Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

Details	
Product Status	Not For New Designs
Applications	Network Processor
Core Processor	ARM9®
Program Memory Type	External Program Memory
Controller Series	-
RAM Size	External
Interface	EBI/EMI, Ethernet, DMA, I²C, IEEE 1284, LCD, PCI/CardBus, SPI, UART, USB
Number of I/O	50
Voltage - Supply	1.4V ~ 3.6V
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	388-BBGA
Supplier Device Package	388-BGA (35x35)
Purchase URL	https://www.e-xfl.com/product-detail/digi-international/ns9750b-a1-i162

features a PCI/CardBus port as well as a USB port for applications that require 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 NS9750B-A1 also 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 the NS9750B-A1 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 a 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

NS9750B-A1 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
- 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

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 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)
- 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

I^2C port

- I^2C 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^2C 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
- Two dedicated DMA channels

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

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.

This table describes 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. Figure 10, "NS9750B-A1 BGA layout," on page 83 shows the bootstrap pins.

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 byte_lane_enable_n (2.4K pulldown added) 1 write_enable_n for byte-wide devices (default)
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; this selects 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]	RESERVED. This pin must not be pulled to logic 0 until reset_done is a logic 1.
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)** gpio[2], [0] — **FS** — **Divide by** 10 — 00 — 1 11 — 01 — 2 00 — 10 — 4 01 — 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
0 1 0 0 0	14
0 1 0 0 1	13
0 1 1 1 0	12

Table 2: PLL ND[4:0]

System boot

There are two ways to boot the NS9750B-A1 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 NS9750B-A1 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, then enables the CPU.

Reset

Master reset using an external reset pin resets NS9750B-A1. 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.

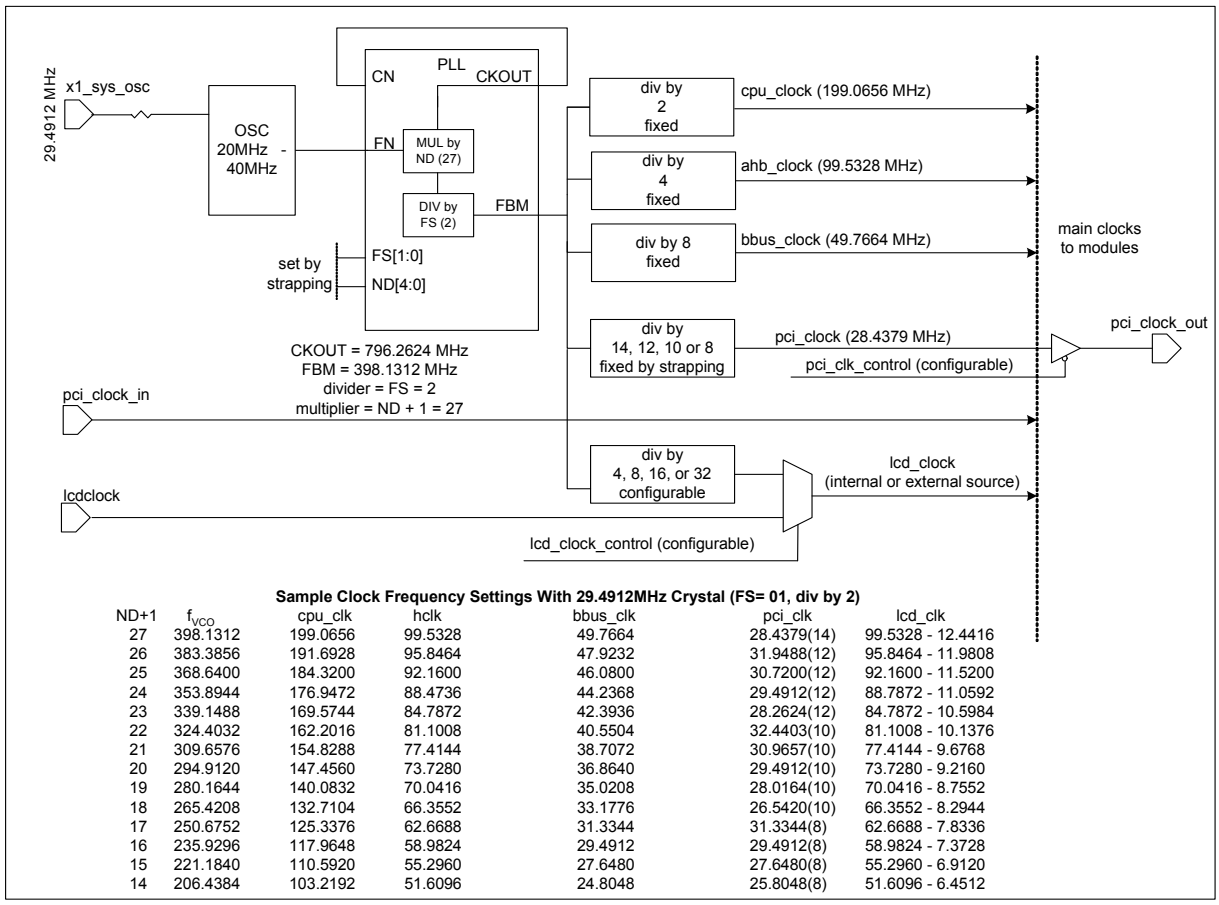
RESET_DONE as an output

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.

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.



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

ND+1	f_{VCO}	cpu_clk	hclk	bbus_clk	pci_clk	lcd_clk
27	398.1312	199.0656	99.5328	49.7664	28.4379(14)	99.5328 - 12.4416
26	383.3856	191.6928	95.8464	47.9232	31.9488(12)	95.8464 - 11.9808
25	368.6400	184.3200	92.1600	46.0800	30.7200(12)	92.1600 - 11.5200
24	353.8944	176.9472	88.4736	44.2368	29.4912(12)	88.7872 - 11.0592
23	339.1488	169.5744	84.7872	42.3936	28.2624(12)	84.7872 - 10.5984
22	324.4032	162.2016	81.1008	40.5504	32.4403(10)	81.1008 - 10.1376
21	309.6576	154.8288	77.4144	38.7072	30.9657(10)	77.4144 - 9.6768
20	294.9120	147.4560	73.7280	36.8640	29.4912(10)	73.7280 - 9.2160
19	280.1644	140.0832	70.0416	35.0208	28.0164(10)	70.0416 - 8.7552
18	265.4208	132.7104	66.3552	33.1776	26.5420(10)	66.3552 - 8.2944
17	250.6752	125.3376	62.6688	31.3344	31.3344(8)	62.6688 - 7.8336
16	235.9296	117.9648	58.9824	29.4912	29.4912(8)	58.9824 - 7.3728
15	221.1840	110.5920	55.2960	27.6480	27.6480(8)	55.2960 - 6.9120
14	206.4384	103.2192	51.6096	24.8048	25.8048(8)	51.6096 - 6.4512

Figure 3: NS9750B-A1 system clock generation (PLL)

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

fvco = (fosc x (ND + 1) / FS)
fcpu_clk = fvco / 2
fhclk = fvco / 4
fbbus_clk = fvco / 8
fpci_clk = fvco / 14, 12, 10 or 8
flcd_clk = programmable, fvco / 4, 8, 16, or 32

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 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

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 [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.
data[31:0]	I/O	Data to/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[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	*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.

Table 6: System Memory interface signal descriptions

Pin #	Signal Name	U/D	OD (mA)	I/O	Description
V26	ad[17][1]		N/A	I/O	PCI time-multiplexed address/data bus
V25	ad[18][1]		N/A	I/O	PCI time-multiplexed address/data bus
W26	ad[19][1]		N/A	I/O	PCI time-multiplexed address/data bus
V24	ad[20][1]		N/A	I/O	PCI time-multiplexed address/data bus
W25	ad[21][1]		N/A	I/O	PCI time-multiplexed address/data bus
Y26	ad[22][1]		N/A	I/O	PCI time-multiplexed address/data bus
W24	ad[23][1]		N/A	I/O	PCI time-multiplexed address/data bus
Y24	ad[24][1]		N/A	I/O	PCI time-multiplexed address/data bus
AA25	ad[25][1]		N/A	I/O	PCI time-multiplexed address/data bus
AB26	ad[26][1]		N/A	I/O	PCI time-multiplexed address/data bus
AA24	ad[27][1]		N/A	I/O	PCI time-multiplexed address/data bus
AB25	ad[28][1]		N/A	I/O	PCI time-multiplexed address/data bus
AC26	ad[29][1]		N/A	I/O	PCI time-multiplexed address/data bus
AD26	ad[30][1]		N/A	I/O	PCI time-multiplexed address/data bus
AC25	ad[31][1]		N/A	I/O	PCI time-multiplexed address/data bus
L25	cbe_n[0][1]		N/A	I/O	Command/byte enable
P25	cbe_n[1][1]		N/A	I/O	Command/byte enable
U25	cbe_n[2][1]		N/A	I/O	Command/byte enable
AA26	cbe_n[3][1]		N/A	I/O	Command/byte enable
T26	devsel_n[2]		N/A	I/O	Device select
U26	frame_n[2]		N/A	I/O	Cycle frame
Y25	idsel[3, 4]		N/A	I	Initialization device select: ■ For PCI host applications, connect to AD11. ■ For PCI device applications, connection is determined by the PCI device number assigned to the NS9750B-A1. ■ For CardBus applications, connect to external pullup resistor. ■ Do not allow input to float in any application.
T24	irdy_n[2]		N/A	I/O	Initiator ready
P24	par[1]		N/A	I/O	Parity signal
R25	perr_n[2]		N/A	I/O	Parity error

Table 10: PCI interface pinout

I²C interface

Pin #	Signal name	U/D	OD (mA)	I/O	Description
AC15	iic_scl		4	I/O	I²C serial clock line. Add a 10K resistor to VDDA(3.3V) if not used.
AF16	iic_sda		4	I/O	I²C serial data line. Add a 10K resistor to VDDA(3.3V) if not used.

Table 17: I²C interface pinout

USB Interface

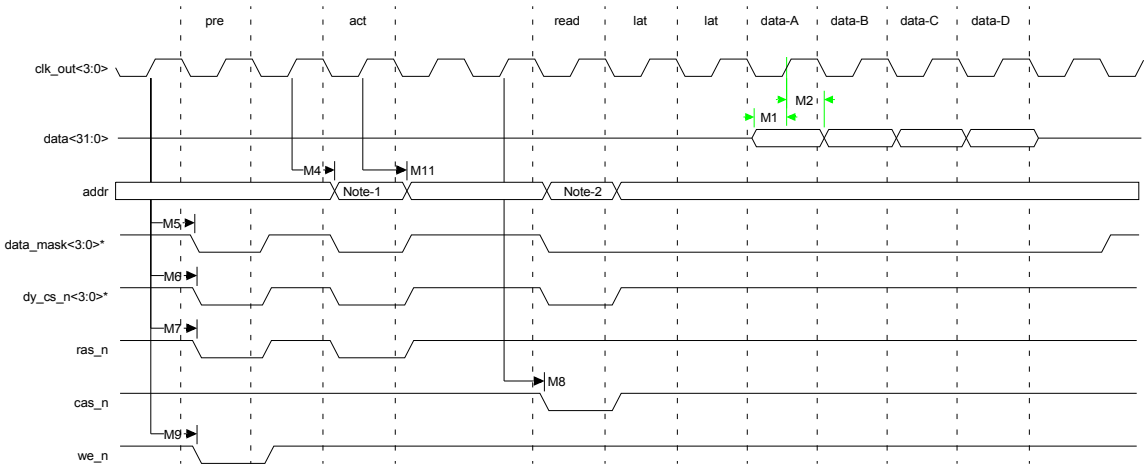
Notes:

- If not using the USB interface, these pins should be pulled down to ground through a 15K ohm resistor.
- All output drivers for USB meet the standard USB driver specification.

Pin #	Signal name	U/D	OD (mA)	I/O	Description
AB4	usb_dm			I/O	USB data -
AC3	usb_dp			I/O	USB data +

Table 18: USB interface pinout

SDRAM burst read (32-bit), CAS latency = 3

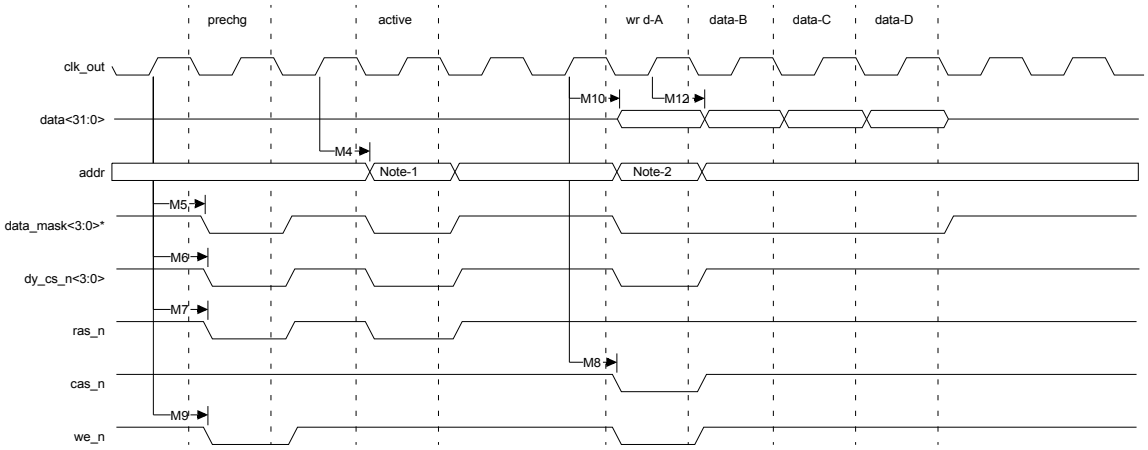


Notes:

1 This is the bank and RAS address.

2 This is the CAS address.

SDRAM burst write (32-bit)

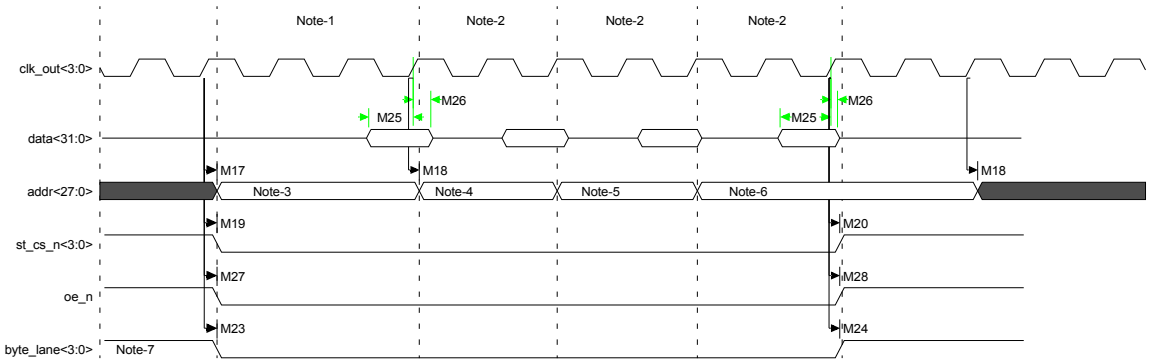


Notes:

1 This is the bank and RAS address.

2 This is the CAS address.

Static RAM asynchronous page mode read, WTPG = 1



- WTPG = 1
  WTRD = 2
- If the PB field is set to 1, all four byte_lane signals will go low for 32-bit, 16-bit, and 8-bit read cycles.
- The asynchronous page mode will read 16 bytes in a page cycle. A 32-bit bus will do four 32-bit reads, as shown (3-2-2-2). A 16-bit bus will do eight 16-bit reads (3-2-2-2-3-2-2-2) per page cycle, and an 8-bit bus will do sixteen reads (3-2-2-2-3-2-2-2-3-2-2-2-3-2-2-2) per page cycle. 3-2-2-2 is the example used here, but the WTRD and WTPG field can set them differently.

Notes:

1 The length of the first cycle in the page is determined by the WTRD field.

2 The length of the 2nd, 3rd, and 4th cycles is determined by the WTPG field.

3 This is the starting address. The least significant two bits will always be ‘00.’

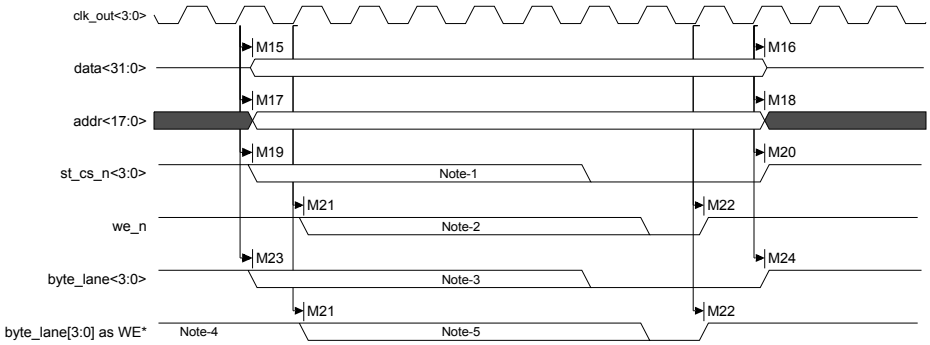
4 The least significant two bits in the second cycle will always be ‘01.’

5 The least significant two bits in the third cycle will always be ‘10.’

6 The least significant two bits in the fourth cycle will always be ‘11.’

7 If the PB field is set to 0, the byte_lane signal will always be high during a read cycle.

Static write cycle with configurable wait states



- WTWR = from 0 to 15
  WWEN = from 0 to 15
- The WTWR field determines the length on the write cycle.
- During a 32-bit transfer, all four byte_lane signals will go low.
- During a 16-bit transfer, two byte_lane signals will go low.
- During an 8-bit transfer, only one byte_lane signal will go low.

Notes:

1 Timing of the st_cs_n signal is determined with a combination of the WTWR and WWEN fields. The st_cs_n signal will always go low at least one clock before we_n goes low, and will go high one clock after we_n goes high.

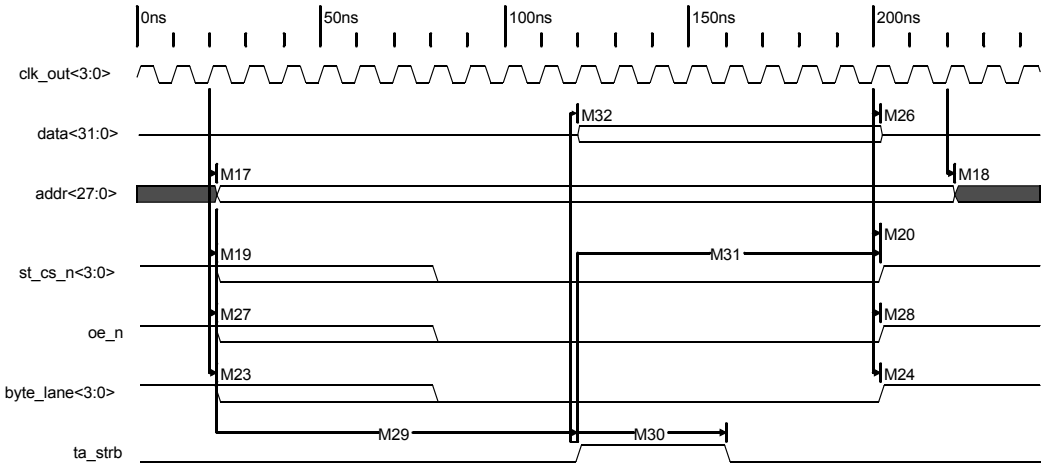
2 Timing of the we_n signal is determined with a combination of the WTWR and WWEN fields.

3 Timing of the byte_lane signals is determined with a combination of the WTWR and WWEN fields. The byte_lane signals will always go low one clock before we_n goes low, and will go one clock high after we_n goes high.

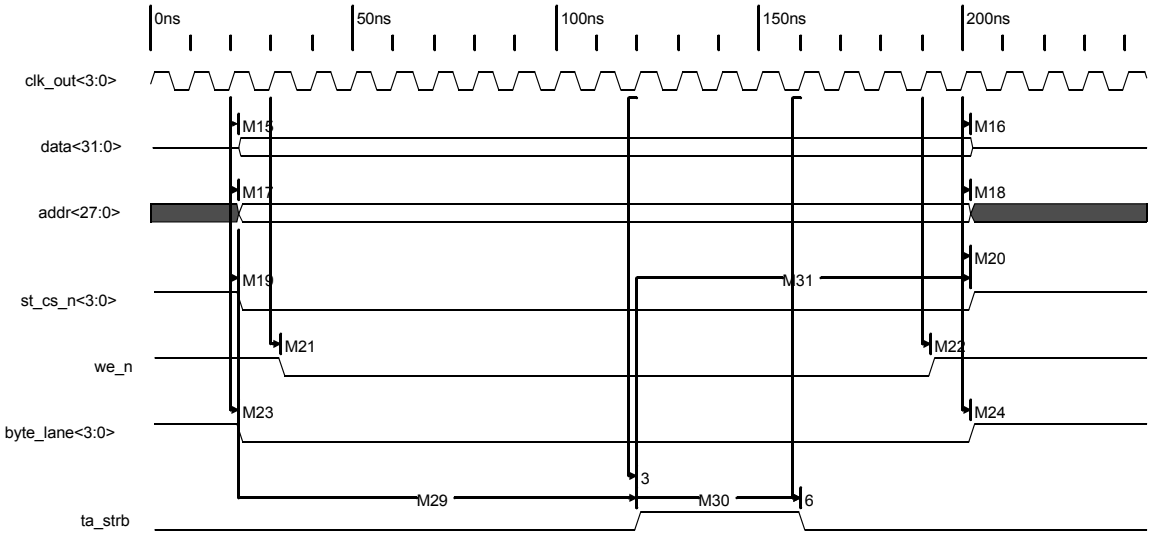
4 If the PB field is set to 0, the byte_lane signals will function as the write enable signals and the we_n signal will always be high.

5 If the PB field is set to 0, the timing for the byte_lane signals is set with the WTWR and WWEN fields.

Slow peripheral acknowledge read



Slow peripheral acknowledge write



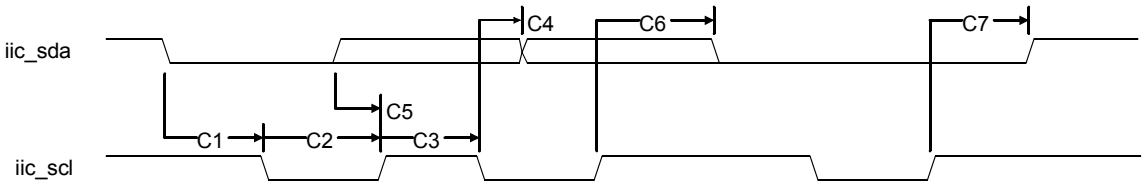
I²C timing

I²C AC characteristics are measured with 10pF, unless otherwise noted.

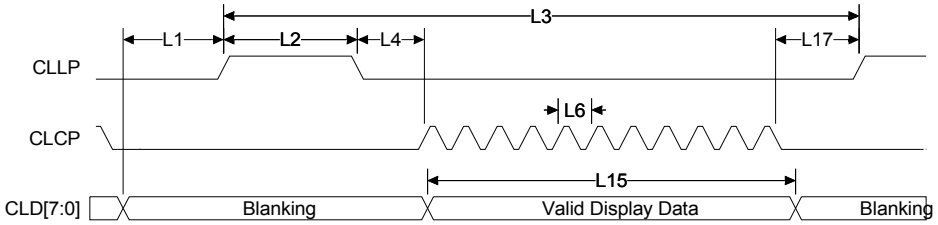
Table 32 describes the values shown in the I²C timing diagram.

		Standard mode		Fast mode		
Parm	**Description**	**Min**	**Max**	**Min**	**Max**	**Unit**
C1	iic_sda to iic_scl START hold time	4.0		0.6		µs
C2	iic_scl low period	4.7		1.3		µs
C3	iic_scl high period	4.7		1.3		µs
C4	iic_scl to iic_sda DATA hold time	0		0		µs
C5	iic_sda to iic_scl DATA setup time	250		100		ns
C6	iic_scl to iic_sda START setup time	4.7		0.6		µs
C7	iic_scl to iic_sda STOP setup time	4.0		0.6		µs

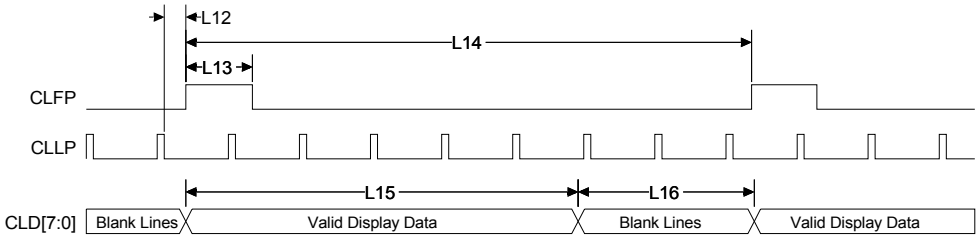
Table 32: I²C timing parameters



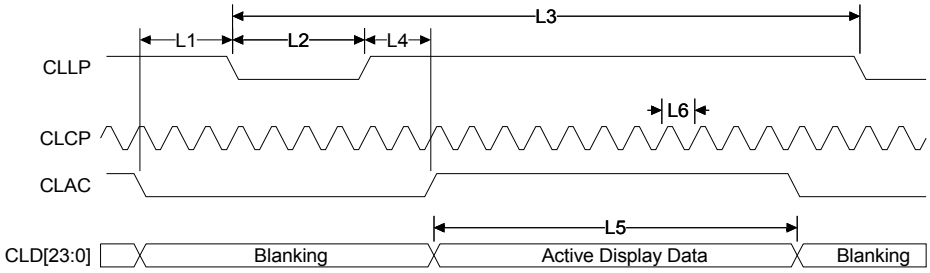
Horizontal timing for STN displays



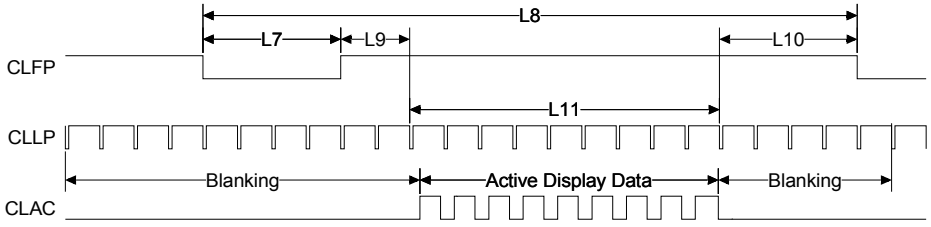
Vertical timing for STN displays



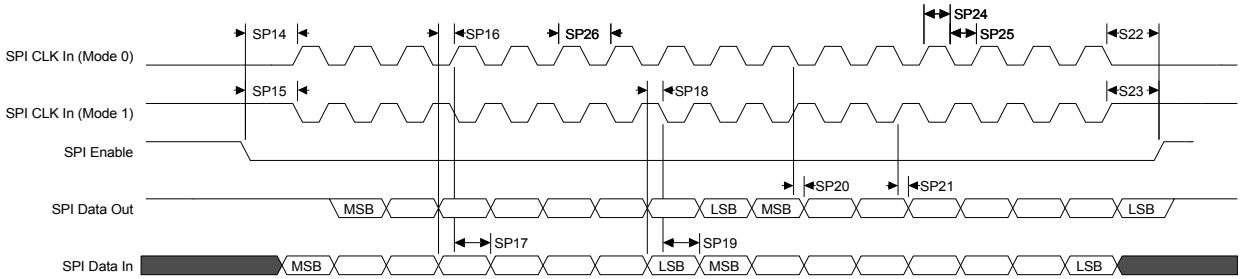
Horizontal timing for TFT displays



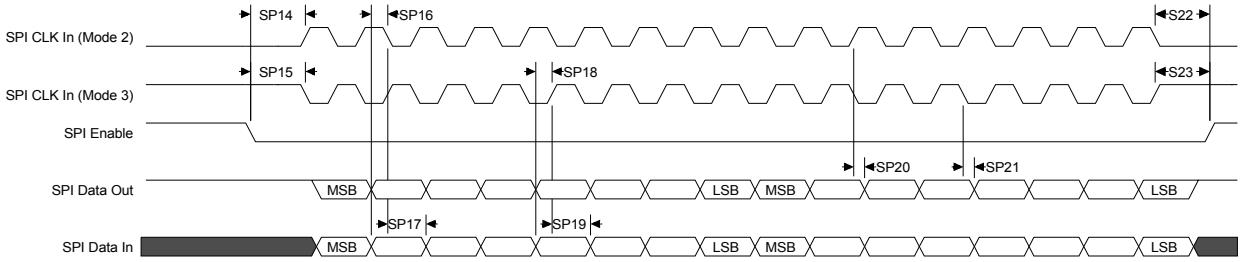
Vertical timing for TFT displays



SPI slave mode 0 and 1: 2-byte transfer (see note 7)



SPI slave mode 2 and 3: 2-byte transfer (see note 7)



IEEE 1284 timing

IEEE 1284 AC characteristics are measured with 10pF, unless otherwise noted.

Table 35 describes the values shown in the IEEE 1284 timing diagram.

Parameter	Description	Min	Max	Unit	Note
IE1	Busy-while-Strobe	0	500	ns	1
IE2	Busy high to nAck low	0		ns	
IE3	Busy high		1000	ns	2
IE4	nAck low		500	ns	3
IE5	nAck high to Busy low		500	ns	3

Table 35: IEEE 1284 timing parameters

Notes:

1 The range is 0ns up to one time unit.

2 Two time units.

3 Three time units.

IEEE 1284 timing example

The IEEE 1284 timing is determined by the BBus clock and the Granularity Count register (GCR) setting. In this example, the BBus clock is 50 MHz and the Granularity Count register is set to 25. The basic time unit is 1/50 MHz x 25, which is 500ns.