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

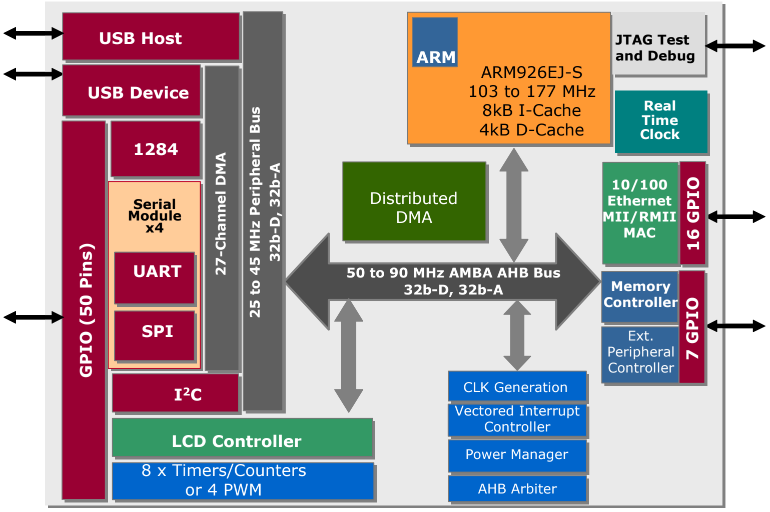
Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

# *NS9360 Datasheet*

The Digi NS9360 is a single chip 0.13µm CMOS network-attached processor. The CPU is the ARM926EJ-S core with MMU, DSP extensions, Jazelle Java accelerator, and 8 kB of instruction cache and 4 kB of data cache in a Harvard architecture. The NS9360 runs up to 177 MHz, with a 88 MHz system and memory bus and 44 MHz peripheral bus. The NS9360 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, the NS9360 is the most capable highly-integrated 32-bit network-attached processor available. The NS9360 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 NS9360 is a member of the award-winning NET+ARM family of system-on-chip (SOC) solutions for embedded systems.

NS9360
272 - pin BGA, lead-free, RoHS compliant



The NS9360 offers a connection to an external bus expansion module as well as a glueless connection to SDRAM, PC100 DIMM, Flash, EEPROM, and SRAM memories. It includes a versatile embedded LCD controller supporting up to 64K color TFT or 3375 color STN LCD display. The NS9360 features a USB port for applications requiring WLAN, external storage, or external sensors, imagers, or scanners. Four multi-function serial ports, an $I^2C$ port, and 1284 parallel port provide a standard glueless interface to a variety of external peripherals. The NS9360 features up to 73 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 the NS9360 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

# NS9360 Features

### 32-bit ARM926EJ-S RISC processor

- 103 to 177 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
- 1-32 wait states per chip select

  A shared Static Extended Wait register allows transfers to have up to 16368 wait states that can be externally terminated.
- 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/RMII 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

### Flexible LCD controller

- Supports most commercially available displays:
  - 18-bit active Matrix color TFT displays
  - Single and dual panel color STN displays
  - Single and dual-panel monochrome STN displays
- Formats image data and generates timing control signals
- Internal programmable palette LUT and grayscaler support different color techniques
- Programmable panel-clock frequency

### USB ports

- USB v.2.0 full speed (12 Mbps) and low speed (1.5 Mbps)
- Independent OHCI Host and Device ports
- Internal USB PHY
- External USB PHY interface
- USB device supports one bidirectional control endpoint and 10 unidirectional endpoints
- All endpoints supported by a dedicated DMA channel
- 32 byte FIFO per endpoint

### 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 11.25 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

### I$^2$C port

- I$^2$C 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 I$^2$C 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 12-channel DMA engine supports USB device
  - 2 DMA channels support control endpoint
  - 10 DMA channels support 10 endpoints
- One 12-channel DMA engine supports:
  - 4 serial modules (8 DMA channels)
  - 1284 parallel port (4 DMA channels)
- All DMA channels support fly-by mode

**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

- 8 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

- 73 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

- 177 MHz: 0 – 70° C
- 155 MHz: -40 – +85° C
- 103 MHz: 0 – 70° C

## System-level interfaces

Figure 1 shows the NS9360 system-level hardware interfaces.

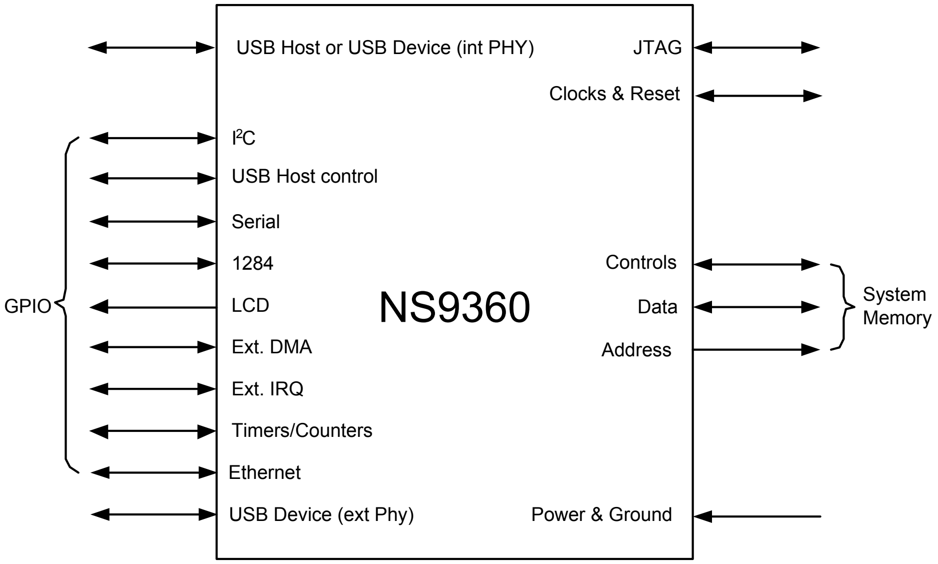


*Figure 1: System-level hardware interfaces*

### *NS9360 interfaces*

- Ethernet MII/RMII 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
- USB Host or Device interface using internal USB PHY
- I$^2$C interface
- 73 GPIO pins muxed with:
  - Four 8-pin-each serial ports, each programmable to UART or SPI
  - 1284 port
  - LCD controller interface
  - 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
- Ethernet interface — USB Device interface to external USB PHY
- 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 configuration 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 shows 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_out | **Chip select 1 byte_lane_enable_n/write_enable_n configuration bootstrap select**<br>0 write_enable_n for byte wide devices (default)<br>1 byte_lane_enable_n (2.4K pulldown added) |
| gpio[24]<br>gpio[20] | **Chip select 1 data width bootstrap select**<br>00 16 bits<br>01 8 bits<br>11 32 bits |
| gpio[49] | Chip select polarity<br>0 Active high<br>1 Active low |
| gpio[44] | Endian mode<br>0 Big endian<br>1 Little endian |
| reset_done | Bootup mode<br>0 Boot from SDRAM using serial SPI EEPROM<br>1 Boot from flash/ROM |
| gpio[19] | Reserved. This pin must not be pulled to logic 0 until reset_done is a logic 1. |
| gpio[17], gpio[12],<br>gpio[10], gpio [8], gpio[4] | PLL ND[4:0] (PLL multiplier, ND+1)<br>(See Table 2: PLL ND[4:0] multiplier values.) |
| gpio[2], gpio[0] | PLL FS[1:0] (PLL frequency select)<br>**GPIO** **FS** **Divide by**<br>10 00 1<br>11 01 2<br>00 10 4<br>01 11 8 |

***Table 1: Configuration pins— Bootstrap initialization***

| Register configuration:<br>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 |
| 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 |

*Table 2: PLL ND[4:0] multiplier values*

| Register configuration:<br>gpio 17, 12, 10, 8, 4 | Multiplier |
|---|---|
| 0 0 1 0 1 | 1 |

*Table 2: PLL ND[4:0] multiplier values*

These are sample frequency settings for each speed grade:

- 176.9472 MHz: pulldown gpio[12], gpio[10], gpio[4]
- 154.8288 MHz: pulldown gpio[12], gpio[10], gpio[8]
- 103.2192 MHz: pulldown gpio[17], gpio[10], gpio[8], gpio[4]

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 NS9360 system:

- 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_DONEn, indicates 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 NS9360 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.

# Reset

Master reset using an external reset pin resets the NS9360. 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.

## SPI 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.

## 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.

You can use one of these software resets to reset NS9360. Select the reset by setting the appropriate bit in the appropriate register:

- Watchdog timer can issue reset upon Watchdog timer expiration.
- Software reset can reset individual internal modules or all modules except memory and CPU.
- The system is reset whenever software sets the PLL SW change bit, in the PLL Configuration register, to 1.

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

| Speed grade | CPU clock cycles | Duration |
|---|---|---|
| 177 MHz | 128 | 723 ns |
| 155 MHz | 128 | 826 ns |
| 103 MHz | 128 | 1243 ns |

*Table 3: Software reset duration*

The minimum reset pulse width is 10 crystal clocks.

## System Clock

The system clock is provided to NS9360 by either a crystal or an external oscillator; this table shows sample clock frequency settings for each chip speed grade.

| Speed | cpu_clk | ahb_clk (main bus) | bbus_clk |
|---|---|---|---|
| 177 MHz | 176.9472 | 88.4736 | 44.2368 |
| 155 MHz | 154.8288 | 77.4144 | 38.7072 |
| 103 MHz | 103.2192 | 51.6096 | 24.8048 |

*Table 4: Sample clock frequency settings with 29.4912 MHz crystal*

Pulldowns are required as follows:

- To produce 176.9472 MHz, pull down gpio[12], gpio[10], gpio[4].
- To produce 154.8288 MHz, pull down gpio[12], gpio[10], gpio[8].
- To produce 103.2192 MHz, pull down gpio[17], gpio[10], gpio[8], gpio[4].

### *Using an oscillator*

If an oscillator is used, it must be connected to the x1_sys_osc input (C8 pin) on the NS9360. If a crystal is used, it must be connected with a circuit such as the one shown in Figure 2, "NS9360 system clock".

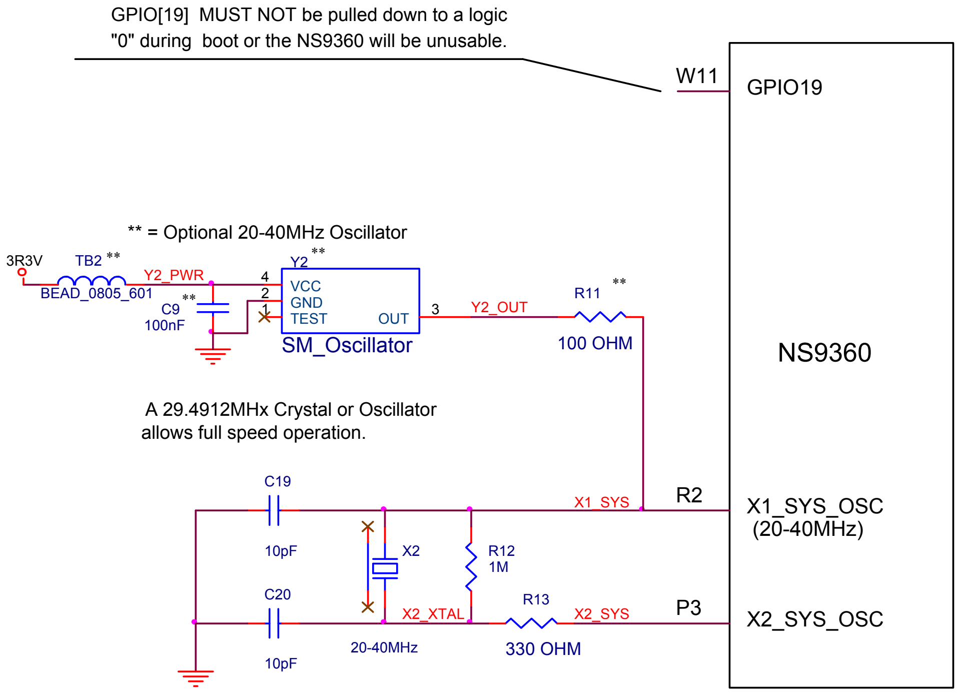


*Figure 2: NS9360 system clock*

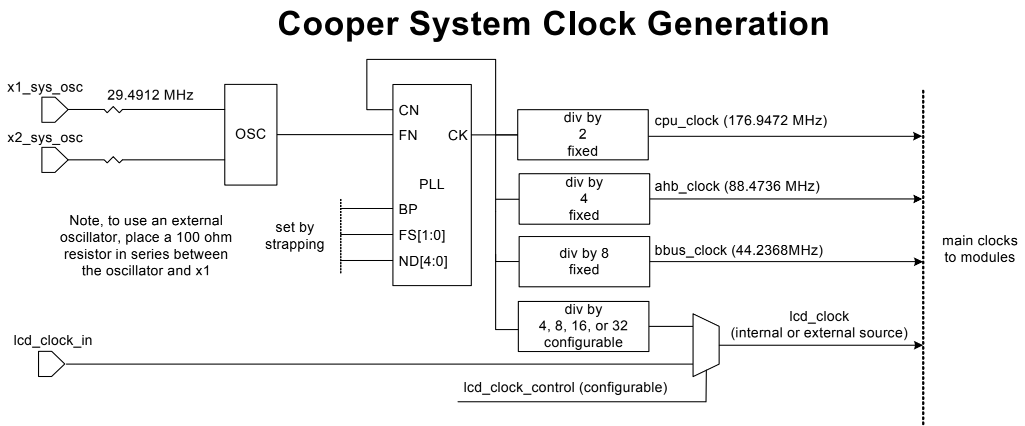
The PLL parameters are initialized on powerup reset, and can be changed by software. For a 177 MHz grade, the CPU may change from 177 MHz to 103 MHz, the AHB system bus may change from 88 MHz to 51 MHz, and the peripheral BBus may change from 44 MHz to 26 MHz. If changed by software, the system resets automatically after the PLL stabilizes (approximately 4 ms).

The system clock provides clocks for CPU, AHB system bus, peripheral BBus, 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 V4 for receive clock and tx_clk on pin V2 for transmit clock. For a RMII, there is only one clock, and it connects to the rx_clk on pin V4. In this case, the transmit clock, tx_clk, should be tied low.

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

Figure 3 shows how the PLL clock is used to provide the NS9360 system clocks by an external 48 MHz oscillator.


Cooper System Clock Generation
x1_sys_osc
29.4912 MHz
x2_sys_osc
OSC
CN
FN
CK
PLL
BP
FS[1:0]
ND[4:0]
set by strapping
Note, to use an external oscillator, place a 100 ohm resistor in series between the oscillator and x1
div by 2 fixed
div by 4 fixed
div by 8 fixed
div by 4, 8, 16, or 32 configurable
cpu_clock (176.9472 MHz)
ahb_clock (88.4736 MHz)
bbus_clock (44.2368MHz)
lcd_clock (internal or external source)
main clocks to modules
lcd_clock_in
lcd_clock_control (configurable)

**Sample Clock Frequency Settings With 29.4912MHz Crystal (FS= 01, div by 2)**

| ND+1 | $f_{VCO}$ | cpu_clk | hclk | bbus_clk | lcd_clk |
|---|---|---|---|---|---|
| 24 | 353.8944 | 176.9472 | 88.4736 | 44.2368 | 88.7872 - 11.0592 |
| 23 | 339.1488 | 169.5744 | 84.7872 | 42.3936 | 84.7872 - 10.5984 |
| 22 | 324.4032 | 162.2016 | 81.1008 | 40.5504 | 81.1008 - 10.1376 |
| 21 | 309.6576 | 154.8288 | 77.4144 | 38.7072 | 77.4144 - 9.6768 |
| 20 | 294.9120 | 147.4560 | 73.7280 | 36.8640 | 73.7280 - 9.2160 |
| 19 | 280.1644 | 140.0832 | 70.0416 | 35.0208 | 70.0416 - 8.7552 |
| 18 | 265.4208 | 132.7104 | 66.3552 | 33.1776 | 66.3552 - 8.2944 |
| 17 | 250.6752 | 125.3376 | 62.6688 | 31.3344 | 62.6688 - 7.8336 |
| 16 | 235.9296 | 117.9648 | 58.9824 | 29.4912 | 58.9824 - 7.3728 |
| 15 | 221.1840 | 110.5920 | 55.2960 | 27.6480 | 55.2960 - 6.9120 |
| 14 | 206.4384 | 103.2192 | 51.6096 | 24.8048 | 51.6096 - 6.4512 |

*Figure 3: NS9360 system clock generation (PLL)*

You can use this formula to calculate the system clock frequencies if a different

system oscillator frequency is used:

You can use this formula to calculate the system clock frequencies if a different system oscillator frequency is used:

$f_{vco}$ = ($f_{osc}$ x (ND + 1) / FS)

$f_{cpu\_clk}$ = $f_{vco}$ / 2

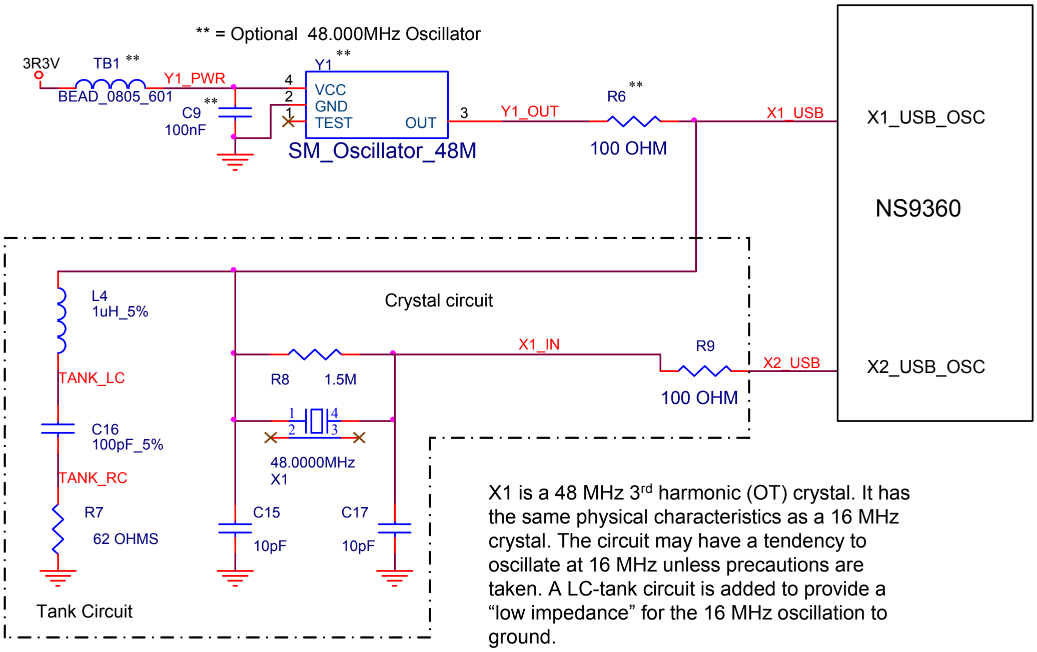
$f_{hclk}$ = $f_{vco}$ / 4

$f_{bbuys\_clk}$ = $f_{vco}$ / 8

$f_{lcd\_clk}$ = programmable, $f_{vco}$ / 4, 8, 16, or 32

## 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.



X1 is a 48 MHz 3rd harmonic (OT) crystal. It has the same physical characteristics as a 16 MHz crystal. The circuit may have a tendency to oscillate at 16 MHz unless precautions are taken. A LC-tank circuit is added to provide a “low impedance” for the 16 MHz oscillation to ground.

*Figure 4: USB clock*

## NS9360 pinout and signal descriptions

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

| Heading | Description |
|---|---|
| Pin # | Pin number assignment for a specific I/O signal. |
| Signal | 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.<br>**_n** in the signal name indicates that this signal is active *low*. |
| U/D | U or D indicates whether the pin has an internal pullup resistor or a pulldown resistor:<br>■ **U** — Pullup (input current source)<br>■ **D** — Pulldown (input current sink)<br>If no value appears, that pin has neither an internal pullup nor pulldown resistor. |
| I/O | The type of signal: input, output, or input/output. |
| OD (mA) | The output drive of an output buffer. NS9360 uses one of three drivers:<br>■ 2 mA<br>■ 4 mA<br>■ 8 mA |

More detailed signal descriptions are provided for selected modules.

### System Memory interface

Some system memory interface signals are muxed behind gpio. These signals are noted in the *Signal name / muxed behind* column. If there is no slash and no gpio pin indicated, the signal is not muxed behind a gpio signal.

| Pin # | Signal Name / muxed behind | U/D | OD (mA) | I/O | Description |
|---|---|---|---|---|---|
| P18 | addr[0] | | 8 | O | Address bus signal |
| R20 | addr[1] | | 8 | O | Address bus signal |
| P19 | addr[2] | | 8 | O | Address bus signal |
| P20 | addr[3] | | 8 | O | Address bus signal |
| N18 | addr[4] | | 8 | O | Address bus signal |
| N19 | addr[5] | | 8 | O | Address bus signal |
| N20 | addr[6] | | 8 | O | Address bus signal |
| M18 | addr[7] | | 8 | O | Address bus signal |
| M19 | addr[8] | | 8 | O | Address bus signal |

***Table 5: System Memory interface pinout***

| Pin # | Signal Name / muxed behind | U/D | OD (mA) | I/O | Description |
|---|---|---|---|---|---|
| M20 | addr[9] | | 8 | O | Address bus signal |
| L19 | addr[10] | | 8 | O | Address bus signal |
| L18 | addr[11] | | 8 | O | Address bus signal |
| L20 | addr[12] | | 8 | O | Address bus signal |
| K20 | addr[13] | | 8 | O | Address bus signal |
| K18 | addr[14] | | 8 | O | Address bus signal |
| K19 | addr[15] | | 8 | O | Address bus signal |
| J20 | addr[16] | | 8 | O | Address bus signal |
| J19 | addr[17] | | 8 | O | Address bus signal |
| J18 | addr[18] | | 8 | O | Address bus signal |
| H20 | addr[19] | | 8 | O | Address bus signal |
| H18 | addr[20] | | 8 | O | Address bus signal |
| G20 | addr[21] | | 8 | O | Address bus signal |
| H19 | addr[22] / gpio[66] | | 8 | O | Address bus signal |
| E18 | addr[23] / gpio[67] | | 8 | O | Address bus signal |
| D19 | addr[24] / gpio[68] | | 8 | O | Address bus signal |
| C20 | addr[25] / gpio[69] | | 8 | O | Address bus signal |
| A17 | addr[26] / gpio[70] | | 8 | O | Address bus signal |
| B16 | addr[27] / gpio[71] | | 8 | O | Address bus signal |
| D19 | clk_en[0] / gpio[68] | | 8 | O | SDRAM clock enable |
| C20 | clk_en[1] / gpio[69] | | 8 | O | SDRAM clock enable |
| A17 | clk_en[2] / gpio[70] | | 8 | O | SDRAM clock enable |
| B16 | clk_en[3] / gpio[71] | | 8 | O | SDRAM clock enable |
| C15 | clk_out[0] | | 8 | O | SDRAM clock |
| A12 | clk_out[1] | | 8 | O | SDRAM reference clock. Connect to clk_in using AC termination. |
| A7 | clk_out[2] | | 8 | O | SDRAM clock |
| G1 | clk_out[3] | | 8 | O | SDRAM clock |
| A16 | data[0] | | 8 | I/O | Data bus signal |
| B15 | data[1] | | 8 | I/O | Data bus signal |
| C14 | data[2] | | 8 | I/O | Data bus signal |
| A15 | data[3] | | 8 | I/O | Data bus signal |

*Table 5: System Memory interface pinout*

| Pin # | Signal Name / muxed behind | U/D | OD (mA) | I/O | Description |
|---|---|---|---|---|---|
| B14 | data[4] | | 8 | I/O | Data bus signal |
| A14 | data[5] | | 8 | I/O | Data bus signal |
| C13 | data[6] | | 8 | I/O | Data bus signal |
| B13 | data[7] | | 8 | I/O | Data bus signal |
| A13 | data[8] | | 8 | I/O | Data bus signal |
| C12 | data[9] | | 8 | I/O | Data bus signal |
| B12 | data[10] | | 8 | I/O | Data bus signal |
| B11 | data[11] | | 8 | I/O | Data bus signal |
| C11 | data[12] | | 8 | I/O | Data bus signal |
| A11 | data[13] | | 8 | I/O | Data bus signal |
| A10 | data[14] | | 8 | I/O | Data bus signal |
| C10 | data[15] | | 8 | I/O | Data bus signal |
| B10 | data[16] | | 8 | I/O | Data bus signal |
| A9 | data[17] | | 8 | I/O | Data bus signal |
| B9 | data[18] | | 8 | I/O | Data bus signal |
| C9 | data[19] | | 8 | I/O | Data bus signal |
| A8 | data[20] | | 8 | I/O | Data bus signal |
| B8 | data[21] | | 8 | I/O | Data bus signal |
| C8 | data[22] | | 8 | I/O | Data bus signal |
| B7 | data[23] | | 8 | I/O | Data bus signal |
| A6 | data[24] | | 8 | I/O | Data bus signal |
| C7 | data[25] | | 8 | I/O | Data bus signal |
| B6 | data[26] | | 8 | I/O | Data bus signal |
| A5 | data[27] | | 8 | I/O | Data bus signal |
| C6 | data[28] | | 8 | I/O | Data bus signal |
| B5 | data[29] | | 8 | I/O | Data bus signal |
| A4 | data[30] | | 8 | I/O | Data bus signal |
| C5 | data[31] | | 8 | I/O | Data bus signal |
| F2 | data_mask[0] | | 8 | O | SDRAM data mask signal |
| G3 | data_mask[1] | | 8 | O | SDRAM data mask signal |
| F1 | data_mask[2] | | 8 | O | SDRAM data mask signal |
| G2 | data_mask[3] | | 8 | O | SDRAM data mask signal |

*Table 5: System Memory interface pinout*

| Pin # | Signal Name / muxed behind | U/D | OD (mA) | I/O | Description |
|---|---|---|---|---|---|
| J2 | clk_in | | | I | SDRAM feedback clock. Connect to clk_out[1]. |
| D1 | byte_lane_sel_n[0] | | 8 | O | Static memory byte_lane_enable[0] or write_enable_n[0] for byte-wide device signals |
| E2 | byte_lane_sel_n[1] | | 8 | O | Static memory byte_lane_enable[1] or write_enable_n[1] for byte-wide device signals |
| F3 | byte_lane_sel_n[2] | | 8 | O | Static memory byte_lane_enable[2] or write_enable_n[2] for byte-wide device signals |
| E1 | byte_lane_sel_n[3] | | 8 | O | Static memory byte_lane_enable[3] or write_enable_n[3] for byte-wide device signals |
| H1 | cas_n | | 8 | O | SDRAM column address strobe |
| B4 | dy_cs_n[0] | | 8 | O | SDRAM chip select signal |
| A3 | dy_cs_n[1] | | 8 | O | SDRAM chip select signal |
| D5 | dy_cs_n[2] | | 8 | O | SDRAM chip select signal |
| C4 | dy_cs_n[3] | | 8 | O | SDRAM chip select signal |
| H2 | st_oe_n | | 8 | O | Static memory output enable |
| J3 | ras_n | | 8 | O | SDRAM row address strobe |
| B3 | st_cs_n[0] | | 8 | O | Static memory chip select signal |
| C1 | st_cs_n[1] | | 8 | O | Static memory chip select signal |
| D2 | st_cs_n[2] | | 8 | O | Static memory chip select signal |
| E3 | st_cs_n[3] | | 8 | O | Static memory chip select signal |
| H3 | we_n | | 8 | O | SDRAM write enable. Used for static and SDRAM devices. |
| J1 | ta_strb / gpio[72] | | | 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.

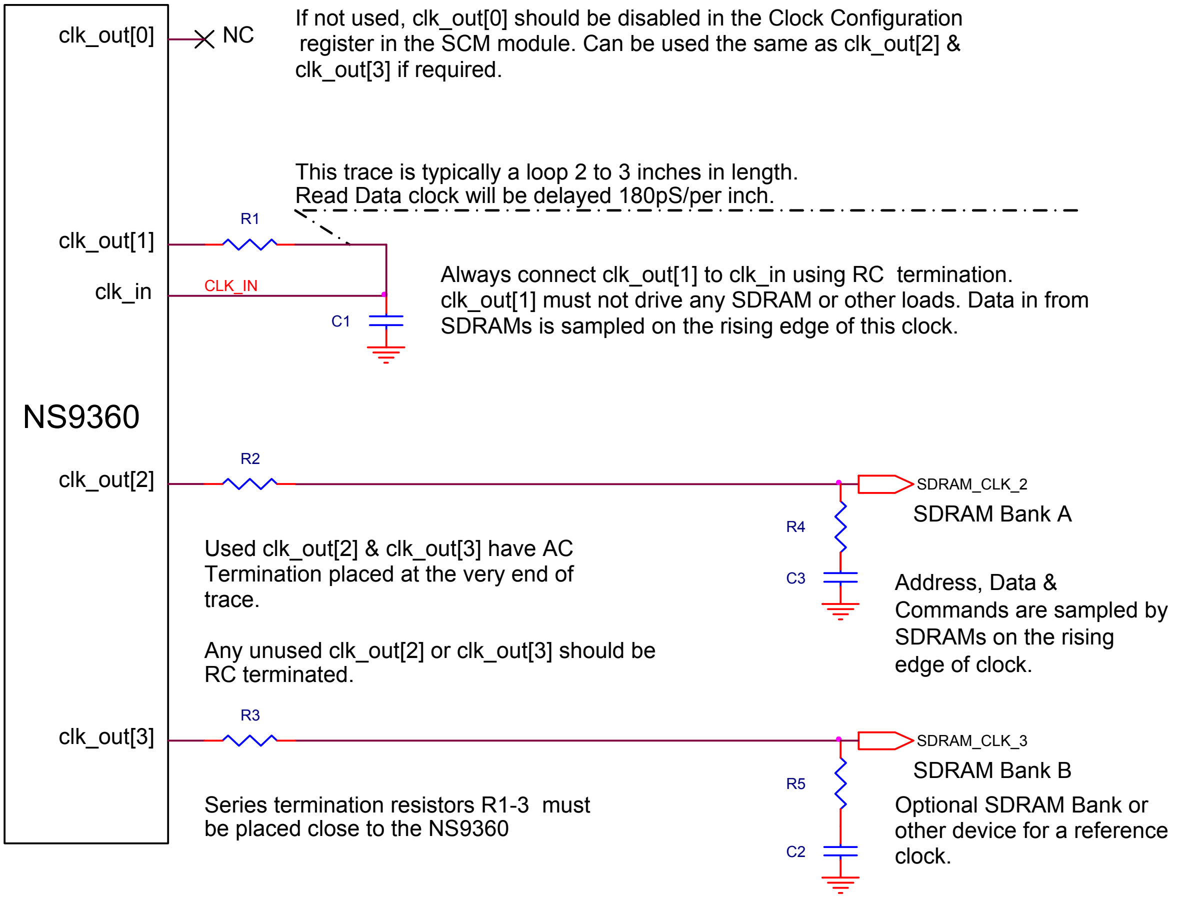
| Name | I/O | Description |
|---|---|---|
| addr[27:0] | O | Address output. Used for both static and SDRAM devices. SDRAM memories use bits [14:0]; static memories use bits [27:0].<br>Note: Address bits [27:22] are muxed behind gpio[71:66]. |

*Table 6: System Memory interface signal descriptions*

| Name | I/O | Description |
|---|---|---|
| clk_en[3:0] | O | SDRAM clock enable. Used for SDRAM devices. These signals are muxed behind gpio[71:68].<br>Note: The clk_en signals are associated with the dy_cs_n signals.<br>If clock enables are used, a pullup resistor is required to prevent floating during startup, and to avoid SDRAM lockup during manual or brown out conditions.<br>If not used, connect the clock enables in the SDRAM devices directly to 3.3v or a pullup resistor. |
| clk_out[3:0] | O | SDRAM clocks. Used for SDRAM devices. SDRAM clk_out[1] is connected to clk_in. |
| 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 | I | Feedback clock. Always connects to clk_out[1]. |
| 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 | *Slow peripheral transfer acknowledge* can be used to terminate static memory cycles sooner than the number of wait states programmed in the chip select setup register.<br>This signal is muxed being gpio[72]. |

***Table 6: System Memory interface signal descriptions***

Figure 5, "SDRAM clock termination," on page 19, shows an example of NS9360 SDRAM clock termination. clk_out[1] is shown, but you can use any clk_out signal (0, 1, 2, or 3).



If not used, clk_out[0] should be disabled in the Clock Configuration register in the SCM module. Can be used the same as clk_out[2] & clk_out[3] if required.

This trace is typically a loop 2 to 3 inches in length. Read Data clock will be delayed 180pS/per inch.

Always connect clk_out[1] to clk_in using RC termination. clk_out[1] must not drive any SDRAM or other loads. Data in from SDRAMs is sampled on the rising edge of this clock.

Used clk_out[2] & clk_out[3] have AC Termination placed at the very end of trace.

Any unused clk_out[2] or clk_out[3] should be RC terminated.

Address, Data & Commands are sampled by SDRAMs on the rising edge of clock.

Optional SDRAM Bank or other device for a reference clock.

Series termination resistors R1-3 must be placed close to the NS9360

*Figure 5: SDRAM clock termination*

## Ethernet interface

**Notes:**

- Most Ethernet MII signals are muxed behind gpio. These are noted in the *Signal name: MII / muxed behind* column. If there is no slash and no gpio pin indicated, the Ethernet signal is not muxed behind a gpio signal.
- N/C indicates *No Connect* or *ground*, as indicated in the description for the pin.

| Pin # | Signal name | | U/D | OD (mA) | I/O | Description | |
|---|---|---|---|---|---|---|---|
| | MII / muxed behind | RMII | | | | MII | RMII |
| P2 | col / gpio[63] | N/C | | | I | Collision | Pull low external to NS9360 |
| R1 | crs / gpio[64] | crs_dv | | | I | Carrier sense | Carrier sense/data valid |

*Table 7: Ethernet interface pinout*

| Pin # | Signal name | | U/D | OD (mA) | I/O | Description | |
|---|---|---|---|---|---|---|---|
| | MII / muxed behind | RMII | | | | MII | RMII |
| P1 | enet_phy_int_n / gpio[65] | enet_phy_int_n | U | | I | Ethernet PHY interrupt | Ethernet PHY interrupt |
| L2 | mdc | mdc | | 4 | O | MII management interface clock | MII management interface clock |
| K2 | mdio / gpio[50] | mdio | | 2 | I/O | MII management data | MII management data |
| V4 | rx_clk | ref_clk | | | I | Receive clock | Reference clock |
| U3 | rx_dv / gpio[51] | N/C | | | I | Receive data valid | Pull low external to NS9360 |
| V1 | rx_er / gpio[52] | rx_er | | | I | Receive error | Optional signal; pull low external to NS9360 if not used |
| N3 | rxd[0] / gpio[53] | rxd[0] | | | I | Receive data bit 0 | Receive data bit 0 |
| N2 | rxd[1] / gpio[54] | rxd[1] | | | I | Receive data bit 1 | Receive data bit 1 |
| N1 | rxd[2] / gpio[55] | N/C | | | I | Receive data bit 2 | Pull low external to NS9360 |
| M3 | rxd[3] / gpio[56] | N/C | | | I | Receive data bit 3 | Pull low external to NS9360 |
| V2 | tx_clk | N/C | | | I | Transmit clock | Pull low external to NS9360 |
| M2 | tx_en / gpio[57] | tx_en | | 2 | O | Transmit enable | Transmit enable |
| M1 | tx_er / gpio[58] | N/C | | 2 | O | Transmit error | N/A |
| L3 | txd[0] / gpio[59] | txd[0] | | 2 | O | Transmit data bit 0 | Transmit data bit 0 |
| L1 | txd[1] / gpio[60] | txd[1] | | 2 | O | Transmit data bit 1 | Transmit data bit 1 |
| K1 | txd[2] / gpio[61] | N/C | | 2 | O | Transmit data bit 2 | N/A |
| K3 | txd[3] / gpio[62] | 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 |
|---|---|---|---|---|---|
| R2 | x1_sys_osc | | | I | System clock crystal oscillator circuit input |
| P3 | x2_sys_osc | | | O | System clock crystal oscillator circuit output |
| T1 | sys_osc_vdd | | | | System oscillator 3.3V power |
| F18 | x1_usb_osc | | | I | USB clock crystal oscillator circuit input. (Connect to GND if USB is not used.) |

***Table 8: Clock generation/system pins pinout***

| Pin # | Signal name | U/D | OD (mA) | I/O | Description |
|---|---|---|---|---|---|
| E20 | x2_usb_osc |  |  | O | USB clock crystal oscillator circuit output |
| E19 | usb_osc_vdd |  |  |  | USB oscillator 3.3V power |
| W4 | reset_done | U | 2 | I/O | CPU is enabled once the boot program is loaded. Reset_done is set to 1. |
| U5 | reset_n | U |  | I | System reset input signal |
| W3 | sreset_n | U |  | I | System reset. sreset_n is the same as reset but does *not* reset the system PLL. |
| V5 | bist_en_n |  |  | I | Enable internal BIST operation |
| U6 | pll_test_n |  |  | I | Enable PLL testing |
| Y3 | scan_en_n |  |  | I | Enable internal scan testing |
| R3 | sys_pll_dvdd |  |  |  | System clock PLL 1.5V digital power |
| T2 | sys_pll_dvss |  |  |  | System clock PLL digital ground |
| U2 | sys_pll_avdd |  |  |  | System clock PLL 3.3V analog power |
| U1 | sys_pll_avss |  |  |  | System clock PLL analog ground |
| T3 | pll_lpf |  |  | O | PLL diagnostic output |
| V10 | lcdclk / gpio[15] | U |  | I | External LCD clock input (muxed behind gpio[15]) |

***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<br>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***

## GPIO MUX

- The BBus utility contains the control pins for each GPIO MUX bit. Each pin can be selected individually; that is, you can select any option (00, 01, 02, 03) for any pin, by setting the appropriate bit in the appropriate register.
- Some signals are muxed to two different GPIO pins, to maximize the number of possible applications. These duplicate signals are marked as such in the Descriptions column in the table. Selecting the primary GPIO pin and the duplicate GPIO pin for the same function is not