# PI7C9X130 PCI Express to PCI Reversible Bridge FAQ

Revision 0.2

## Table of Contents

## 1 What are the current datasheet and silicon versions?

The current revision of the datasheet is rev.1.6
The current revision of the schematic for Forward bridge is “9X130 V402 050908”.
The current revision of the schematic for Reverse bridge is “PI7C9X130 REVERSE V2.2”.
The current revision of the silicon is: PI7C9X130**D**

## 2 What is a Reversible Bridge?

PI7C9X130 can be configured via “REVRSB” strap pin to work in Forward or Reverse mode.
In Forward mode, the device is capable of PCIe-to-PCI bridging and fan-out, I.e., CPU ←PCIe→ PI7C9X130 ←PCI→ PCI end devices. The device is connected to CPU via PCIe interface and to up to 8 PCI devices via PCI interface. In this mode, CPU handles the bridge as a PCIe end point, and for the PCI end devices, the bridge acts as a PCI host.

In Reverse mode, the device is capable of PCI-to-PCIe bridging, I.e., CPU ←PCI→ PI7C9X130 ←PCIe→ PCIe end device. The bridge in Reverse mode is useful in the application where the CPU has only PCI interface, but needs to connect to PCIe end device. In this mode, CPU handles the bridge as a PCI device, and for the PCIe end device, the bridge acts as the root complex.

## 3 What are Transparent and Non-transparent modes?

In Transparent mode, the bridge appears transparent to CPU and PCI end devices. In this mode, CPU communicates to the PCI end devices as if they are directly connected.

Non-transparent mode is usually for multi-CPU scenarios, where individual CPU needs to be segmented in order to associate specific CPU to PCI end devices. In this mode, CPU can not directly communicate the PCI end devices, and the bridge must translate the addresses for them.

## 4 What is the bandwidth of PCIe port of PI7C9X130?

PI7C9X130’s PCIe port is a x4 PCIe interface and capable of 2.5Gb/s clock rate.

## 5 What is the bandwidth of PCI ports of PI7C9X130?

PI7C9X130’s PCI ports are a 64-bit PCI-X interface and capable of 133MHz clock rate.

## 6 What should I do with the PCI interface if I intend to use it in 32-bit mode?

Per PCI Specification, 64-bit ports are compatible with 32-bit mode. AD[63:32], C/BE[7:4], PAR64, REQ64#, and ACK64# signals should be pulled up through a 5.1k-ohm resistor.

DEV64 signal should be pulled down to indicate 32-bit mode.

## 7 Does PI7C9X130 support 5V PCI devices on the PCI interface?

Yes, PI7C9X130 supports 5V PCI devices on the PCI interface.

## 8 Does PI7C9X130 support Industrial Temperature range?

Yes, PI7C9X130 supports Industrial Temperature range (-40$^{o}$C to 85$^{o}$C).

## 9 Does PI7C9X130 support Hot-Plug function?

Yes, PI7C9X130 supports Hot-Plug function, but only the system supports with compatible hardware, software, and connectors. Please refer to the following specification for details:

- PCI Hot-Plug Specification
- PCI Standard Hot-Plug Controller and Subsystem Specification

## 10 Does PI7C9X130 support Burst Transmission mode?

Yes, PI7C9X130 supports Burst Transmission mode.

## 11 Does PI7C9X130 support 64-bit address space?

Yes, PI7C9X130 supports 64-bit address space.

## 12 What is the requirement of power-up sequence for PI7C9X130?

PI7C9X130 requires 3.3V power supply before 1.8V.

## 13 Does PI7C9X130 work without EEPROM?

Yes, PI7C9X130 works without EEPROM. The device uses the default values in the registers. If EEPROM is available during power-up, the device loads the values from EEPROM after validating the content, and over-writes the default values in the registers. After initial power-up, the register values can be written via PCI/PCIe configuration registers.

## 14 If EEPROM is not used in the system, how should be done to the SMBCLK/SCL and SMBDATA/SDA pins?

Per PCI Specification 4.3.3, these two pins should be pulled up. The recommended value of pull-up register is 5.1k-ohm.

## 15 Which EEPORM models are compatible with PI7C9X130?

PI7C9X130 is compatible with AT24C02B, AT24C04B, AT24C08B, AT24C16B, and other compatible EEPROM models. However, AT24C64 is not recommended due to the addressing method of this model.

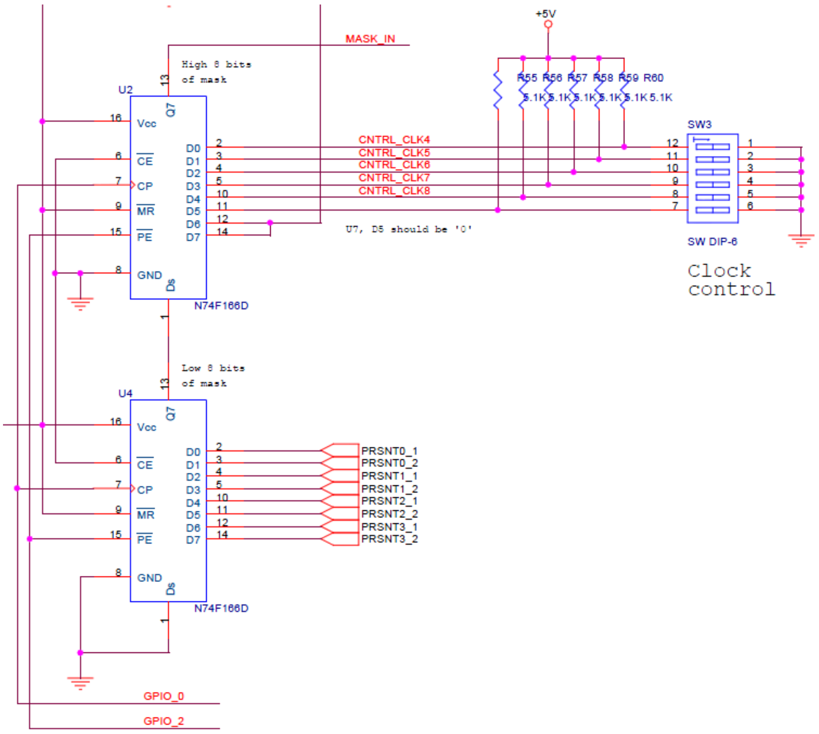
## 16 What is the length of trace that high speed PCIe signals support, and how is the length calculated if backplane is used?

Many factors affect length of the trace that can be supported including PCB material, connector, etc. Please refer to PCIe Specification. If the trace length is longer than expected, or if backplane is used, we recommend that PCIe re-drivers be used to improve the signals.

Please contact Pericom FAE for Re-Driver product information.

## 17 Voltage on “MSK_IN” pin must kept stable. How do we achieve this?

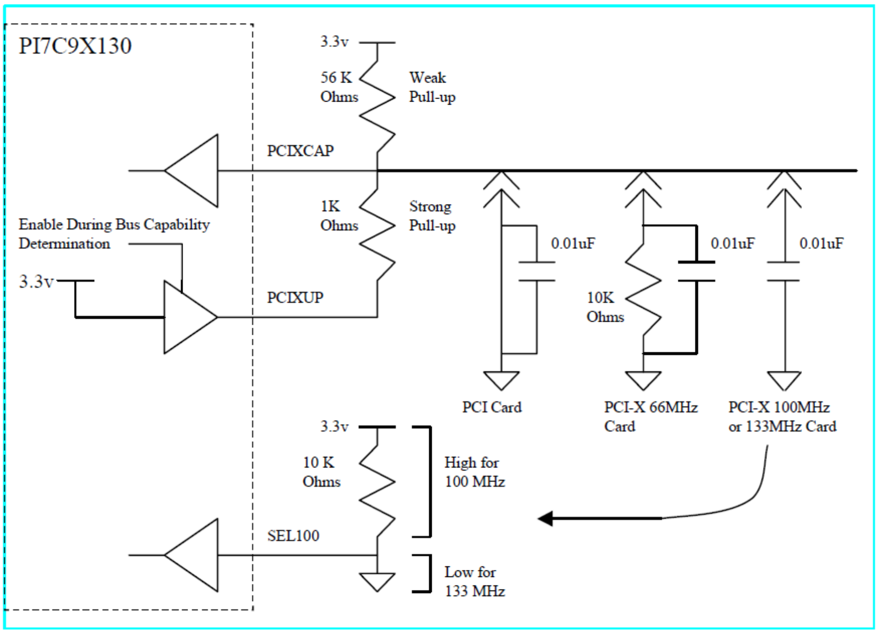
MSK_IN pin is multi-functional. One of its functions is the input for shift registers to enable and disable the corresponding CLKOUT signal (Please refer to register description: SECONDARY CLOCK AND CLKRUN CONTROL REGISTER – OFFSET A4h). This is a legacy PCI function. Please see the legacy application circuit below. However, most designers do not use this legacy circuit anymore. Instead, EEPROM or firmware is used to configure the registers. In the case, MSK_IN must be pulled down through 1k-ohm resistor or grounded to keep voltage stable. Keeping the voltage level stable will ensure no interference occurs to the shift registers.



## 18 How should PCIXCAP and PCIXUP signals be handled?

Please refer to the application circuit diagram below based on the PCI bus requirement.

Figure 3-4 PCI / PCI-X Selection



## 19 How should JTAG signals be handled if JTAG is not used?

TRST# signal should be pulled down through a 330ohm resistor.
TMS and TDI signals should be pulled up through a 5.1k-ohm resistor.
TCK signal should be pulled down through a 5.1k-ohm resistor.
TDO signal should be left floating.

## 20 Which PCI Bus signals should be pulled up?

Per PCI Specification 4.3.3, the following PCI Bus signals should be pulled up through a 5.1k-ohm resistor: FRAME#, TRDY#, IRDY#, DEVSEL#, STOP#, SERR#, PERR#, LOCK#, INTA#, INTB#, INTC#, INTD#, REQ64#, and ACK64#.

Pull-up resistors are required on main board, but not on add-on card.
In forward mode, PI7C9X130 requires pull-up resistors on PCI side. In reverse mode, PI7C9X130 requires pull-up resistors on CPU side.

## 21 Should AD, CBE, and PAR signals on PCI Bus be pulled up too?

No, AD, CBE, and PAR signals do not require pull-up resistors. Their states are ensured by PCI bus parking.

## 22 If LOCK_L and PME_L signals are not used, can they be left floating?

LOCK_L and PME_L signals must be pulled up through a 5.1k-ohm resistor whether they are used or not.

## 23 If INTA_L, INTB_L, INTC_L, and INTD_L signals are not used, can they be left floating?

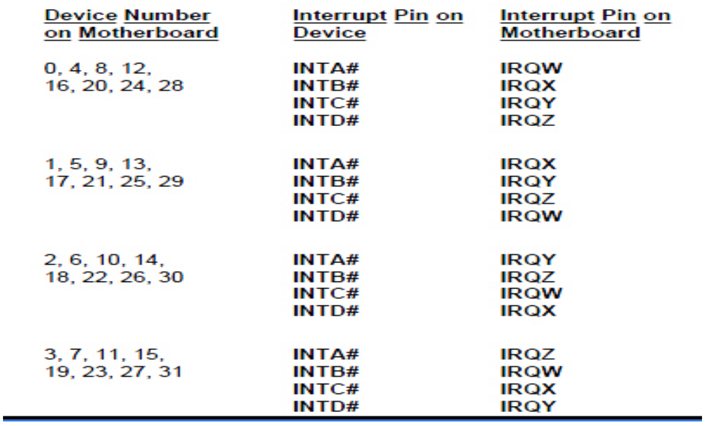
These interrupt signals must be pulled up through a 5.1k-ohm resistor whether they are used or not.

## 24 If there are multiple PCI devices connected, how should they be connected to INTA#, INTB#, INTC#, and INTD# signals?

INTA#, INTB#, INTC#, and INTD# signals are shared by multiple PCI devices in a daisy chain. It depends on which AD signal is used on IDSEL pin. Please refer to the mapping table below, and PCI Specification 2.2.6 for more information.

In Forward mode, INTA#,INTB#, INTC#, INTD# of PI7C9X130 are equivalent to the interrupt pins on the Motherboard side in the table below - IRQW, IRQX, IRQY, and IRQZ. INTA#,INTB#, INTC#, INTD# on the PCI bus of PI7C9X130 are equivalent to the interrupt pins on the device side in the table below - INTA#,INTB#, INTC#, and INTD#.
In Reverse mode, INTA#,INTB#, INTC#, INTD# of PI7C9X130 are equivalent to the interrupt pins on the device side in the table below - INTA#,INTB#, INTC#, and INTD#.

To clarify, the Motherboard side in the table below means the side close to the CPU. If PI7C9X130 is built on an add-on card, INTA#,INTB#, INTC#, and INTD# are connected to the corresponding INTA#,INTB#, INTC#, and INTD# on gold fingers.



| Device Number on Motherboard | Interrupt Pin on Device | Interrupt Pin on Motherboard |
|---|---|---|
| 0, 4, 8, 12,<br>16, 20, 24, 28 | INTA#<br>INTB#<br>INTC#<br>INTD# | IRQW<br>IRQX<br>IRQY<br>IRQZ |
| 1, 5, 9, 13,<br>17, 21, 25, 29 | INTA#<br>INTB#<br>INTC#<br>INTD# | IRQX<br>IRQY<br>IRQZ<br>IRQW |
| 2, 6, 10, 14,<br>18, 22, 26, 30 | INTA#<br>INTB#<br>INTC#<br>INTD# | IRQY<br>IRQZ<br>IRQW<br>IRQX |
| 3, 7, 11, 15,<br>19, 23, 27, 31 | INTA#<br>INTB#<br>INTC#<br>INTD# | IRQZ<br>IRQW<br>IRQX<br>IRQY |

## 25 What are the differences of INTA_L, INTB_L, INTC_L, and INTD_L in forward mode and reverse mode?

In Forward mode, INTA_L, INTB_L, INTC_L, and INTD_L are input signals. Interrupt signals of downstream PCI devices should be connected to them.

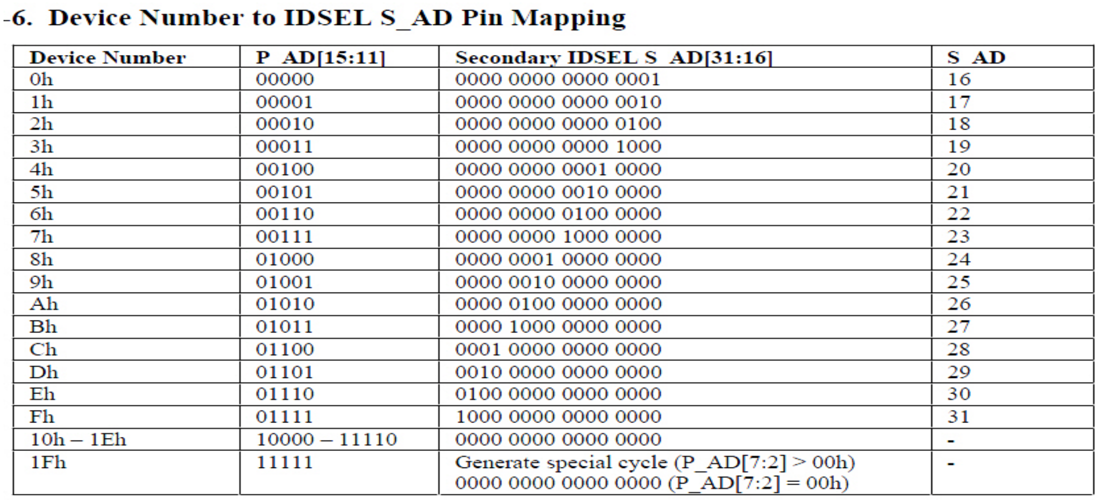
In Reverse mode, INTA_L, INTB_L, INTC_L, and INTD_L are output signals, and should be connected to interrupt signals of CPU.

## 26 What are the differences of PERST_L and RESET_L in forward mode and reverse mode?

Forward mode:
PERST_L is an input signal. CPU drives the reset signal to PI7C9X130.
RESET_L is an output signal to control the downstream PCI devices connected to PI7C9X130.
Reverse mode:
RESET_L is an input signal. CPU drives the reset signal to PI7C9X130.
PERST_L is an output signal to control the downstream PCIe device connected to PI7C9X130.

## 27 How should the PCI device's IDSEL signal be connected?

Forward mode:
PI7C9X130's IDSEL should be grounded through a 1k ohm resistor. IDSEL signals of the connected PCI devices should be connected to one of AD16-AD31 signals per PCI Specification. The AD signal used determines the PCI device's device number. Please refer to the table below:



-6. Device Number to IDSEL S_AD Pin Mapping

| Device Number | P AD[15:11] | Secondary IDSEL S AD[31:16] | S AD |
|---|---|---|---|
| 0h | 00000 | 0000 0000 0000 0001 | 16 |
| 1h | 00001 | 0000 0000 0000 0010 | 17 |
| 2h | 00010 | 0000 0000 0000 0100 | 18 |
| 3h | 00011 | 0000 0000 0000 1000 | 19 |
| 4h | 00100 | 0000 0000 0001 0000 | 20 |
| 5h | 00101 | 0000 0000 0010 0000 | 21 |
| 6h | 00110 | 0000 0000 0100 0000 | 22 |
| 7h | 00111 | 0000 0000 1000 0000 | 23 |
| 8h | 01000 | 0000 0001 0000 0000 | 24 |
| 9h | 01001 | 0000 0010 0000 0000 | 25 |
| Ah | 01010 | 0000 0100 0000 0000 | 26 |
| Bh | 01011 | 0000 1000 0000 0000 | 27 |
| Ch | 01100 | 0001 0000 0000 0000 | 28 |
| Dh | 01101 | 0010 0000 0000 0000 | 29 |
| Eh | 01110 | 0100 0000 0000 0000 | 30 |
| Fh | 01111 | 1000 0000 0000 0000 | 31 |
| 10h – 1Eh | 10000 – 11110 | 0000 0000 0000 0000 | - |
| 1Fh | 11111 | Generate special cycle (P_AD[7:2] > 00h)<br>0000 0000 0000 0000 (P_AD[7:2] = 00h) | - |

Reverse mode:
IDSEL signal should be connected to the IDSEL signal of the PCI controller on the CPU side.

## 28 How should the decoupling capacitor be placed in TN/TP pins of PCIe port?

Per PCI Express Specification, the egress signals of PCIe port require 0.1μF decoupling capacitor to be placed. Therefore, PI7C9X130's TN/TP pins require 0.1μF decoupling capacitor. Similarly, the egress signals of the PCIe port of the connected PCIe device require 0.1μF decoupling capacitor.

## 29 How should the REFCLKN/REFCLKP pins of PCIe port of PI7C9X130 be connected?

The REFCLKN/REFCLKP pins require 0.1μF decoupling capacitor. In testing the signals, only AC characteristics should be tested to comply with PCIe Specification. There is no requirement to testing DC characteristics.

## 30 Does PI7C9X130's PCIe port support asynchronous clock source?

Yes, PI7C9X130 supports asynchronous differential 100Mhz clock sources on two sides of the PCIe port, provided the PCIe device on the other side supports asynchronous clock sources too. It should be noted that not all PCIe devices support asynchronous clock sources.

## 31 Which 100MHz differential clock generators do you recommend?

We recommend Pericom's own PI6C557-03, PI6C20400, and PI6C20800 models, which are specifically designed to provide PCIe differential clock sources. PI6C557-03 provides 2-channel output, while PI6C20400 provides 4-channel and PI6C20800 provides 8-channel. PI6C557-10 is capable of PCIe differential clock output and 33MHz clock output simultaneously.

Please contact Pericom FAEs for more information.

## 32 One of the CLKOUT output signals must be connected to FBCLKIN. The length of this trace must equal to other CLKOUT output signal traces?

Yes, all CLKOUT output traces must have the same length.

## 33 What should be done to CLKRUN_L pin if Clock Management function is disabled?

It is recommended that CLKRUN_L pin is pulled down through a 1k-ohm resistor. Please refer to PCI Mobile Design Guide and PCI Express Card Electromechanical Specification for more details on Clock Management.

## 34 What should be done to CLKIN, CLKOUT, and FBCLKIN pins in Reverse Mode?

In Reverse Mode, FBCLKIN is the PCI clock input, which should get PCI clock source from CPU or system, i.e. 33MHz. CLKIN pin is not used, and should be pulled down through a 1k-ohm resistor. CLKOUT is not used, and should be left floating.

## 35 What should be done to CFN_L pin in Reverse Mode?

In Reverse Mode, CFN_L pin should be pulled up through a 1k-ohm resistor.

## 36 What should be done to REQ and GNT pins in Reverse Mode?

In Reverse Mode, REQ0 is an input pin, and should be connected to GNT pin on CPU side. GNT0 is an output pin, and should be connected to REQ on CPU side.

## 37 What should be done to PME_L pin in Reverse Mode?

In Reverse Mode, PME_L is an output pin, and should be connected to PME_L pin on CPU side through a 5.1k-ohm pull-up resistor.

## 38 Why is the software unable to detect the Bridge in Non-Transparent Mode?

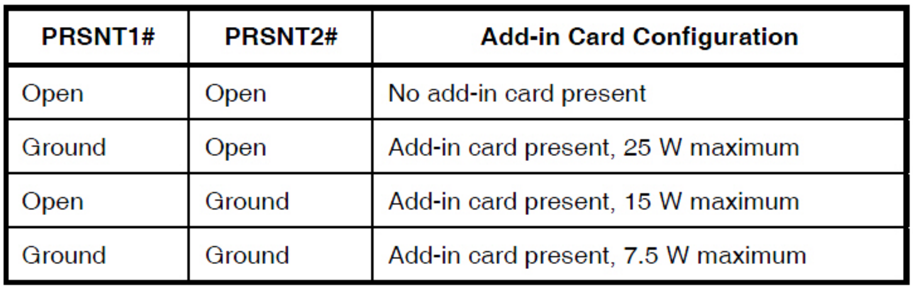
In Non-Transparent Mode, the “class code” of the Bridge is not one of a regular device. And therefore, the BIOS does not list the bridge in the device list.

In this mode, in addition to the electrical handling in board design to isolate two sides of the bridge, software driver is needed and bridge needs to be configured correctly for the system to work correctly.

## 39 How should PRSNT1# and PRSNT2# pins in PCI slot be handled?

PRSNT1# and PRSNT2# pins in PCI slot are used by add-in card to indicate the presence status. On the Motherboard, they should be pulled up. On the add-in card, they should be configured based on the card’s power status.

Table 4-12: Present Signal Definitions



| PRSNT1# | PRSNT2# | Add-in Card Configuration |
|---|---|---|
| Open | Open | No add-in card present |
| Ground | Open | Add-in card present, 25 W maximum |
| Open | Ground | Add-in card present, 15 W maximum |
| Ground | Ground | Add-in card present, 7.5 W maximum |

Please refer to PCI Specification 4.4.1 for more information.

## 40 Are there any other documents I can refer to in addition to the datasheet to clarify the questions I have?

PI7C9X130 is a standard PCIe Bridge and compliant with standard PCI/PCIe specifications. For more information, please refer to the following specifications:

- PCI Express Base Specification
- PCI Express Card Electromechanical Specification
- PCI Local Bus Specification
- PCI Express to PCI/PCI-X Bridge Specification