



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





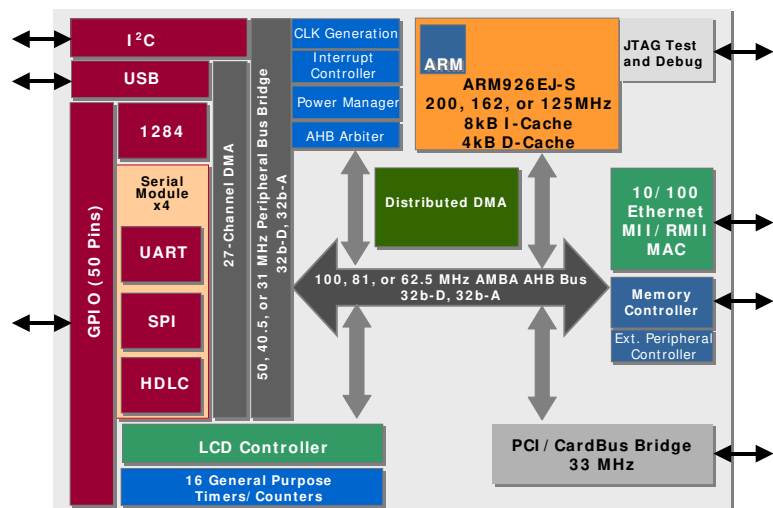
NS9750B-0 Datasheet

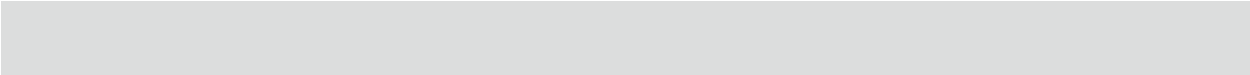
The Digi NS9750 is a single chip 0.13 μ m CMOS network-attached processor. The CPU is the ARM926EJ-Score with MMU, DSP extensions, Jazelle Java accelerator, and 8 kB of instruction cache and 4 kB of data cache in a Harvard architecture. NS9750 runs up to 200 MHz, with a 100 MHz system and memory bus and 50 MHz peripheral bus. NS9750 operates at a 1.5V core and 3.3V I/O ring voltages.

With its extensive set of I/O interfaces, Ethernet high-speed performance and processing capacity, NS9750 is the most capable of highly integrated 32-bit network-attached processors available. NS9750 is designed specifically for use in high-performance intelligent networked devices and Internet appliances including high-performance/ low-latency remote I/O, intelligent networked information displays, and streaming and surveillance cameras.

The NS9750 is a member of the award-winning NET+ARM family of system-on-chip (SOC) solutions for embedded systems.

NS9750 offers a connection to an external bus expansion module as well as a glueless connection to SDRAM, PC100 DIMM, Flash, EEPROM, and SRAM memories, and an external bus expansion module. It includes a versatile embedded LCD controller supporting up to 16M color TFT or 3375 color STN. NS9750 features a PCI/ CardBus port as well as a USB port for applications requiring WLAN, external storage, or external sensors, imagers, or scanners. Four multi-function serial ports, an I²C port, and 1284 parallel port provide a standard glueless interface to a variety of external peripherals. NS9750





features up to 50 general purpose I/ O (GPIO) pins and highly-configurable power management with sleep mode.

NET+ARM processors are the foundation for the NET+Works® family of integrated hardware and software solutions for device networking. These comprehensive platforms include drivers, operating systems, networking software, development tools, APIs, and complete development boards.

Using NS9750 and associated Net+Works packages allows system designers to achieve dramatic time-to-market reductions with pre-integrated and tested NET+ARM hardware, NET+Works software, and tools. Product unit costs are reduced dramatically with complete system-on-chip, including Ethernet, display support, a robust peripheral set, and the processing headroom to meet the most demanding applications. Customers save engineering resources, as no network development is required. Companies will reduce their design risk with a fully integrated and tested solution.

A complete NET+Works development package includes ThreadX™picokernel RTOS, Green Hills™ MULTI® 2000 IDE or Microcross GNU X-Tools™, drivers, networking protocols and services with APIs, NET+ARM-based development board, Digi-supplied utilities, Integrated File System, JTAG In Circuit Emulator (ICE), and support for Boundary Scan Description Language (BSDL). One year software maintenance and technical support is available.

Contents

NS9750 Features	1
System-level interfaces.....	4
System configuration.....	5
System boot.....	8
Reset	9
System Clock	11
USB clock.....	12
NS9750 pinout and signal descriptions	13
System Memory interface	13
System Memory interface signals	17
Ethernet interface.....	19
Clock generation/ system pins.....	20
bist_en_n, pll_test_n, and scan_en_n.....	21
PCI interface	21
PCI/ CardBus signals	24
GPIO MUX	26
LCD module signals.....	31
I2C interface	35
USB Interface.....	35
JTAG interface for ARM core/ boundary scan	35
Reserved pins	37
Power ground	38
Address and register maps	39
System address map	39
BBus peripheral address map	40
Electrical characteristics	41
Absolute maximum ratings.....	41
Recommended operating conditions	41
Maximum power dissipation	42
Typical power dissipation	42
DC electrical characteristics.....	43
Inputs.....	43
Outputs.....	44
Power sequencing.....	44
AC Characteristics	45
Memory timing	45
Ethernet timing.....	56
PCI timing.....	58
I2C timing.....	62
LCD timing.....	63
SPI timing.....	67
IEEE 1284 timing.....	70
USB timing.....	71



Reset and hardware strapping timing 73
JTAG timing 74
Clock timing 75
Packaging 77

NS9750 Features

32-bit ARM926EJ-S RISC processor

- 125 to 200 MHz
- 5-stage pipeline with interlocking
- Harvard architecture
- 8 kB instruction cache and 4 kB data cache
- 32-bit ARM and 16-bit Thumb instruction sets. Can be mixed for performance/ code density tradeoffs
- MMU to support virtual memory-based OSs such as Linux, WinCE/ Pocket PC, VxWorks, others
- DSP instruction extensions, improved divide, single cycle MAC
- ARM Jazelle, 1200CM (coffee marks) Java accelerator
- EmbeddedICE-RT debug unit
- JTAG boundary scan, BSDL support

External system bus interface

- 32-bit data, 32-bit internal address bus, 28-bit external address bus
- Glueless interface to SDRAM, SRAM, EEPROM, buffered DIMM, Flash
- 4 static and 4 dynamic memory chip selects
- 0-63 wait states per chip select
- Self-refresh during system sleep mode
- Automatic dynamic bus sizing to 8 bits, 16 bits, 32 bits
- Burst mode support with automatic data width adjustment
- Two external DMA channels for external peripheral support

System Boot

- High-speed boot from 8-bit, 16-bit, or 32-bit ROM or Flash
- Hardware-supported low cost boot from serial EEPROM through SPI port (patent pending)

High performance 10/100 Ethernet MAC

- 10/ 100 Mbps MII/ RMI PHY interfaces
- Full-duplex or half-duplex
- Station, broadcast, or multicast address filtering
- 2 kB RX FIFO
- 256 byte TX FIFO with on-chip buffer descriptor ring
 - Eliminates underruns and decreases bus traffic
- Separate TX and RX DMA channels
- Intelligent receive-side buffer size selection
- Full statistics gathering support
- External CAM filtering support

PCI/CardBus port

- PCI v2.2, 32-bit bus, up to 33 MHz bus speed
- Programmable to:
 - PCI device mode
 - PCI host mode:
 - Supports up to 3 external PCI devices
 - Embedded PCI arbiter or external arbiter
- CardBus host mode

Flexible LCD controller

- Supports most commercially available displays:
 - Active Matrix color TFT displays—Up to 24bpp direct 8:8:8 RGB; 16M colors
 - Single and dual panel color STN displays—Up to 16bpp 4:4:4 RGB; 3375 colors
 - Single and dual-panel monochrome STN displays—1, 2, 4bpp palettized gray scale
- Formats image data and generates timing control signals
- Internal programmable palette LUT and grayscale support different color techniques
- Programmable panel-clock frequency

USB ports

- USB v.2.0 full speed (12 Mbps) and low speed (1.5 Mbps)
- Configurable to device or OHCI host
 - USB host is bus master
 - USB device supports one bidirectional control endpoint and 11 unidirectional endpoints
- All endpoints supported by a dedicated DMA channel; 13 channels total
- 20 byte RX FIFO and 20 byte TX FIFO

Serial ports

- 4 serial modules, each independently configurable to UART mode, SPI master mode, or SPI slave mode
- Bit rates from 75 bps to 921.6 kbps: asynchronous x16 mode
- Bit rates from 1.2 kbps to 6.25 Mbps: synchronous mode
- UART provides:
 - High-performance hardware and software flow control
 - Odd, even, or no parity

- 5, 6, 7, or 8 bits
- 1 or 2 stop bits
- Receive-side character and buffer gap timers

- Internal or external clock support, digital PLL for RX clock extraction
- 4 receive-side data match detectors
- 2 dedicated DMA channels per module, 8 channels total
- 32 byte TX FIFO and 32 byte RX FIFO per module

I²C port

- I²C v.1.0, configurable to master or slave mode
- Bit rates: fast (400 kHz) or normal (100 kHz) with clock stretching
- 7-bit and 10-bit address modes
- Supports I²C bus arbitration

1284 parallel peripheral port

- All standard modes: ECP, byte, nibble, compatibility (also known as SPP or “Centronix”)
- RLE (run length encoding) decoding of compressed data in ECP mode
- Operating clock from 100 kHz to 2 MHz

High performance multiple-master/distributed DMA system

- Intelligent bus bandwidth allocation (patent pending)
- System bus and peripheral bus

System bus:

- Every system bus peripheral is a bus master with a dedicated DMA engine

Peripheral bus:

- One 13-channel DMA engine supports USB device
 - 2 DMA channels support control endpoint
 - 11 DMA channels support 11 endpoints

- One 12-channel DMA engine supports:
 - 4 serial modules (8 DMA channels)
 - 1284 parallel port (4 DMA channels)

External peripheral:

- One 2-channel DMA engine supports external peripheral connected to memory bus
- Each DMA channel supports memory-to-memory transfers

Power management (patent pending)

- Power save during normal operation
 - Disables unused modules
- Power save during sleep mode
 - Sets memory controller to refresh
 - Disables all modules except selected wakeup modules
 - Wakeup on valid packets or characters

Vector interrupt controller

- Decreased bus traffic and rapid interrupt service
- Hardware interrupt prioritization

General purpose timers/counters

- 16 independent 16-bit or 32-bit programmable timers or counters
 - Each with an I/O pin
- Mode selectable into:
 - Internal timer mode
 - External gated timer mode
 - External event counter
- Can be concatenated
- Resolution to measure minute-range events

- Source clock selectable: internal clock or external pulse event
- Each can be individually enabled/ disabled

System timers

- Watchdog timer
- System bus monitor timer
- System bus arbiter timer
- Peripheral bus monitor timer

General purpose I/O

- 50 programmable GPIO pins (muxed with other functions)
- Software-readable powerup status registers for every pin for customer-defined bootstrapping

External interrupts

- 4 external programmable interrupts
 - Rising or falling edge-sensitive
 - Low level- or high level-sensitive

Clock generator

- Low cost external crystal
- On-chip phase locked loop (PLL)
- Software programmable PLL parameters
- Optional external oscillator
- Separate PLL for USB

Operating grades/Ambient temperatures

- 200 MHz: 0 –70° C
- 162 MHz: -40 –+85° C
- 125 MHz: 0 –70° C

System-level interfaces

Figure 1 shows the NS9750 system-level hardware interfaces, which are detailed after the figure.

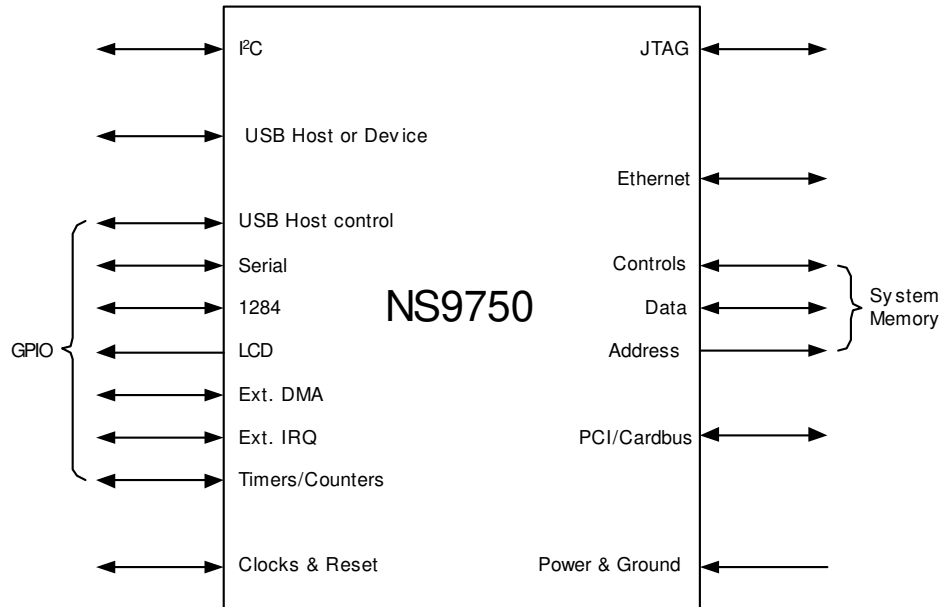


Figure 1: System-level hardware interfaces

NS9750 interfaces

- Ethernet MII/ RMI interface to an external PHY
- System Memory interface
 - Glueless connection to SDRAM
 - Glueless connection to buffered PC100 DIMM
 - Glueless connection to SRAM
 - Glueless connection to Flash memory or ROM
- PCI muxed with CardBus interface
- USB host or device interface
- I²C interface
- 50 GPIO pins muxed with:
 - Four 8-pin-each serial ports, each programmable to UART or SPI
 - 1284 port
 - Up to 24-bit TFT or STN color and monochrome LCD controller
 - Two external DMA channels
- Four external interrupt pins programmed to rising or falling edge, or to high or low level
- Sixteen 16-bit or 32-bit programmable timers or counters
- Two control signals to support USB host
- JTAG development interface
- Clock interfaces for crystal or external oscillator
 - System clock
 - USB clock
- Clock interface for optional LCD external oscillator
- Power and ground

System configuration

The PLL and other system settings can be configured at powerup before the CPU boots. External pins configure the necessary control register bits at powerup. External pulldown resistors can be used to configure the PLL and system configuration registers depending on the application. The recommended value is 2.2k ohm to 2.4k ohm.

Table 1 indicates how each bit is used to configure the powerup settings, where 1 indicates the internal pullup resistor and 0 indicates an external pulldown resistor. Table 2 shows PLL ND[4:0] multiplier values.

Pin name	Configuration bits
rtck	PCI arbiter configuration 0 External PCI arbiter 1 Internal PCI arbiter
boot_strap[0]	Chip select 1 byte_lane_enable_n/write_enable_n configuration bootstrap select 0 write_enable_n for byte-wide devices (default) 1 byte_lane_enable_n (2.4K pulldown added)
boot_strap[4:3]	Chip select 1 data width bootstrap select 00 16 bits 01 8 bits 11 32 bits
boot_strap[2]	Memory interface read mode bootstrap select Note: An external pulldown resistor must be used to select command delayed mode. Clock delayed mode is reserved for future use. 0 Command delayed mode Commands are launched on a 90-degree phase-shifted AHB clock, and AHB clock is routed to the external dynamic memory. 1 Clock delayed mode Reserved for future use.
boot_strap[1]	CardBus mode bootstrap select 0 CardBus mode 1 PCI mode
gpio[49]	Chip select polarity 0 Active high 1 Active low
gpio[44]	Endian mode 0 Big Endian 1 Little Endian
reset_done	Bootup mode 0 Boot from SDRAM using serial SPI EEPROM 1 Boot from flash/ROM

Table 1: Configuration pins— Bootstrap initialization

Pin name	Configuration bits										
gpio[19]	PLL BP (PLL bypass) 0 PLL bypassed 1 PLL not bypassed										
gpio[17], gpio[12], gpio[10], gpio [8], gpio[4]	PLL ND[4:0] (PLL multiplier, ND + 1) See Table 2: PLL ND[4:0].										
gpio[2], gpio[0]	PLL FS[1:0] (PLL frequency select) <table border="1"> <thead> <tr> <th>FS</th> <th>Divide by</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>1</td> </tr> <tr> <td>01</td> <td>2</td> </tr> <tr> <td>10</td> <td>4</td> </tr> <tr> <td>11</td> <td>8</td> </tr> </tbody> </table>	FS	Divide by	00	1	01	2	10	4	11	8
FS	Divide by										
00	1										
01	2										
10	4										
11	8										

Table 1: Configuration pins— Bootstrap initialization

Register configuration: gpio 17, 12, 10, 8, 4	Multiplier
1 1 0 1 0	32
0 0 1 0 0	31
1 1 0 0 0	30
1 1 0 0 1	29
1 1 1 1 0	28
1 1 1 1 1	27
1 1 1 0 0	26
1 1 1 0 1	25
1 0 0 1 0	24
1 0 0 1 1	23
1 0 0 0 0	22
1 0 0 0 1	21
1 0 1 1 0	20
1 0 1 1 1	19
1 0 1 0 0	18
1 0 1 0 1	17
0 1 0 1 0	16
0 1 0 1 1	15

Table 2: PLL ND[4:0]

Register configuration: gpio 17, 12, 10, 8, 4	Multiplier
0 1 0 0 0	14
0 1 0 0 1	13
0 1 1 1 0	12
0 1 1 1 1	11
0 1 1 0 0	10
0 1 1 0 1	9
0 0 0 1 0	8
0 0 0 1 1	7
0 0 0 0 0	6
0 0 0 0 1	5
0 0 1 1 0	4
0 0 1 1 1	3
0 0 1 0 0	2
0 0 1 0 1	1

Table 2: PLL ND[4:0]

There are 32 additional GPIO pins that are used to create a general purpose, user-defined ID register. These are external signals that are registered at powerup.

gpio[41]	gpio[40]	gpio[39]	gpio[38]
gpio[37]	gpio[36]	gpio[35]	gpio[34]
gpio[33]	gpio[32]	gpio[31]	gpio[30]
gpio[29]	gpio[28]	gpio[27]	gpio[26]
gpio[25]	gpio[23]	gpio[22]	gpio[21]
gpio[18]	gpio[16]	gpio[15]	gpio[14]
gpio[13]	gpio[11]	gpio[9]	gpio[7]
gpio[6]	gpio[5]	gpio[3]	gpio[1]

Read these signals for general purpose status information.

System boot

There are two ways to boot the NS9750 system (see Figure 2):

- From a fast Flash over the system memory bus
- From an inexpensive, but slower, serial EEPROM through SPI port B

Both boot methods are glueless. The bootstrap pin, RESET_DONE_n, is used to indicate where to boot on a system powerup. Flash boot can be done from 8-bit, 16-bit, or 32-bit ROM or Flash.

Serial EEPROM boot is supported by NS9750 hardware. A configuration header in the EEPROM specifies total number of words to be fetched from EEPROM, as well as a system memory configuration and a memory controller configuration. The boot engine configures the memory controller and system memory, fetches data from low-cost serial EEPROM, and writes the data to external system memory, holding the CPU in reset.

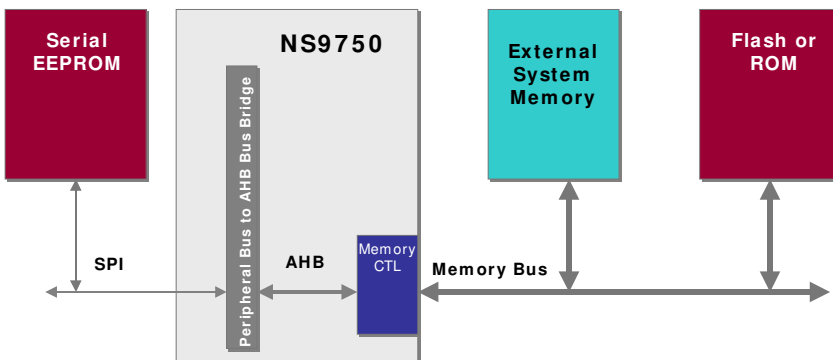


Figure 2: Two methods of booting NS9750 system

Reset

Master reset using an external reset pin resets NS9750. Only the AHB bus error status registers retain their values; software read resets these error status registers. The input reset pin can be driven by a system reset circuit or a simple power-on reset circuit.

RESET_DONE as an input

Used at bootup only:

- When set to 0, the system boots from SDRAM through the serial SPI EEPROM.
- When set to 1, the system boots from Flash/ ROM. This is the default.

RESET_DONE as an output

Sets to 1, per Step 6 in the boot sequence:

If the system is booting from serial EEPROM through the SPI port, the boot program must be loaded into the SDRAM before the CPU is released from reset. The memory controller is powered up with `dy_cs_n[0]` enabled with a default set of SDRAM configurations. The default address range for `dy_cs_n[0]` is from `0x0000 0000`. The other chip selects are disabled.

Boot sequence

- 1 When the system reset turns to inactive, the reset signal to the CPU is still held active.
- 2 An I/O module on the peripheral bus (BBus) reads from a serial ROM device that contains the memory controller settings and the boot program.
- 3 The BBus-to-AHB bridge requests and gets the system bus.
- 4 The memory controller settings are read from the serial EEPROM and used to initialize the memory controller.
- 5 The BBus-to-AHB bridge loads the boot program into the SDRAM, starting at address 0.
- 6 The reset signal going to the CPU is released once the boot program is loaded. `RESET_DONE` is now set to 1.
- 7 The CPU begins to execute code from address `0x0000 0000`.

Figure 3 shows a sample reset circuit.

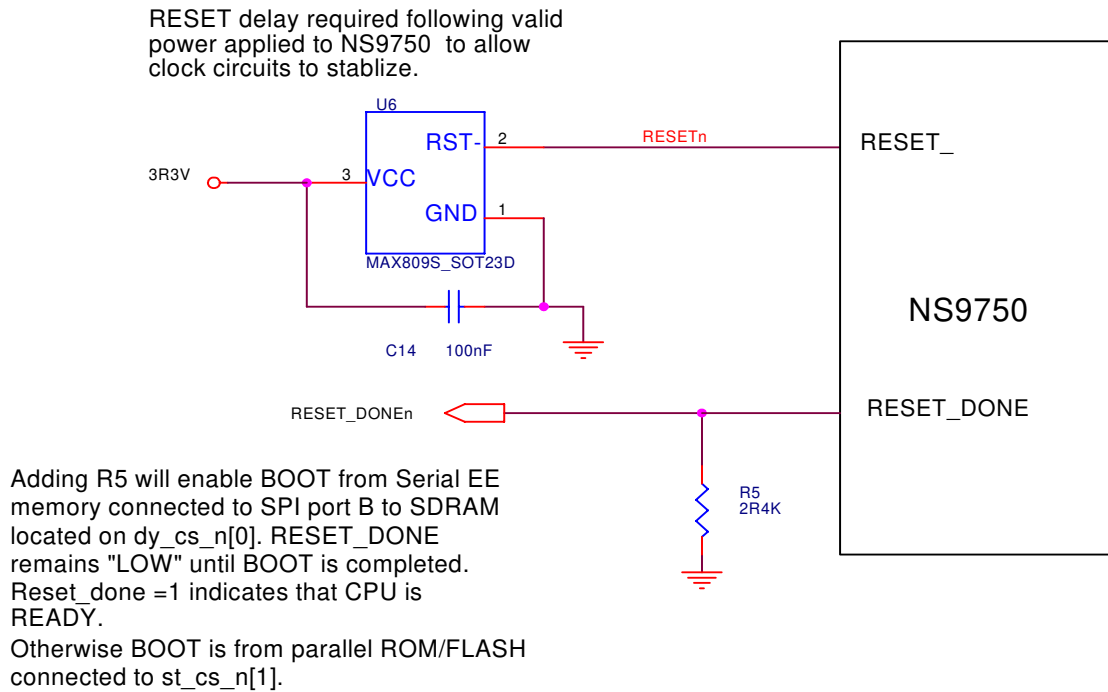


Figure 3: Sample reset circuit

You can use one of four software resets to reset the NS9750. Select the reset by setting the appropriate bit in the appropriate register:

- Watchdog timer can issue reset upon Watchdog timer expiration.
- AHB bus arbiter can issue reset upon AHB bus arbiter timer expiration.
- AHB bus monitor can issue reset upon AHB bus monitor timer expiration.
- Software reset can reset individual internal modules or all modules (except memory and CPU).

Hardware reset duration is 4ms for PLL to stabilize. Software reset duration depends on speed grade, as shown in Table 3.

Speed grade	CPU clock cycles	Duration
200 MHz	128	640 ns
162 MHz	128	790 ns
125 MHz	128	1024 ns

Table 3: Software reset duration

The minimum reset pulse width is 10 crystal clocks.

System Clock

The system clock is provided to NS9750 by either a crystal or an external oscillator; Table 4 shows sample clock frequency settings for each chip speed grade.

Speed	cpu_clk	hclk (main bus)	bbus_clk
200 MHz	200 (199.0656)	99.5328	49.7664
162 MHz	162.2016	81.1008	40.5504
125 MHz	125.3376	62.6688	31.3344

Table 4: Sample clock frequency settings with 29.4912 MHz crystal

If an oscillator is used, it must be connected to the x1_sys_osc input (C8 pin) on the NS9750. If a crystal is used, it must be connected with a circuit such as the one shown in Figure 4.

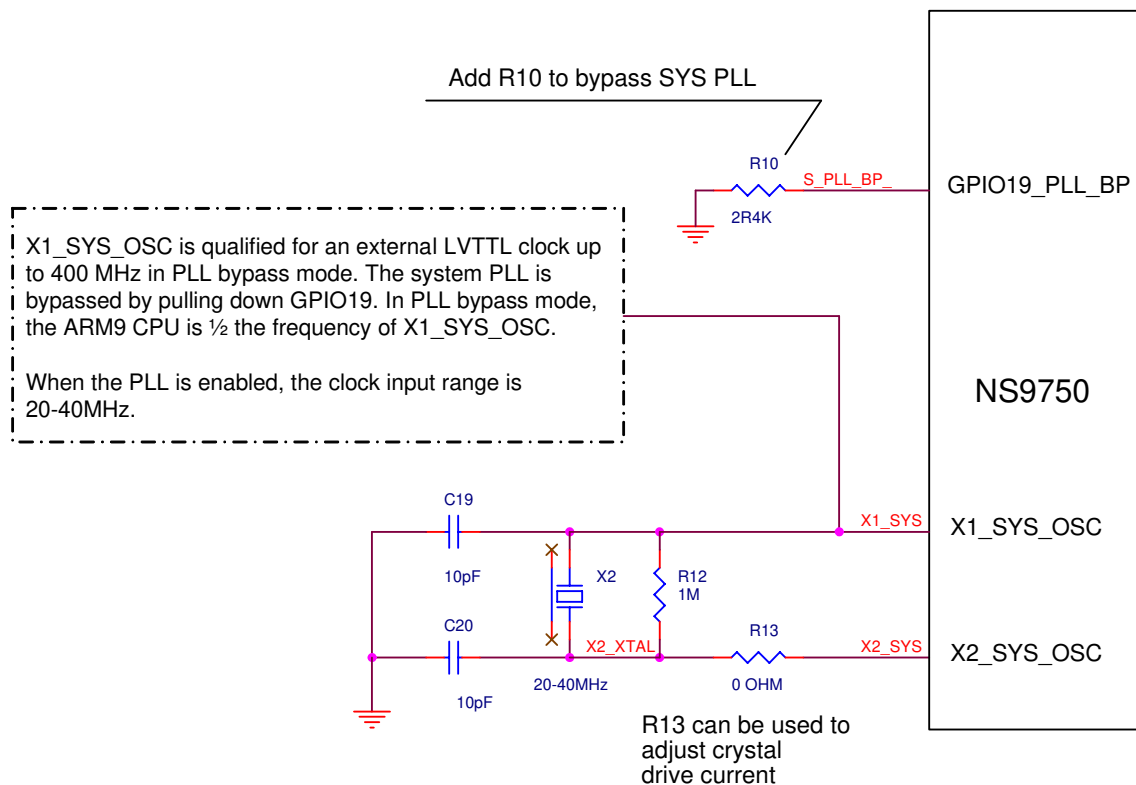


Figure 4: NS9750 system clock

The PLL parameters are initialized on powerup reset and can be changed by software from f_{max} to $1/2 f_{max}$. For a 200 MHz grade, then, the CPU may change from 200 MHz to 100 MHz, the AHB system bus may change from 100 MHz to 50 MHz, and the peripheral BBus may change from 50 MHz

to 25 MHz. If changed by software, the system resets automatically after the PLL stabilizes (approximately 4 ms).

The system clock provides clocks for CPU, AHB system bus, peripheral BBus, PCI/ CardBus, LCD, timers, memory controller, and BBus modules (serial modules and 1284 parallel port).

The Ethernet MAC uses external clocks from a MII PHY or a RMII PHY. For a MII PHY, these clocks are input signals: rx_clk on pin T3 for receive clock and tx_clk on pin V3 for transmit clock. For a RMII, there is only one clock, and it connects to the rx_clk on pin T3. In this case, the transmit clock tx_clk, pin V3, should be tied low.

PCI/ CardBus, LCD controller, serial modules (UART, SPI), and 1284 port can optionally use external clock signals.

USB clock

USB is clocked by a separate PLL driven by an external 48 MHz crystal, or it can be driven directly by an external 48 MHz oscillator. Figure 5 shows a USB circuit.

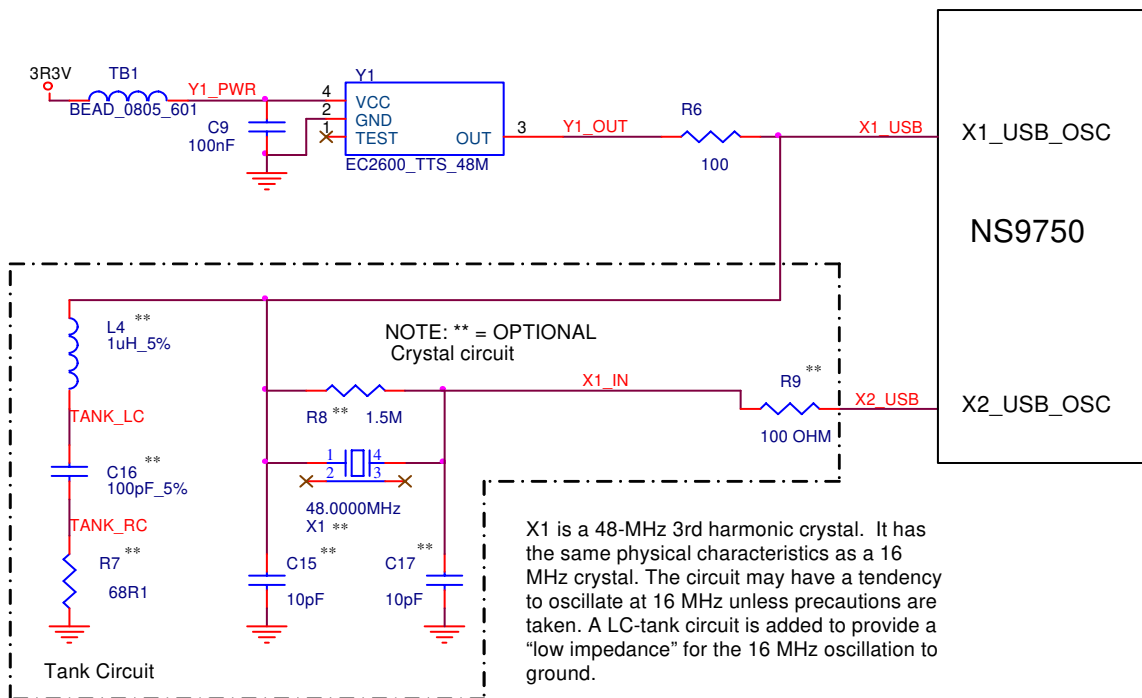


Figure 5: USB clock

NS9750 pinout and signal descriptions

Each pinout table applies to a specific interface, and contains the following information:

Heading	Description
Pin #	The pin number assignment for a specific I/O signal.
Signal Name	The pin name for each I/O signal. Some signals have multiple function modes and are identified accordingly. The mode is configured through firmware using one or more configuration registers. <i>_n</i> in the signal name indicates that this signal is active <i>low</i> .
U/D	U or D indicates whether the pin is a pullup resistor or a pulldown resistor: <ul style="list-style-type: none"> ■ U – Pullup (input current source) ■ D – Pulldown (input current sink) If no value appears, that pin is neither a pullup nor pulldown resistor.
I/O	The type of signal – input, output, or input/output.
OD (mA)	The output drive strength of an output buffer. The NS9750 uses one of three drivers: <ul style="list-style-type: none"> ■ 2 mA ■ 4 mA ■ 8 mA

More detailed signal descriptions are provided for selected modules.

System Memory interface

Pin #	Signal Name	OD		I/O	Description
		U/D	(mA)		
A21	addr[0]		8	O	Address bus signal
B20	addr[1]		8	O	Address bus signal
C19	addr[2]		8	O	Address bus signal
A20	addr[3]		8	O	Address bus signal
B19	addr[4]		8	O	Address bus signal
C18	addr[5]		8	O	Address bus signal
A19	addr[6]		8	O	Address bus signal
A17	addr[7]		8	O	Address bus signal
C16	addr[8]		8	O	Address bus signal
B16	addr[9]		8	O	Address bus signal
A16	addr[10]		8	O	Address bus signal
D15	addr[11]		8	O	Address bus signal

Table 5: System Memory interface pinout

Pin #	Signal Name	U/D	OD (mA)	I/O	Description
C15	addr[12]		8	O	Address bus signal
B15	addr[13]		8	O	Address bus signal
A15	addr[14]		8	O	Address bus signal
C14	addr[15]		8	O	Address bus signal
B14	addr[16]		8	O	Address bus signal
A14	addr[17]		8	O	Address bus signal
A13	addr[18]		8	O	Address bus signal
B13	addr[19]		8	O	Address bus signal
C13	addr[20]		8	O	Address bus signal
A12	addr[21]		8	O	Address bus signal
B12	addr[22]		8	O	Address bus signal
C12	addr[23]		8	O	Address bus signal
D12	addr[24]		8	O	Address bus signal
A11	addr[25]		8	O	Address bus signal
B11	addr[26]		8	O	Address bus signal
C11	addr[27]		8	O	Address bus signal
G2	clk_en[0]		8	O	SDRAM clock enable
H3	clk_en[1]		8	O	SDRAM clock enable
G1	clk_en[2]		8	O	SDRAM clock enable
H2	clk_en[3]		8	O	SDRAM clock enable
A10	clk_out[0]		8	O	SDRAM reference clock. Connect to clk_in[0] using series termination.
A9	clk_out[1]		8	O	SDRAM clock
A5	clk_out[2]		8	O	SDRAM clock
A4	clk_out[3]		8	O	SDRAM clock
G26	data[0]		8	I/O	Data bus signal
H24	data[1]		8	I/O	Data bus signal
G25	data[2]		8	I/O	Data bus signal
F26	data[3]		8	I/O	Data bus signal
G24	data[4]		8	I/O	Data bus signal
F25	data[5]		8	I/O	Data bus signal
E26	data[6]		8	I/O	Data bus signal

Table 5: System Memory interface pinout

Pin #	Signal Name	U/D	OD (mA)	I/O	Description
F24	data[7]		8	I/O	Data bus signal
E25	data[8]		8	I/O	Data bus signal
D26	data[9]		8	I/O	Data bus signal
F23	data[10]		8	I/O	Data bus signal
E24	data[11]		8	I/O	Data bus signal
D25	data[12]		8	I/O	Data bus signal
C26	data[13]		8	I/O	Data bus signal
E23	data[14]		8	I/O	Data bus signal
D24	data[15]		8	I/O	Data bus signal
C25	data[16]		8	I/O	Data bus signal
B26	data[17]		8	I/O	Data bus signal
D22	data[18]		8	I/O	Data bus signal
C23	data[19]		8	I/O	Data bus signal
B24	data[20]		8	I/O	Data bus signal
A25	data[21]		8	I/O	Data bus signal
C22	data[22]		8	I/O	Data bus signal
D21	data[23]		8	I/O	Data bus signal
B23	data[24]		8	I/O	Data bus signal
A24	data[25]		8	I/O	Data bus signal
A23	data[26]		8	I/O	Data bus signal
B22	data[27]		8	I/O	Data bus signal
C21	data[28]		8	I/O	Data bus signal
A22	data[29]		8	I/O	Data bus signal
B21	data[30]		8	I/O	Data bus signal
C20	data[31]		8	I/O	Data bus signal
E1	data_mask[0]		8	O	SDRAM data mask signal
F2	data_mask[1]		8	O	SDRAM data mask signal
G3	data_mask[2]		8	O	SDRAM data mask signal
F1	data_mask[3]		8	O	SDRAM data mask signal
C5	clk_in[0]			I	SDRAM feedback clock. Connect to clk_out[0].
D2	clk_in[1]			I	Connect to GND
E3	clk_in[2]			I	Connect to GND

Table 5: System Memory interface pinout

Pin #	Signal Name	U/D	OD (mA)	I/O	Description
E2	clk_in[3]			I	Connect to GND
B4	byte_lane_sel_n[0]		8	O	Static memory byte_lane_enable[0] or write_enable_n[0] for byte-wide device signals
F4	byte_lane_sel_n[1]		8	O	Static memory byte_lane_enable[1] or write_enable_n[1] for byte-wide device signals
D1	byte_lane_sel_n[2]		8	O	Static memory byte_lane_enable[2] or write_enable_n[2] for byte-wide device signals
F3	byte_lane_sel_n[3]		8	O	Static memory byte_lane_enable[3] or write_enable_n[3] for byte-wide device signals
B5	cas_n		8	O	SDRAM column address strobe
A8	dy_cs_n[0]		8	O	SDRAM chip select signal
B8	dy_cs_n[1]		8	O	SDRAM chip select signal
A6	dy_cs_n[2]		8	O	SDRAM chip select signal
C7	dy_cs_n[3]		8	O	SDRAM chip select signal
C6	st_oe_n		8	O	Static memory output enable
D6	ras_n		8	O	SDRAM row address strobe
H1	dy_pwr_n		8	O	SyncFlash power down
B10	st_cs_n[0]		8	O	Static memory chip select signal
C10	st_cs_n[1]		8	O	Static memory chip select signal
B9	st_cs_n[2]		8	O	Static memory chip select signal
C9	st_cs_n[3]		8	O	Static memory chip select signal
B6	we_n		8	O	SDRAM write enable. Used for static and SDRAM devices.
J3	ta_strb	U		I	Slow peripheral transfer acknowledge

Table 5: System Memory interface pinout

System Memory interface signals

Table 6 describes the System Memory interface signals in more detail. All signals are internal to the chip. Figure 6, "SDRAM clock termination," on page 18, shows NS9750 SDRAM clock termination.

Name	I/O	Description
addr[27:0]	O	Address output. Used for both static and SDRAM devices. SDRAM memories use bits [14:0]; static memories use bits [25:0].
clk_en[3:0]	O	SDRAM clock enable. Used for SDRAM devices. Note: The clk_en signals are associated with the dy_cs_n signals.
clk_out[3:1]	O	SDRAM clocks. Used for SDRAM devices.
clk_out[0]	O	SDRAM clk_out[0] is connected to clk_in[0].
data[31:0]	I/O	Read data from memory. Used for the static memory controller and the dynamic memory controller.
data_mask[3:0]	O	Data mask output to SDRAMs. Used for SDRAM devices.
clk_in[3:1]	I	Feedback clocks. Used for SDRAM devices.
clk_in[0]	I	Feedback clock [0]. Always connects to clk_out[0].
byte_lane_sel_n[3:0]	O	Static memory byte lane select, active low, or write_enable_n for byte-wide devices.
cas_n	O	Column address strobe. Used for SDRAM devices.
dy_cs_n[3:0]	O	SDRAM chip selects. Used for SDRAM devices.
st_oe_n	O	Output enable for static memories. Used for static memory devices.
ras_n	O	Row address strobe. Used for SDRAM devices.
st_cs_n[3:0]	O	Static memory chip selects. Default active low. Used for static memory devices.
we_n	O	Write enable. Used for SDRAM and static memories.
ta_strb	I	<i>Slow peripheral transfer acknowledge</i> can be used to terminate static memory cycles sooner than the number of wait states programmed in the chip select setup register.

Table 6: System Memory interface signal descriptions

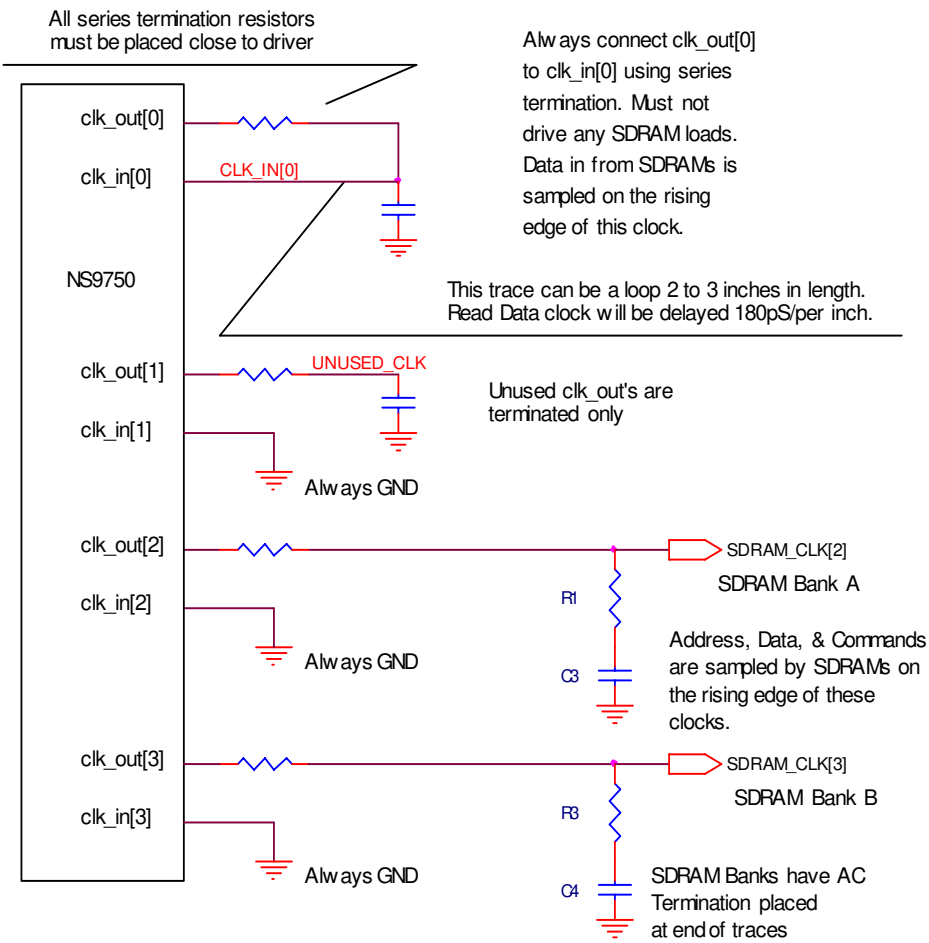


Figure 6: SDRAM clock termination

Ethernet interface

Pin #	Signal name		U/D	OD (mA)	I/O	Description	
	MII	RMII				MII	RMII
AB1	col	N/C			I	Collision	Pull low external to NS9750
AA2	crs	crs_dv			I	Carrier sense	Carrier sense
AC1	enet_phy_int_n	enet_phy_int_n	U		I	Ethernet PHY interrupt	Ethernet PHY interrupt
AA3	mdc	mdc		4	O	MII management interface clock	MII management interface clock
AB2	mdio	mdio	U	2	I/O	MII management data	MII management data
T3	rx_clk	ref_clk			I	Receive clock	Reference clock
V2	rx_dv	N/C			I	Receive data valid	Pull low external to NS9750
W1	rx_er	rx_er			I	Receive error	Optional signal; pull low to NS9750 if not used.
V1	rx_d[0]	rx_d[0]			I	Receive data bit 0	Receive data bit 0
U3	rx_d[1]	rx_d[1]			I	Receive data bit 1	Receive data bit 1
U2	rx_d[2]	N/C			I	Receive data bit 2	Pull low external to NS9750
U1	rx_d[3]	N/C			I	Receive data bit 3	Pull low external to NS9750
V3	tx_clk	N/C			I	Transmit clock	Pull low external to NS9750
AA1	tx_en	tx_en		2	O	Transmit enable	Transmit enable
Y3	tx_er	N/C		2	O	Transmit error	N/A
Y2	tx_d[0]	tx_d[0]		2	O	Transmit data bit 0	Transmit data bit 0
W3	tx_d[1]	tx_d[1]		2	O	Transmit data bit 1	Transmit data bit 1
Y1	tx_d[2]	N/C		2	O	Transmit data bit 2	N/A
W2	tx_d[3]	N/C		2	O	Transmit data bit 3	N/A

Table 7: Ethernet interface pinout

Clock generation/system pins

Pin #	Signal name	U/D	OD (mA)	I/O	Description
C8	x1_sys_osc			I	System clock crystal oscillator circuit input
B7	x2_sys_osc			O	System clock crystal oscillator circuit output
D9	x1_usb_osc			I	USB clock crystal oscillator circuit input. (Connect to GND if USB is not used.)
A7	x2_usb_osc			O	USB clock crystal oscillator circuit output
AC21	reset_done	U	2	I/O	CPU is enabled once the boot program is loaded. Reset_done is set to 1.
H25	reset_n	U		I	System reset input signal
AD20	bist_en_n			I	Enable internal BIST operation
AF21	pll_test_n			I	Enable PLL testing
AE21	scan_en_n			I	Enable internal scan testing
B18	sys_pll_dvdd				System clock PLL 1.5V digital power
A18	sys_pll_dvss				System clock PLL digital ground
B17	sys_pll_avdd				System clock PLL 3.3V analog power
C17	sys_pll_avss				System clock PLL analog ground
J2	lcdclk	U		I	External LCD clock input
T2	boot_strap[0]	U	2	I/O	Chip select 1 static memory byte_lane_enable_n, or write_enable_n for byte-wide devices bootstrap select
N3	boot_strap[1]	U	2	I/O	CardBus mode bootstrap select
P1	boot_strap[2]	U	2	I/O	Memory interface read mode bootstrap select
P2	boot_strap[3]	U	2	I/O	Chip select 1 data width bootstrap select
P3	boot_strap[4]	U	2	I/O	Chip select 1 data width bootstrap select

Table 8: Clock generation/system pins pinout

bist_en_n, pll_test_n, and scan_en_n

Table 9 is a truth/termination table for bist_en_n, pll_test_n, and scan_en_n.

	Normal operation	Arm debug	
pll_test_n	pull up	pull up	10K recommended
bist_en_n	pull down	pull up	10K pullup = debug 2.4K pulldown = normal
scan_en_n	pull down	pull down	2.4K recommended

Table 9: bist_en_n, pll_test_n, & scan_en_n truth/termination table

PCI interface

The PCI interface can be set to PCI host or PCI device (slave) using the pci_central_resource_n pin.

Notes:

- All output drivers for PCI meet the standard PCI driver specification.
- All table notes can be found after Table 11: CardBus IO muxed signals.

Pin #	Signal Name	U/D	OD (mA)	I/O	Description
J24	ad[0] ¹		N/A	I/O	PCI time-multiplexed address/data bus
H26	ad[1] ¹		N/A	I/O	PCI time-multiplexed address/data bus
J25	ad[2] ¹		N/A	I/O	PCI time-multiplexed address/data bus
J26	ad[3] ¹		N/A	I/O	PCI time-multiplexed address/data bus
K24	ad[4] ¹		N/A	I/O	PCI time-multiplexed address/data bus
K25	ad[5] ¹		N/A	I/O	PCI time-multiplexed address/data bus
K26	ad[6] ¹		N/A	I/O	PCI time-multiplexed address/data bus
L24	ad[7] ¹		N/A	I/O	PCI time-multiplexed address/data bus
L26	ad[8] ¹		N/A	I/O	PCI time-multiplexed address/data bus
M24	ad[9] ¹		N/A	I/O	PCI time-multiplexed address/data bus
M25	ad[10] ¹		N/A	I/O	PCI time-multiplexed address/data bus
M26	ad[11] ¹		N/A	I/O	PCI time-multiplexed address/data bus
N24	ad[12] ¹		N/A	I/O	PCI time-multiplexed address/data bus
N25	ad[13] ¹		N/A	I/O	PCI time-multiplexed address/data bus
N26	ad[14] ¹		N/A	I/O	PCI time-multiplexed address/data bus
P26	ad[15] ¹		N/A	I/O	PCI time-multiplexed address/data bus

Table 10: PCI interface pinout