA6	
USBCLK	CLK3, CLK5, CLK7, CLK9, CLK13,CLK17, CLK19, BRG9, BRG10	

6.7.5 Serial Peripheral Interface (SPI)

The MPC8323E supports two serial peripheral interfaces (SPI). SPI2 is dedicated to Ethernet PHY management.

Table 18. SPI pin listing

SPI No.	Signal	Pin Type	QE Port	Termination
SPI1	SPIMOSI	I/O	PD0	• Configure for another function if not used for SPI2. • For systems supporting SPI master and slave modes, configure to open drain and pull up to OV_{DD}.
	SPIMISO	I/O	PD1	• Configure for another function if not used for SPI2 • For systems supporting SPI master and slave modes, configure to open drain and pull up to OV_{DD}.
	SPICLK	I/O	PD2	• Configure for another function if not used for SPI2 • For systems supporting SPI master and slave modes, configure to open drain and pull up to OV_{DD}.
	SPISEL	I	PD3	Master mode: Pullup to OV_{DD} Slave mode: Pulldown to GND
SPI2	MDIO	I/O	PD4	Configure for other function if not used for SPI2
	MDC	O	PD5	Configure for other function if not used for SPI2

7 Revision history

This table provides a revision history for this application note.

Table 19. Document revision history

Rev, Number	Date	Substantive Change(s)
5	01/2015	• In Table 1, included information for Rev 1.3. • Updated Figure 2 to include CLKIN/$\overline{\text{CLKIN}}$ signals. • In Figure 6, removed $\overline{\text{SRESET}}$ connections. $\overline{\text{SRESET}}$ doesn't work in 8323, it's an erratum. • In Table 17, “USB pins and connections,” removed reference to CLK21.
4	04/2008	• In Section 1.1, “References,” removed third bullet under “Available Tools.” • Removed Section 1.5.
3	01/2008	The following updates are made to the current revision: • For the $\overline{\text{CFG\_CLKIN\_DIV}}$ signal in Table 5, the pin type is changed from I/O to I. • For the LGPL5 signal in Table 9, the changes are made as follows: – The pin type is changed from I/O to O – Connection: If used, changed to As needed; if not used, changed to Open • Row corresponding to the $\overline{\text{CFG\_CLKIN\_DIV}}$ signal is removed
2	07/2007	In Section 3.3, “System Clock if PCI is Disabled,” changed the second sentence to: CLKIN should be tied to GND and $\overline{\text{CFG\_CLKIN\_DIV}}$ should be tied high.
1	06/2007	Page 20: PCI_C/$\overline{\text{BE}}$[3:0] do not need pull-up. Changed fourth column to “As needed” only. Page 18: Updated Table 9 to state that the LGPL5 pin and the $\overline{\text{CFG\_CLKIN\_DIV\_B}}$ pin are two different pins. They had been shown as one pin.
0	12/2006	Initial release.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.