Nios II 3C120 Microprocessor with LCD Controller Data Sheet

DS-01002-1.0 © November 2008

Introduction

This data sheet describes a single instance of a Nios® II-based processor system with a built-in LCD controller targeted for an Altera® Cyclone® III 3C120F780 FPGA on the Altera Embedded Systems Development Kit, Cyclone III Edition. The Nios II 3C120 Microprocessor with LCD Controller is a complete system-on-a-programmable-chip (SOPC) solution that incorporates a rich set of system peripherals and standard interfaces for a wide range of embedded applications involving video processing and LCD touch panel control.

The key benefit of implementing a processor system in an FPGA is that you can customize your system using intellectual property (IP) cores, custom logic, and hardware acceleration to optimize to your target application. Nearly every feature in the system is available for you to configure, customize, duplicate, or remove easily. You can further enhance your system by adding additional IP to the Cyclone III 3C120 device, or you can remove IP or select options that reduce logic utilization allowing you to port to a smaller device to reduce cost.

☞ Although this data sheet describes a system targeted for the Cyclone III 3C120 FPGA device, the data sheet also shows you how to configure the processor system for another Altera FPGA device and hardware platform of your choice.

Features

The following list summarizes the main features of the Nios II 3C120 microprocessor with LCD controller.

Target Hardware Board

- Altera Embedded Systems Development Kit, Cyclone III Edition

Device

- System name: cycloneIII_3c120_niosII_video
- Family: Cyclone III
- Device: 3C120F780
- Total logic elements (LE) used: 24,500 / 119,000 (21%)
- Total pins used: 214 / 532 (40%)
- Total memory used: 713,764 / 3,981,312 (18%)

Processor

- Nios II/f processor core
- Nominal metrics: 113 DMIPS at 100 MHz, 1,400–1,800 LEs, MMU/MPU option disabled
- 32-KByte data cache, 32-KByte instruction cache

- JTAG debug module for downloading software, 300–400 LEs

Memory Interfaces

- Common flash interface (CFI) flash memory
  - 64 MBytes
- High performance DDR2 SDRAM memory (arranged in two memory banks)
  - Nominal frequency: 150 MHz, 32 bits, 32 MBytes
- SD/MMC card serial peripheral interface (SPI)
  - 100-MHz SPI interface clock frequency
  - Supports up to 1-Gbit SD card memory

Communication Interfaces

- Ethernet MAC 10/100/1000 Base T
  - Integrated in receive and transmit FIFO 1024 × 32 bits each
  - Media independent interface (MII)/gigabit media independent interface (GMII) support
  - 32-bit transmit and receive scatter gather direct memory access (SG-DMA) channels
- JTAG UART with integrated read and write FIFO
- UART for RS-232 serial communication
  - 115,200 baud rate, no parity, 8 data bits, 1 stop bit
- 2-wire interface
  - Implemented using general purpose PIOs
  - Dedicated to LCD controller interface

Video Subsystem

- Integrated LCD controller IP
  - Configured to 800 × 480 resolution
  - Interface for LCD control using 2-wire interface
    - Implemented using general purpose PIOs
- Integrated touch panel controller IP
  - Interfaces to LCD using 3-wire SPI
  - Master mode 8 bit data register, 32 KHz
- Video pipeline
  - Streaming video data path
  - Video frame buffer
  - RGB Sync generation IP
  - 128-byte dual clock FIFO

System Peripherals

- Timers/counters
  - System clock timer
    - 32-bit counter size, 10-ms time-out period
    - High resolution timer
    - 32-bit counter size, 10-µs time-out period
  - Performance counter
    - 1 simultaneous measured section
- 4 button PIOs (input only)
- 8 LED PIOs (output only)
- System ID

Description

The Nios II 3C120 microprocessor with LCD controller incorporates a Nios II processor, LCD controller, memories, a video pipeline, and more in a single Cyclone III 3C120 FPGA. You can configure nearly every aspect of the processor system to suit your application requirements. You can configure the Nios II processor as one of following cores:

- A size-optimized economy (/e) core
- A performance-optimum fast (/f) core
- An optimum size-to-performance standard (/s) core

In addition, the fast core comes with options to include a memory management unit (MMU) as well as various precise exceptions and memory protection features. The Nios II processor supports custom instructions allowing you to implement software functions in hardware to increase system performance. You can configure the JTAG debug module to support hardware breakpoints, data triggers, and instruction and data on-chip and off-chip trace.

The microprocessor supports an LCD color touch panel by integrating the LCD controller hardware, which has been implemented as part of a video pipeline. The video pipeline is fully logic based, composed of IP cores you can modify to suit any resolution or aspect ratio.

A DDR2 SDRAM memory interface provides memory to execute processor program code. A second high performance DDR2 SDRAM memory interface holds the video buffers. User-selectable SRAM memory (on-chip) is available for tightly coupled low latency memory. A CFI flash controller supports flash memory to store application code and FPGA configuration data.

The microprocessor integrates an SD/MMC controller that is SD card compliant. The system also includes a 10/100/1000 Ethernet MAC with SG-DMA channels for network access.

Figure 1 shows the functional blocks of the Nios II 3C120 Processor System.

Figure 1. Block Diagram

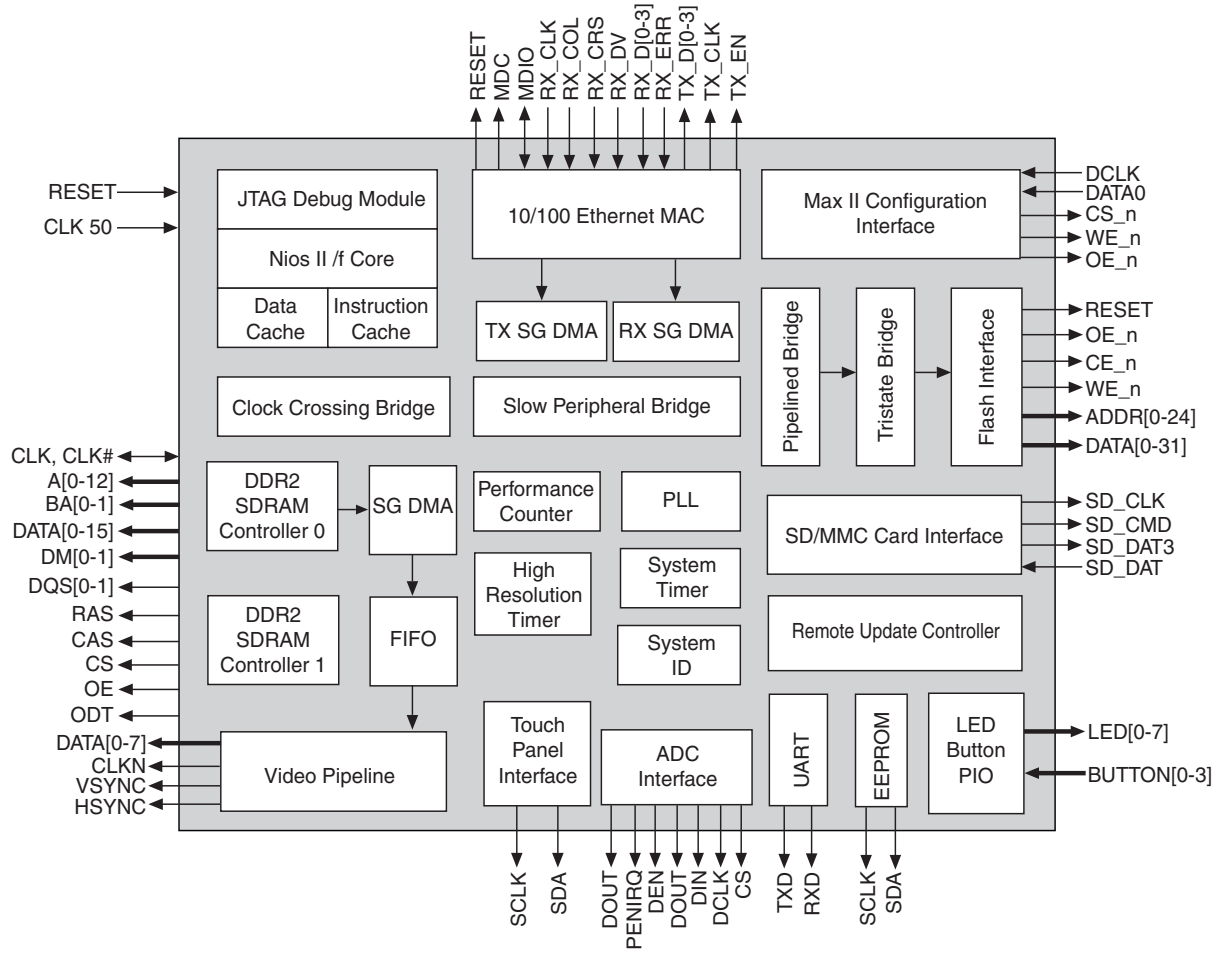


Table 1 lists the pin definitions and signal descriptions. In the Location column, letter values refer to the pin row within the FPGA and numeric values refer to the pin column. So for example, A1 denotes the pin located in row A and column 1.

For the Cyclone III 3C120 device pin list, refer to *Pin Information for the Cyclone III EP3C120 Device*.

Table 1. Signal Descriptions (Part 1 of 7)

Pin Name	Location	Signal Description	Direction	I/O Standard
System Clock and Reset Inputs				
top_reset_n	T21	FPGA reset input	Input	2.5 V
top_clkin_125	A14	125-MHz crystal oscillator Input	Input	1.8 V
top_clkin_50	AH15	50-MHz crystal oscillator Input	Input	1.8 V
Configuration Device (MAX® II - EPM2210_F256FBGA) Interface				
~ALTERA_DATA0~ / RESERVED_INPUT	N7	FPGA configuration data	Input	2.5 V

Table 1. Signal Descriptions (Part 2 of 7)

Pin Name	Location	Signal Description	Direction	I/O Standard
~ALTERA_DCLK~ / RESERVED_INPUT	P3	FPGA configuration clock	Input	2.5 V
top_cs_n_to_the_max2	D23	Chip select for MAX II	Output	1.8 V
top_we_n_to_the_max2	C15	Write enable (active low)	Output	1.8 V
top_oe_n_to_the_max2	E25	Output enable (active low)	Output	1.8 V
TOP DDR2 SDRAM Memory (MT47H32M16CC) Interface				
top_ddr2_ck_p[1]	H12	Positive differential clock Input	Bidirectional	SSTL-18 Class I
top_ddr2_ck_n[1]	G11	Negative differential clock Input	Bidirectional	SSTL-18 Class I
top_ddr2top_a[0]	J13	Address	Output	SSTL-18 Class I
top_ddr2top_a[1]	G18	Address	Output	SSTL-18 Class I
top_ddr2top_a[2]	E8	Address	Output	SSTL-18 Class I
top_ddr2top_a[3]	D24	Address	Output	SSTL-18 Class I
top_ddr2top_a[4]	D7	Address	Output	SSTL-18 Class I
top_ddr2top_a[5]	J15	Address	Output	SSTL-18 Class I
top_ddr2top_a[6]	H15	Address	Output	SSTL-18 Class I
top_ddr2top_a[7]	J16	Address	Output	SSTL-18 Class I
top_ddr2top_a[8]	H8	Address	Output	SSTL-18 Class I
top_ddr2top_a[9]	D16	Address	Output	SSTL-18 Class I
top_ddr2top_a[10]	A17	Address	Output	SSTL-18 Class I
top_ddr2top_a[11]	D8	Address	Output	SSTL-18 Class I
top_ddr2top_a[12]	D25	Address	Output	SSTL-18 Class I
top_ddr2top_ba[0]	C23	Bank address	Output	SSTL-18 Class I
top_ddr2top_ba[1]	D19	Bank address	Output	SSTL-18 Class I
top_ddr2top_dm[0]	B10	Data mask	Output	SSTL-18 Class I
top_ddr2top_dm[1]	A8	Data mask	Output	SSTL-18 Class I
top_ddr2top_dq[0]	A12	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[1]	C14	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[2]	A11	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[3]	C13	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[4]	D15	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[5]	C12	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[6]	E14	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[7]	D13	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[8]	B7	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[9]	C11	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[10]	A7	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[11]	C10	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[12]	E11	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[13]	B6	Data I/O	Bidirectional	SSTL-18 Class I

Table 1. Signal Descriptions (Part 3 of 7)

Pin Name	Location	Signal Description	Direction	I/O Standard
top_ddr2top_dq[14]	H13	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dq[15]	D10	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2top_dqs[0]	D12	Data strobe	Bidirectional	SSTL-18 Class I
top_ddr2top_dqs[1]	E12	Data strobe	Bidirectional	SSTL-18 Class I
top_ddr2top_odt	A6	On-die termination	Output	SSTL-18 Class I
top_ddr2top_rasn	F8	RAS command	Output	SSTL-18 Class I
top_ddr2top_wen	A10	Write enable command	Output	SSTL-18 Class I
top_ddr2top_casn	F14	CAS command	Output	SSTL-18 Class I
top_ddr2top_cke	E21	Clock enable	Output	SSTL-18 Class I
top_ddr2top_csn	C7	Chip select input	Output	SSTL-18 Class I
BOTTOM DDR2 SDRAM Memory (MT47H32M16CC) Interface				
top_ddr2_ck_n[0]	AF14	Negative differential clock Input	Bidirectional	SSTL-18 Class I
top_ddr2_ck_p[0]	AE14	Positive differential clock Input	Bidirectional	SSTL-18 Class I
top_ddr2bot_a[0]	AB22	Address	Output	SSTL-18 Class I
top_ddr2bot_a[1]	AG6	Address	Output	SSTL-18 Class I
top_ddr2bot_a[2]	Y13	Address	Output	SSTL-18 Class I
top_ddr2bot_a[3]	AE7	Address	Output	SSTL-18 Class I
top_ddr2bot_a[4]	AB12	Address	Output	SSTL-18 Class I
top_ddr2bot_a[5]	AC7	Address	Output	SSTL-18 Class I
top_ddr2bot_a[6]	AD12	Address	Output	SSTL-18 Class I
top_ddr2bot_a[7]	AB8	Address	Output	SSTL-18 Class I
top_ddr2bot_a[8]	AH12	Address	Output	SSTL-18 Class I
top_ddr2bot_a[9]	AB10	Address	Output	SSTL-18 Class I
top_ddr2bot_a[10]	AE4	Address	Output	SSTL-18 Class I
top_ddr2bot_a[11]	AF21	Address	Output	SSTL-18 Class I
top_ddr2bot_a[12]	Y12	Address	Output	SSTL-18 Class I
top_ddr2bot_dm[0]	AH19	Data mask	Output	SSTL-18 Class I
top_ddr2bot_dm[1]	AC15	Data mask	Output	SSTL-18 Class I
top_ddr2bot_dq[0]	AG22	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[1]	AH21	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[2]	AH22	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[3]	AG21	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[4]	AD17	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[5]	AH23	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[6]	AE19	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[7]	AF24	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[8]	AG18	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[9]	AG17	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[10]	AH18	Data I/O	Bidirectional	SSTL-18 Class I

Table 1. Signal Descriptions (Part 4 of 7)

Pin Name	Location	Signal Description	Direction	I/O Standard
top_ddr2bot_dq[11]	AH17	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[12]	AF15	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[13]	AE17	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[14]	AF16	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_dq[15]	AB16	Data I/O	Bidirectional	SSTL-18 Class I
top_ddr2bot_ba[0]	AF3	Bank address	Output	SSTL-18 Class I
top_ddr2bot_ba[1]	AF5	Bank address	Output	SSTL-18 Class I
top_ddr2bot_casn	AD21	CAS strobe (active low) command	Output	SSTL-18 Class I
top_ddr2bot_cke	AG4	Clock Enable	Output	SSTL-18 Class I
top_ddr2bot_csn	AC21	Chip select (active low)	Output	SSTL-18 Class I
top_ddr2bot_dqs[0]	AE18	Data strobe	Bidirectional	SSTL-18 Class I
top_ddr2bot_dqs[1]	AF17	Data strobe	Bidirectional	SSTL-18 Class I
top_ddr2bot_odt	AE24	On-die termination (ODT)	Output	SSTL-18 Class I
top_ddr2bot_rasn	AE21	RAS strobe command	Output	SSTL-18 Class I
top_ddr2bot_wen	AE5	Write enable	Output	SSTL-18 Class I
Flash Memory (S29GL512N) Interface				
top_flash_cen	Y16	Chip enable (active low)	Output	1.8 V
top_flash_oen	Y17	Output enable (active low)	Output	1.8 V
top_flash_resetn	AB20	Reset (active low)	Output	1.8 V
top_flash_wen	AA21	Write enable (active low)	Output	1.8 V
top_fsa[0]	AC11	Flash address	Output	1.8 V
top_fsa[1]	AH10	Flash address	Output	1.8 V
top_fsa[2]	AA13	Flash address	Output	1.8 V
top_fsa[3]	AC10	Flash address	Output	1.8 V
top_fsa[4]	Y15	Flash address	Output	1.8 V
top_fsa[5]	AF22	Flash address	Output	1.8 V
top_fsa[6]	AF26	Flash address	Output	1.8 V
top_fsa[7]	AF4	Flash address	Output	1.8 V
top_fsa[8]	AD8	Flash address	Output	1.8 V
top_fsa[9]	AG26	Flash address	Output	1.8 V
top_fsa[10]	AH6	Flash address	Output	1.8 V
top_fsa[11]	AD24	Flash address	Output	1.8 V
top_fsa[12]	AF9	Flash address	Output	1.8 V
top_fsa[13]	AA8	Flash address	Output	1.8 V
top_fsa[14]	AC22	Flash address	Output	1.8 V
top_fsa[15]	AE8	Flash address	Output	1.8 V
top_fsa[16]	AF13	Flash address	Output	1.8 V
top_fsa[17]	AB14	Flash address	Output	1.8 V
top_fsa[18]	AF23	Flash address	Output	1.8 V

Table 1. Signal Descriptions (Part 5 of 7)

Pin Name	Location	Signal Description	Direction	I/O Standard
top_fsa[19]	AG12	Flash address	Output	1.8 V
top_fsa[20]	AB18	Flash address	Output	1.8 V
top_fsa[21]	Y19	Flash address	Output	1.8 V
top_fsa[22]	AG3	Flash address	Output	1.8 V
top_fsa[23]	AE16	Flash address	Output	1.8 V
top_fsa[24]	AB7	Flash address	Output	1.8 V
top_fsd[0]	J14	Flash data	Bidirectional	1.8 V
top_fsd[1]	D6	Flash data	Bidirectional	1.8 V
top_fsd[2]	J17	Flash data	Bidirectional	1.8 V
top_fsd[3]	G7	Flash data	Bidirectional	1.8 V
top_fsd[4]	F18	Flash data	Bidirectional	1.8 V
top_fsd[5]	C6	Flash data	Bidirectional	1.8 V
top_fsd[6]	H17	Flash data	Bidirectional	1.8 V
top_fsd[7]	C18	Flash data	Bidirectional	1.8 V
top_fsd[8]	D18	Flash data	Bidirectional	1.8 V
top_fsd[9]	G16	Flash data	Bidirectional	1.8 V
top_fsd[10]	G22	Flash data	Bidirectional	1.8 V
top_fsd[11]	F12	Flash data	Bidirectional	1.8 V
top_fsd[12]	D11	Flash data	Bidirectional	1.8 V
top_fsd[13]	E24	Flash data	Bidirectional	1.8 V
top_fsd[14]	H21	Flash data	Bidirectional	1.8 V
top_fsd[15]	G9	Flash data	Bidirectional	1.8 V
top_fsd[16]	A4	Flash data	Bidirectional	1.8 V
top_fsd[17]	G13	Flash data	Bidirectional	1.8 V
top_fsd[18]	H14	Flash data	Bidirectional	1.8 V
top_fsd[19]	B8	Flash data	Bidirectional	1.8 V
top_fsd[20]	C8	Flash data	Bidirectional	1.8 V
top_fsd[21]	F7	Flash data	Bidirectional	1.8 V
top_fsd[22]	B11	Flash data	Bidirectional	1.8 V
top_fsd[23]	B22	Flash data	Bidirectional	1.8 V
top_fsd[24]	A18	Flash data	Bidirectional	1.8 V
top_fsd[25]	G8	Flash data	Bidirectional	1.8 V
top_fsd[26]	J12	Flash data	Bidirectional	1.8 V
top_fsd[27]	D9	Flash data	Bidirectional	1.8 V
top_fsd[28]	C9	Flash data	Bidirectional	1.8 V
top_fsd[29]	E7	Flash data	Bidirectional	1.8 V
top_fsd[30]	H10	Flash data	Bidirectional	1.8 V
top_fsd[31]	J10	Flash data	Bidirectional	1.8 V
LCD Touch Panel Controller Interface				

Table 1. Signal Descriptions (Part 6 of 7)

Pin Name	Location	Signal Description	Direction	I/O Standard
Analog to Digital Converter (AD7843) Interface				
top_HSMB_ADC_CS_N	W22	Chip select for ADC	Output	2.5 V
top_HSMB_ADC_DCLK	AD28	Data clock for ADC	Output	2.5 V
top_HSMB_ADC_DIN	AD27	Data input for ADC	Output	2.5 V
top_HSMB_ADC_DOUT	U28	Data output for ADC	Input	2.5 V
top_HSMB_ADC_PENIRQ_N	Y27	Pen Interrupt (active low) ADC	Input	2.5 V
top_HSMB_DEN	V23	Data enable For ADC	Output	2.5 V
Touch Panel Interface				
top_HSMB_SCEN	AB27	Serial clock enable	Output	2.5 V
top_HSMB_SDAT	F28	Serial data enable	Bidirectional	2.5 V
Video Pipeline Interface				
top_HSMB_LCD_NCLK	AC26	Clock for LCD touch panel	Output	2.5 V
top_HSMB_HSYNC	W27	Horizontal sync	Output	2.5 V
top_HSMB_VSYNC	W28	Vertical sync	Output	2.5 V
top_HSMB_LCD_DATA[0]	Y22	Multiplexed RGB data for LCD touch panel	Output	2.5 V
top_HSMB_LCD_DATA[1]	V21	Multiplexed RGB data for LCD touch panel	Output	2.5 V
top_HSMB_LCD_DATA[2]	W21	Multiplexed RGB data for LCD touch panel	Output	2.5 V
top_HSMB_LCD_DATA[3]	U22	Multiplexed RGB data for LCD touch panel	Output	2.5 V
top_HSMB_LCD_DATA[4]	V22	Multiplexed RGB data for LCD touch panel	Output	2.5 V
top_HSMB_LCD_DATA[5]	AB28	Multiplexed RGB data for LCD touch panel	Output	2.5 V
top_HSMB_LCD_DATA[6]	AC27	Multiplexed RGB data for LCD touch panel	Output	2.5 V
top_HSMB_LCD_DATA[7]	AC28	Multiplexed RGB data for LCD touch panel	Output	2.5 V
10/100 Ethernet Interface (National PHY DP83848C)				
top_HSMB_ETH_RESET_N	AA26	Ethernet PHY reset (active low)	Output	2.5 V
top_HSMB_MDC	AE28	Management data clock	Output	2.5 V
top_HSMB_MDIO	J26	Management data IO	Bidirectional	2.5 V
top_HSMB_RX_COL	T25	MII collision detect	Input	2.5 V
top_HSMB_RX_CRS	P28	MII carrier sense/receive	Input	2.5 V
top_HSMB_RX_CLK	J27	MII receive clock	Input	2.5 V
top_HSMB_RX_D[0]	P21	Receive data	Input	2.5 V
top_HSMB_RX_D[1]	R21	Receive data	Input	2.5 V
top_HSMB_RX_D[2]	R22	Receive data	Input	2.5 V

Table 1. Signal Descriptions (Part 7 of 7)

Pin Name	Location	Signal Description	Direction	I/O Standard
top_HSMB_RX_D[3]	R23	Receive data	Input	2.5 V
top_HSMB_RX_DV	T26	Receive data valid	Input	2.5 V
top_HSMB_RX_ERR	P27	Receive error	Input	2.5 V
top_HSMB_TX_CLK	Y28	MII transmit clock	Input	2.5 V
top_HSMB_TX_D[0]	AB26	Transmit data	Output	2.5 V
top_HSMB_TX_D[1]	Y23	Transmit data	Output	2.5 V
top_HSMB_TX_D[2]	Y24	Transmit data	Output	2.5 V
top_HSMB_TX_D[3]	AE27	Transmit data	Output	2.5 V
top_HSMB_TX_EN	AB25	Transmit enable	Output	2.5 V
SD/MMC Card Interface				
top_HSMB_SD_CLK	V25	SD clock	Output	2.5 V
top_HSMB_SD_CMD	H24	SD command	Output	2.5 V
top_HSMB_SD_DAT	F27	SD data	Bidirectional	2.5 V
top_HSMB_SD_DAT1	G28	SD data	Bidirectional	2.5 V
top_HSMB_SD_DAT2	K25	SD data	Bidirectional	2.5 V
top_HSMB_SD_DAT3	L23	SD data3	Output	2.5 V
EEPROM (24LC00) Interface				
top_HSMB_ID_I2CDAT	H23	Serial data	Bidirectional	2.5 V
top_HSMB_ID_I2CSCL	G24	Serial clock	Output	2.5 V
RS-232 UART Transceiver ADM3202 Interface				
top_HSMB_UART_RXD	Y26	UART receive data	Input	2.5 V
top_HSMB_UART_TXD	AA25	UART transmit data	Output	2.5 V
LED Interface				
top_led[0]	AD15	LED output	Output	1.8 V
top_led[1]	AE20	LED output	Output	1.8 V
top_led[2]	AF18	LED output	Output	1.8 V
top_led[3]	AD19	LED output	Output	1.8 V
top_led[4]	AE15	LED output	Output	1.8 V
top_led[5]	AC17	LED output	Output	1.8 V
top_led[6]	AG19	LED output	Output	1.8 V
top_led[7]	AF19	LED output	Output	1.8 V
Push Button Interface				
top_button[0]	AD7	Parallel I/O for button 0	Input	1.8 V
top_button[1]	AC12	Parallel I/O for button 1	Input	1.8 V

Table 2 lists the device pin-outs.

Table 2. Device Pin-Outs (Part 1 of 6)

Pin Name	Location	Direction	I/O Standard	I/O Bank
top_fsd[16]	A4	Bidirectional	1.8 V	8
top_ddr2top_odt	A6	Output	SSTL-18 Class I	8
top_ddr2top_dq[10]	A7	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dm[1]	A8	Output	SSTL-18 Class I	8
top_ddr2top_wen	A10	Output	SSTL-18 Class I	8
top_ddr2top_dq[2]	A11	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[0]	A12	Bidirectional	SSTL-18 Class I	8
top_clkin_125	A14	Input	1.8 V	8
top_ddr2top_a[10]	A17	Output	SSTL-18 Class I	7
top_fsd[24]	A18	Bidirectional	1.8 V	7
top_fsa[13]	AA8	Output	1.8 V	3
top_button[3]	AA12	Input	1.8 V	3
top_fsa[2]	AA13	Output	1.8 V	3
top_flash_wen	AA21	Output	1.8 V	4
top_HSMB_UART_TXD	AA25	Output	2.5 V	5
top_HSMB_ETH_RESET_N	AA26	Output	2.5 V	5
top_fsa[24]	AB7	Output	1.8 V	3
top_ddr2bot_a[7]	AB8	Output	SSTL-18 Class I	3
top_ddr2bot_a[9]	AB10	Output	SSTL-18 Class I	3
top_ddr2bot_a[4]	AB12	Output	SSTL-18 Class I	3
top_fsa[17]	AB14	Output	1.8 V	3
top_ddr2bot_dq[15]	AB16	Bidirectional	SSTL-18 Class I	4
top_fsa[20]	AB18	Output	1.8 V	4
top_flash_resetn	AB20	Output	1.8 V	4
top_ddr2bot_a[0]	AB22	Output	SSTL-18 Class I	4
top_HSMB_TX_EN	AB25	Output	2.5 V	5
top_HSMB_TX_D[0]	AB26	Output	2.5 V	5
top_HSMB_SCEN	AB27	Output	2.5 V	5
top_HSMB_LCD_DATA[5]	AB28	Output	2.5 V	5
top_ddr2bot_a[5]	AC7	Output	SSTL-18 Class I	3
top_fsa[3]	AC10	Output	1.8 V	3
top_fsa[0]	AC11	Output	1.8 V	3
top_button[1]	AC12	Input	1.8 V	3
top_ddr2bot_dm[1]	AC15	Output	SSTL-18 Class I	4
top_led[5]	AC17	Output	1.8 V	4
top_ddr2bot_csn	AC21	Output	SSTL-18 Class I	4
top_fsa[14]	AC22	Output	1.8 V	4
top_HSMB_LCD_NCLK	AC26	Output	2.5 V	5
top_HSMB_LCD_DATA[6]	AC27	Output	2.5 V	5

Table 2. Device Pin-Outs (Part 2 of 6)

Pin Name	Location	Direction	I/O Standard	I/O Bank
top_HSMB_LCD_DATA[7]	AC28	Output	2.5 V	5
top_button[0]	AD7	Input	1.8 V	3
top_fsa[8]	AD8	Output	1.8 V	3
top_ddr2bot_a[6]	AD12	Output	SSTL-18 Class I	3
top_led[0]	AD15	Output	1.8 V	4
top_ddr2bot_dq[4]	AD17	Bidirectional	SSTL-18 Class I	4
top_led[3]	AD19	Output	1.8 V	4
top_ddr2bot_casn	AD21	Output	SSTL-18 Class I	4
top_fsa[11]	AD24	Output	1.8 V	4
top_HSMB_ADC_DIN	AD27	Output	2.5 V	5
top_HSMB_ADC_DCLK	AD28	Output	2.5 V	5
top_ddr2bot_a[10]	AE4	Output	SSTL-18 Class I	3
top_ddr2bot_wen	AE5	Output	SSTL-18 Class I	3
top_ddr2bot_a[3]	AE7	Output	SSTL-18 Class I	3
top_fsa[15]	AE8	Output	1.8 V	3
top_ddr2_ck_p[0]	AE14	Bidirectional	SSTL-18 Class I	3
top_led[4]	AE15	Output	1.8 V	4
top_fsa[23]	AE16	Output	1.8 V	4
top_ddr2bot_dq[13]	AE17	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dqs[0]	AE18	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dq[6]	AE19	Bidirectional	SSTL-18 Class I	4
top_led[1]	AE20	Output	1.8 V	4
top_ddr2bot_rasn	AE21	Output	SSTL-18 Class I	4
top_ddr2bot_odt	AE24	Output	SSTL-18 Class I	4
top_HSMB_TX_D[3]	AE27	Output	2.5 V	5
top_HSMB_MDC	AE28	Output	2.5 V	5
top_ddr2bot_ba[0]	AF3	Output	SSTL-18 Class I	3
top_fsa[7]	AF4	Output	1.8 V	3
top_ddr2bot_ba[1]	AF5	Output	SSTL-18 Class I	3
top_fsa[12]	AF9	Output	1.8 V	3
top_fsa[16]	AF13	Output	1.8 V	3
top_ddr2_ck_n[0]	AF14	Bidirectional	SSTL-18 Class I	3
top_ddr2bot_dq[12]	AF15	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dq[14]	AF16	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dqs[1]	AF17	Bidirectional	SSTL-18 Class I	4
top_led[2]	AF18	Output	1.8 V	4
top_led[7]	AF19	Output	1.8 V	4
top_ddr2bot_a[11]	AF21	Output	SSTL-18 Class I	4
top_fsa[5]	AF22	Output	1.8 V	4

Table 2. Device Pin-Outs (Part 3 of 6)

Pin Name	Location	Direction	I/O Standard	I/O Bank
top_fsa[18]	AF23	Output	1.8 V	4
top_ddr2bot_dq[7]	AF24	Bidirectional	SSTL-18 Class I	4
top_fsa[6]	AF26	Output	1.8 V	4
top_fsa[22]	AG3	Output	1.8 V	3
top_ddr2bot_cke	AG4	Output	SSTL-18 Class I	3
top_ddr2bot_a[1]	AG6	Output	SSTL-18 Class I	3
top_fsa[19]	AG12	Output	1.8 V	3
top_ddr2bot_dq[9]	AG17	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dq[8]	AG18	Bidirectional	SSTL-18 Class I	4
top_led[6]	AG19	Output	1.8 V	4
top_ddr2bot_dq[3]	AG21	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dq[0]	AG22	Bidirectional	SSTL-18 Class I	4
top_fsa[9]	AG26	Output	1.8 V	4
top_button[2]	AH3	Input	1.8 V	3
top_fsa[10]	AH6	Output	1.8 V	3
top_fsa[1]	AH10	Output	1.8 V	3
top_ddr2bot_a[8]	AH12	Output	SSTL-18 Class I	3
top_clkin_50	AH15	Input	1.8 V	4
top_ddr2bot_dq[11]	AH17	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dq[10]	AH18	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dm[0]	AH19	Output	SSTL-18 Class I	4
top_ddr2bot_dq[1]	AH21	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dq[2]	AH22	Bidirectional	SSTL-18 Class I	4
top_ddr2bot_dq[5]	AH23	Bidirectional	SSTL-18 Class I	4
top_ddr2top_dq[13]	B6	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[8]	B7	Bidirectional	SSTL-18 Class I	8
top_fsd[19]	B8	Bidirectional	1.8 V	8
top_ddr2top_dm[0]	B10	Output	SSTL-18 Class I	8
top_fsd[22]	B11	Bidirectional	1.8 V	8
top_fsd[23]	B22	Bidirectional	1.8 V	7
top_fsd[5]	C6	Bidirectional	1.8 V	8
top_ddr2top_csn	C7	Output	SSTL-18 Class I	8
top_fsd[20]	C8	Bidirectional	1.8 V	8
top_fsd[28]	C9	Bidirectional	1.8 V	8
top_ddr2top_dq[11]	C10	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[9]	C11	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[5]	C12	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[3]	C13	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[1]	C14	Bidirectional	SSTL-18 Class I	8

Table 2. Device Pin-Outs (Part 4 of 6)

Pin Name	Location	Direction	I/O Standard	I/O Bank
top_we_n_to_the_max2	C15	Output	1.8 V	7
top_fsd[7]	C18	Bidirectional	1.8 V	7
top_ddr2top_ba[0]	C23	Output	SSTL-18 Class I	7
top_fsd[1]	D6	Bidirectional	1.8 V	8
top_ddr2top_a[4]	D7	Output	SSTL-18 Class I	8
top_ddr2top_a[11]	D8	Output	SSTL-18 Class I	8
top_fsd[27]	D9	Bidirectional	1.8 V	8
top_ddr2top_dq[15]	D10	Bidirectional	SSTL-18 Class I	8
top_fsd[12]	D11	Bidirectional	1.8 V	8
top_ddr2top_dqs[0]	D12	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[7]	D13	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[4]	D15	Bidirectional	SSTL-18 Class I	7
top_ddr2top_a[9]	D16	Output	SSTL-18 Class I	7
top_fsd[8]	D18	Bidirectional	1.8 V	7
top_ddr2top_ba[1]	D19	Output	SSTL-18 Class I	7
top_cs_n_to_the_max2	D23	Output	1.8 V	7
top_ddr2top_a[3]	D24	Output	SSTL-18 Class I	7
top_ddr2top_a[12]	D25	Output	SSTL-18 Class I	7
top_fsd[29]	E7	Bidirectional	1.8 V	8
top_ddr2top_a[2]	E8	Output	SSTL-18 Class I	8
top_ddr2top_dq[12]	E11	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dqs[1]	E12	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[6]	E14	Bidirectional	SSTL-18 Class I	8
top_ddr2top_cke	E21	Output	SSTL-18 Class I	7
top_fsd[13]	E24	Bidirectional	1.8 V	7
top_oe_n_to_the_max2	E25	Output	1.8 V	7
top_fsd[21]	F7	Bidirectional	1.8 V	8
top_ddr2top_rasn	F8	Output	SSTL-18 Class I	8
top_fsd[11]	F12	Bidirectional	1.8 V	8
top_ddr2top_casn	F14	Output	SSTL-18 Class I	8
top_fsd[4]	F18	Bidirectional	1.8 V	7
top_HSMB_SD_DAT	F27	Input	2.5 V	6
top_HSMB_SDAT	F28	Bidirectional	2.5 V	6
top_fsd[3]	G7	Bidirectional	1.8 V	8
top_fsd[25]	G8	Bidirectional	1.8 V	8
top_fsd[15]	G9	Bidirectional	1.8 V	8
top_ddr2_ck_n[1]	G11	Bidirectional	SSTL-18 Class I	8
top_fsd[17]	G13	Bidirectional	1.8 V	8
top_fsd[9]	G16	Bidirectional	1.8 V	7

Table 2. Device Pin-Outs (Part 5 of 6)

Pin Name	Location	Direction	I/O Standard	I/O Bank
top_ddr2top_a[1]	G18	Output	SSTL-18 Class I	7
top_fsd[10]	G22	Bidirectional	1.8 V	7
top_HSMB_ID_I2CSCL	G24	Output	2.5 V	6
top_HSMB_SD_DAT2	G28	Output	2.5 V	6
top_ddr2top_a[8]	H8	Output	SSTL-18 Class I	8
top_fsd[30]	H10	Bidirectional	1.8 V	8
top_ddr2_ck_p[1]	H12	Bidirectional	SSTL-18 Class I	8
top_ddr2top_dq[14]	H13	Bidirectional	SSTL-18 Class I	8
top_fsd[18]	H14	Bidirectional	1.8 V	8
top_ddr2top_a[6]	H15	Output	SSTL-18 Class I	7
top_fsd[6]	H17	Bidirectional	1.8 V	7
top_fsd[14]	H21	Bidirectional	1.8 V	7
top_HSMB_ID_I2CDAT	H23	Bidirectional	2.5 V	6
top_HSMB_SD_CMD	H24	Output	2.5 V	6
top_fsd[31]	J10	Bidirectional	1.8 V	8
top_fsd[26]	J12	Bidirectional	1.8 V	8
top_ddr2top_a[0]	J13	Output	SSTL-18 Class I	8
top_fsd[0]	J14	Bidirectional	1.8 V	8
top_ddr2top_a[5]	J15	Output	SSTL-18 Class I	7
top_ddr2top_a[7]	J16	Output	SSTL-18 Class I	7
top_fsd[2]	J17	Bidirectional	1.8 V	7
top_HSMB_MDIO	J26	Bidirectional	2.5 V	6
top_HSMB_RX_CLK	J27	Input	2.5 V	6
top_HSMB_SD_DAT1	K25	Output	2.5 V	6
top_HSMB_SD_DAT3	L23	Output	2.5 V	6
~ALTERA_DATA0~ / RESERVED_INPUT	N7	Input	2.5 V	1
~ALTERA_DCLK~ / RESERVED_INPUT	P3	Input	2.5 V	1
top_HSMB_RX_D[0]	P21	Input	2.5 V	5
top_HSMB_RX_ERR	P27	Input	2.5 V	6
top_HSMB_RX_CRS	P28	Input	2.5 V	6
top_HSMB_RX_D[1]	R21	Input	2.5 V	5
top_HSMB_RX_D[2]	R22	Input	2.5 V	5
top_HSMB_RX_D[3]	R23	Input	2.5 V	5
top_reset_n	T21	Input	2.5 V	5
top_HSMB_RX_COL	T25	Input	2.5 V	5
top_HSMB_RX_DV	T26	Input	2.5 V	5
top_HSMB_LCD_DATA[3]	U22	Output	2.5 V	5
top_HSMB_ADC_DOUT	U28	Input	2.5 V	5

Table 2. Device Pin-Outs (Part 6 of 6)

Pin Name	Location	Direction	I/O Standard	I/O Bank
top_HSMB_LCD_DATA[1]	V21	Output	2.5 V	5
top_HSMB_LCD_DATA[4]	V22	Output	2.5 V	5
top_HSMB_DEN	V23	Output	2.5 V	5
top_HSMB_SD_CLK	V25	Output	2.5 V	5
top_HSMB_LCD_DATA[2]	W21	Output	2.5 V	5
top_HSMB_ADC_CS_N	W22	Output	2.5 V	5
top_HSMB_HSYNC	W27	Output	2.5 V	5
top_HSMB_VSYNC	W28	Output	2.5 V	5
top_ddr2bot_a[12]	Y12	Output	SSTL-18 Class I	3
top_ddr2bot_a[2]	Y13	Output	SSTL-18 Class I	3
top_fsa[4]	Y15	Output	1.8 V	3
top_flash_cen	Y16	Output	1.8 V	4
top_flash_oen	Y17	Output	1.8 V	4
top_fsa[21]	Y19	Output	1.8 V	4
top_HSMB_LCD_DATA[0]	Y22	Output	2.5 V	5
top_HSMB_TX_D[1]	Y23	Output	2.5 V	5
top_HSMB_TX_D[2]	Y24	Output	2.5 V	5
top_HSMB_UART_RXD	Y26	Input	2.5 V	5
top_HSMB_ADC_PENIRQ_N	Y27	Input	2.5 V	5
top_HSMB_TX_CLK	Y28	Input	2.5 V	5

Device Specifications

The microprocessor consists of a soft core processor and other components within an FPGA device.

Refer to the *Cyclone III Device Handbook* for FPGA device specific information, such as device core architecture, I/O considerations, power considerations, external memory (including interfaces, configuration, hot socketing, and remote system update), and packaging information.

For specific information about electrical DC and switching characteristics, refer to the *DC and Switching Characteristics* chapter in volume 2 of the *Cyclone III Device Handbook*.

Address Map

Table 3 defines the address mapping for the reference design from the viewpoint of the instruction and data masters of the Nios II processor. Each peripheral might have several ports. For instance, `pll.s1` designates a slave port for the phase locked loop (PLL) component.

Table 3. Address Map

Port Name	cpu.instruction_master	cpu.data_master
pll.s1	—	0x080000c0 – 0x080000ff
cpu.jtag_debug_module	0x09001800 – 0x09001fff	0x09001800 – 0x09001fff
descriptor_memory.s1	—	0x09000000 – 0x09000fff
pipeline_bridge_before_tristate_bridge.s1	0x10000000 – 0x17ffffff	0x10000000 – 0x17ffffff
flash_tristate_bridge.avalon_slave	—	—
ext_flash.s1	0x10000000 – 0x13ffffff	0x10000000 – 0x13ffffff
max2.s1	0x14000000 – 0x1400001f	0x14000000 – 0x1400001f
sls_sdhc.avalon_slave	—	0x09002c00 – 0x09002c7f
sdhc_ddr_clock_bridge.s1	—	—
ddr2_sdram.s1	—	0x00000000 – 0x03ffffff
ddr2_sdram_1.s1	0x1c000000 – 0x1fffffff	0x1c000000 – 0x1fffffff
cpu_ddr_clock_bridge.s1	—	0x00000000 – 0x07ffffff
cpu_ddr_1_clock_bridge.s1	0x1c000000 – 0x1fffffff	0x1c000000 – 0x1fffffff
slow_peripheral_bridge.s1	—	0x08000000 – 0x08003fff
tse_ddr_clock_bridge.s1	—	—
sgdma_tx.csr	—	0x09002400 – 0x090027ff
sgdma_rx.csr	—	0x09002000 – 0x090023ff
lcd_sgdma.csr	—	0x04000000 – 0x040003ff
tse_mac.control_port	—	0x09002800 – 0x09002bff
sys_clk_timer.s1	—	0x08000100 – 0x0800013f
high_res_timer.s1	—	0x08000000 – 0x0800003f
sysid.control_slave	—	0x08000290 – 0x0800029f
performance_counter.control_slave	—	0x08000080 – 0x080000bf
jtag_uart.avalon_jtag_slave	—	0x08000280 – 0x0800028f
uart1.s1	—	0x08000040 – 0x0800007f
button_pio.s1	—	0x08000180 – 0x0800019f
led_pio.s1	—	0x080001a0 – 0x080001bf
pio_id_eeprom_scl.s1	—	0x08000240 – 0x0800025f
lcd_i2c_scl.s1	—	0x080001e0 – 0x080001ff
lcd_i2c_en.s1	—	0x080001c0 – 0x080001df
pio_id_eeprom_dat.s1	—	0x08000220 – 0x0800023f
lcd_i2c_sdat.s1	—	0x08000200 – 0x0800021f
touch_panel_pen_irq_n.s1	—	0x08000260 – 0x0800027f
touch_panel_spi.spi_control_port	—	0x08000140 – 0x0800017f

System Architecture

The microprocessor design is based on the Nios II/f core and provides a typical mix of peripherals, memories, and a video pipeline. The design provides an interface to each hardware component on the Altera Embedded Systems Development Kit, Cyclone III Edition, such as DDR2 SDRAM, LEDs, RS-232 connector, Ethernet MAC/10/100 PHY, and 800 × 480 pixel LCD. The video pipeline provides high bandwidth memory access that allows for flicker-free display on the color LCD.

Design Considerations

Use of Clock Domains

To increase the overall f_{MAX} performance, exercise prudent use of independent clock domains where slow peripheral components run in a slower clock region than the Nios II processor and other memory components co-exist. Your attentiveness can increase the overall f_{MAX} of the design without adding additional memory latency.

The Nios II processor runs at a frequency of 100 MHz and is connected to high performance DDR2 SDRAM memory, on-chip descriptor memory, and CFI flash memory. Clock crossing bridges are required between the Nios II processor and the DDR2 SDRAM memories and slow peripherals components because these components run in different clock regions. A pipeline bridge between the Nios II processor and the flash tri-state bridge to external flash component ensures system f_{MAX} is not affected.

The DDR2 SDRAM memories run at 153.85 MHz for program memory and 153.85 MHz for video data memory. The `ddr2_sdram_1` memory core stores program code and runs at full rate at local interface with a 32-bit data width, connected to a Nios II 32-bit data bus. The `ddr2_sdram` memory core stores video frames and runs at half rate at local interface with a 64-bit data width, connected with the 64-bit data width `lcd_sgdma`.

Summary of Clock Domains

The system assumes an oscillator clock of 50 MHz and uses a PLL to generate the rest of the clocks in the system.

- `pll` is driven by 50-MHz oscillator clock using two generated clocks:
  - `pll_c0`, 100 MHz: used as system clock for Nios II processor and video clock for LCD touch panel circuitry
  - `pll_c2`, 60 MHz: used as slow peripherals clock
- `ddr2_sdram` is driven by 50-MHz oscillator clock, and run at 153.9 MHz
  - run at half-rate 76.95 MHz at local interface for video data memory (64-bit)
- `ddr2_sdram_1` is driven by 125-MHz oscillator clock, and run at 153.9 MHz
  - run at full-rate 153.9 MHz at local interface for Nios II processor program memory (32 -bit) connected to the instruction master

- 100-MHz clock region:
  - Nios II processor
  - Descriptor memory
  - CFI flash
  - MAX II
  - Triple speed ethernet components
  - Video pipeline components
- 60-MHz clock region:
  - System peripherals
  - SD MMC components
  - JTAG UART
  - UART
  - PIOs
- 76.95-MHz clock region (must run at the same speed as `ddr2_sdram`):
  - `lcd_sgdma` SG-DMA, 64-bit data width
  - `lcd_ta_sgdma_to_fifo` timing adapter

Use of Bridges

Bridges control the topology of an SOPC Builder system. Without bridges, SOPC Builder generates a system interconnect fabric with maximum parallelism, where all masters drive slaves concurrently, as long as each master accesses different slaves. For systems that do not require such a high degree of concurrency, you can use a bridge to control topology and provide optimal performance.

For more information, refer to the *Avalon Memory-Mapped Bridges* chapter in volume 4 of the *Quartus II Handbook*.

Bridges are used for the following reasons:

- Clock crossing between two components clocked at different frequencies:
  - `cpu_ddr_1_clock_bridge`: between Nios II processor and `ddr2_sdram_1`
  - `cpu_ddr_clock_bridge`: between Nios II processor and `ddr2_sdram`
  - `tse_ddr_clock_bridge`: between the `sgdma_rx`, `sgdma_tx` and `ddr2_sdram`
  - `slow_peripheral_bridge`: between Nios II processor and slow peripheral components
- Increase the performance and f_{MAX} of the path between two components:
  - `pipeline_bridge_before_tristate_bridge`: between Nios II and `flash_tristate_bridge`

The triple speed ethernet components run in the 100-MHz clock region:

- `tse_mac`

- sgdma_tx
- sgdma_rx
- descriptor_memory

The following components are partitioned by the slow_peripheral_bridge. This allows non-f_{MAX} critical components to be clocked at a slower clock rate to give better fitting for components that are f_{MAX}-critical. Unless otherwise noted, all the components run in the 60-MHz clock region.

- sys_clk_timer
- high_res_timer
- performance_counter
- jtag_uart
- uart
- sysid
- pll (50-MHz external clock)
- button_pio
- led_pio
- pio_id_eeprom_dat
- pio_id_eeprom_scl
- touch_panel_spi
- touch_panel_pen_irq_n
- lcd_i2c_scl
- lcd_i2c_en
- lcd_i2c_sdat
- sls_sdhc_sd_card_controller

Arbitration Priority Considerations

Arbitration priority allows the Avalon® Memory-Mapped (Avalon-MM) interface masters to obtain the required bandwidth for transfers in the SOPC Builder system. For the Nios II processor, the optimum value of arbitration priority is eight in this system, and uses 32-byte instruction and data cache line sizes. The scatter-gather direct memory access (SG-DMA) requires an arbitration priority value of eight for descriptor read and write operations. To ensure video pipelining components operate smoothly, highest arbitration priority is given to lcd_sgdma to access the ddr_sdram memory component.

Interrupt Priority Considerations

The Nios II system processes interrupts from components with the lowest IRQ value first. In this system, priority is given to the video pipeline component, then the triple speed ethernet component, timers, JTAG UART, and lastly the slow peripheral components.

Processor

Processor Core

Processor name: cpu

Version: 8.1

Processor type: Fast (Nios /f)

- 32-bit RISC
- Instruction cache
- Data cache
- Branch prediction
- Hardware multiply
- Hardware divide
- Barrel shifter
- Dynamic branch prediction

Nominal metrics:

- Nominal performance at 100 MHz: Up to 113 DMIPS
- Nominal logic usage: 1400-1800 LEs
- Nominal memory usage: Three M9K + Cache

Reset vector:

- Memory: ext_flash
- Offset: 0x0
- Physical address: 0x10000000

Exception vector:

- Memory: ddr_sdram_1
- Offset: 0x40
- Physical address: 0x1c000040

Memory Management Unit

MMU option: Disabled

MPU option: Disabled

Cache

Instruction master:

- Cache size: 32 KBytes

Data master:

- Cache size: 32 KBytes
- Data cache line size: 32 Bytes

JTAG Debug Module (Level 1)

- JTAG target connection
- Download software
- Software breakpoints
- Nominal logic usage: 300–400 LEs
- Nominal memory usage: Two M9Ks

Custom Instructions

Custom Instructions Enabled: None

Processor Instruction Set Reference

For details about the Nios II processor core, the processor architecture, exception and interrupt control, JTAG debug module, programming model, and instruction set architecture, refer to the *Nios II Processor Reference Handbook*.

System Clock Settings

The system expects two external clock sources named `clk` (50 MHz) and `clk_125` (125 MHz). A PLL creates the various clocks required for the clock domains and external memories from the 50-MHz clock source. Table 4 shows the clock sources for the system.

Table 4. Clock Sources

Name	Source	MHz
`clk`	external	50.0
`clk_125`	external	125.0
`pll_c0`	`pll.c0`	100.0
`pll_c2`	`pll.c2`	60.0
`ddr2_sdram_sysclk`	`ddr2_sdram.sysclk`	76.95
`ddr2_sdram_auxfull`	`ddr2_sdram.auxfull`	153.9
`ddr2_sdram_auxhalf`	`ddr2_sdram.auxhalf`	76.95
`ddr2_sdram_1_sysclk`	`ddr2_sdram_1.sysclk`	166.6
`ddr2_sdram_1_auxfull`	`ddr2_sdram_1.auxfull`	166.6
`ddr2_sdram_1_auxhalf`	`ddr2_sdram_1.auxhalf`	83.3

For more information about the PLL megafunction, refer to the *altpll Megafunction User Guide*.

System Boot and Configuration

MAX II Configuration Interface

The microprocessor is configured for passive serial configuration on power up. You configure the device through the MAX II parallel flash loader (PFL) feature. The MAX II PFL provides an efficient method to program CFI flash memory devices through the JTAG interface and the logic to control configuration from the flash memory device to the FPGA.

For more information about passive serial and other configuration schemes, refer to the *Configuring Cyclone III Devices* chapter in volume 1 of the *Cyclone III Device Handbook*.

The MAX II device is accessed through shared address and data bus with the CFI flash. To start the configuration, the configuration file is loaded into CFI flash. Then the PSR (page select register) in the MAX II device is set with the location of the file and the MAX II device is set to perform system reset. The PFL configures the FPGA with configuration data stored in CFI flash.

For more information on the MAX II PFL, refer to *AN386: Using the Parallel Flash Loader with the Quartus II Software*.

System ID

The Altera System ID core is a simple read-only device that provides SOPC Builder systems with a unique identifier. Nios II processor systems use the system ID core to verify that an executable program was compiled targeting the actual hardware image configured in the FPGA.

For more information about the system ID core, refer to the *System ID Core* chapter in volume 5 of the *Quartus II Handbook*.

Timers

High Resolution Timer

Use a high resolution Altera interval timer core as an interval timekeeper, watch-dog timer, or counter. Table 5 shows the high resolution timer parameters and corresponding parameter values.

Table 5. High Resolution Timer Parameters

Parameter	Value
Name	`high_res_timer`
Time-out period	10 ms
Timer counter size	32 bits
Hardware presets	Full featured

For more information about the high resolution interval timer core, refer to the *Timer Core* chapter in volume 5 of the *Quartus II Handbook*.

Performance Counter

The Altera performance counter core measures software performance by tracking the time to execute one or more functions. A performance counter is useful for software engineers during software development. Table 6 shows the performance counter parameters and corresponding parameter values.

Table 6. Performance Counter Parameters

Parameter	Value
Name	`performance_counter`
Simultaneously-measured sections	1
Counter width	32 bits

For more information about this core, refer to the *Performance Counter Core* chapter in volume 5 of the *Quartus II Handbook*.

PIOs

The Altera programmed input/output (PIO) cores transfer data between the processor and certain input/output (I/O) devices. The microprocessor includes LED and Button PIOs. Table 7 shows the LED PIO parameters and corresponding parameter values.

Table 7. LED PIO Parameters

Parameter	Value
Name	`led_pio`
Width (1-32 bits)	8
Direction	Output ports only
Output port reset value	0x0

Table 8 shows the Button PIO parameters and corresponding parameter values.

Table 8. Button PIO Parameters

Parameter	Value
Name	`button_pio`
Width (1-32 bits)	4
Direction	Input ports only
Input options	■ Edge capture register: synchronous capture, rising edge ■ Interrupt generated: edge based

For information about these cores, refer to the *PIO Core* chapter in volume 5 of the *Quartus II Handbook*.

EEPROM ID Interface

The EEPROM ID interface is a 2-bit interface to an I2C serial EEPROM constructed using the PIOs, `pio_id_eeprom_dat` and `pio_id_eeprom_scl`. Table 9 shows the EEPROM I2C data PIO parameters and corresponding parameter values.

Table 9. EEPROM I2C Data PIO Parameters

Parameter	Value
Name	pio_id_eeprom_dat
Width (1-32 bits)	1
Direction	Bidirectional (tri-state) ports
Output port reset value	0x0

Table 10 shows the EEPROM I2C clock PIO parameters and corresponding parameter values.

Table 10. EEPROM I2C Clock PIO Parameters

Parameter	Value
Name	pio_id_eeprom_scl
Width (1-32 bits)	1
Direction	Output ports only
Output port reset value	0x0

For information about these cores, refer to the *PIO Core* chapter in volume 5 of the *Quartus II Handbook*.

Memory Interface

CFI Flash Interface

The Altera CFI-compliant flash memory controller core controls an external flash device (Spansion Flash). This flash device stores both application program code and FPGA configuration data. With 64 MByte capacity, it is possible to store multiple configuration images in flash memory and configure the FPGA with one of the images. Table 11 shows the CFI flash interface parameters and corresponding parameter values.

Table 11. CFI Flash Interface Parameters

Parameter	Value
Name	`ext_flash`
Address width (bits)	25
Data width	16
Flash capacity	64 MByte
Timing settings	■ Setup: 80 ns ■ Wait: 40 ns ■ Hold: 20 ns

For more information about this core, refer to the *Common Flash Interface Controller Core* chapter in volume 5 of the *Quartus II Handbook*.

DDR2 SDRAM Memory Controller

The Altera DDR2 SDRAM High Performance MegaCore function is used to interface to a Micron MT4732M16CC-3 DDR2 SDRAM device, creating a video frame buffer, and Ethernet data transmit and receive buffer. The video data (RGB) is stored in the video frame buffer in unpacked 64-bit format. An SG-DMA is used to transfer the 64-bit-wide video stream from the memory into the video pipeline. The MegaCore function is configured to a 32-bit width clocked at 153.9 MHz. The local interface to the SG-DMA is configured to a 64-bit width clocked at 76.95 MHz and hence the memory controller is configured for half-rate operation. Table 12 shows the MegaCore function parameters and corresponding parameter values.

Table 12. DDR2 SDRAM Memory Controller Parameters

Parameter	Value
Name	ddr2_sdram
PLL reference clock frequency	50 MHz
Memory clock frequency	153.9 MHz
Local interface clock frequency	76.95 MHz
Local interface width	64 bits

For more information about this MegaCore function, refer to the *DDR and DDR2 SDRAM High-Performance Controller User Guide*.

DDR2 SDRAM 1 Memory Controller

The Altera DDR2 SDRAM High Performance MegaCore function is used to interface to a Micron MT4732M16CC-3 DDR2 SDRAM device to store program code and data. To maximize system performance, the memory controller is configured for full-rate operation. Table 13 shows the MegaCore function parameters and corresponding parameter values.

Table 13. DDR2 SDRAM 1 Memory Controller Parameters

Parameter	Value
Name	ddr2_sdram_1
PLL reference clock frequency	125 MHz
Memory clock frequency	153.9 MHz
Local interface clock frequency	153.9 MHz
Local interface width	32 bits

For more information about this core, refer to the *DDR and DDR2 SDRAM High-Performance Controller User Guide*.

SD/MMC SPI

The SD/MMC SPI core connects to standard multimedia card (MMC) and secure digital (SD) flash based memory devices. The MMC and SD card are universal low cost data storage memories. The SD/MMC SPI core is available from SLS Corp. in encrypted format, for evaluation purposes. The SD/MMC SPI core also comes with low-level driver routines to access the MMC and SD devices.

Features:

- Support both SD 1-bit and 4-bit mode for data communication.
- Variable SD clock frequency selection using software.
- Internal FIFO for data transmit/receive operation.
- Hardware implementation of CRC7 and CRC16 module for generation and verification.
- Variable block length support.
- Multiple block transfer support.
- Support for interrupt driven functionality.
- 8-bit internal DMA engine for data transfer.
- Follows SDA v2.0 specification.

Table 14 shows the SD/MMC SPI parameters and corresponding parameter values.

Table 14. SD/MMC SPI Parameters

Parameter	Value
Name	`sls_sdhc`
Avalon bus clock frequency	60 MHz
SPI clock frequency	20 MHz

For detailed information about the SLS SD Card Host Controller core, refer to http://www.slscorp.com/pages/ip_sdhostcontroller.php. Purchase this core directly from SLS Corp.

Communications Interface

Ethernet Interface

Because Ethernet has become a common communication interface for embedded systems, this data sheet provides a pre-generated Ethernet solution featuring the Altera Triple Speed Ethernet MegaCore function and NicheStack TCP/IP Network Stack, Nios II Edition.

Triple Speed Ethernet MegaCore Function

Table 15 shows the Triple Speed Ethernet MegaCore function parameters and corresponding parameter values.

Table 15. Triple Speed Ethernet MegaCore Function Parameters (Part 1 of 2)

Parameter	Value
Name	`tse_mac`
Core Configuration	
Core variation	10/100/1000 Mbit Ethernet MAC
Interface	MII/GMII
Use internal FIFO	Enabled

Table 15. Triple Speed Ethernet MegaCore Function Parameters (Part 2 of 2)

Parameter	Value
MAC Options	
Enable MAC 10/100 half duplex support	
Enable MII/GMII/RGMII loopback logic	
Enable supplemental MAC unicast addresses	
Implement statistics counters	
Implement multicast hash table	
Enable magic packet detection	
MDIO Module	
Include MDIO module	
Host clock divisor	40
FIFO Options	
Memory block	Auto
Memory width	32 bits
Transmit and receive FIFO depth	1024 × 32 bits

For more details about the Triple Speed Ethernet MegaCore function, refer to the following documents:

- *AN483: Triple Speed Ethernet Data Path Reference Design*
- *Triple Speed Ethernet MegaCore Function User Guide*

Transmit SG-DMA

As part of the Ethernet solution, the Altera transmit SG-DMA controller core facilitates high-speed Ethernet data transfer between the DDR2 SDRAM High Performance MegaCore function and the transmit FIFO in the Triple Speed Ethernet MegaCore function. The processor software creates a set of descriptors to specify the data to transmit and then the SG-DMA performs the series of DMA transfers. Table 16 shows the transmit SG-DMA parameters and corresponding parameter values.

Table 16. Transmit SG-DMA Parameters

Parameter	Value
Name	`sgdma_tx`
Transfer mode	Memory to stream
Data width	32
Source error width	1
FIFO depth	2

For more information about this core, refer to the *Scatter-Gather DMA Controller Core* chapter in volume 5 of the *Quartus II Handbook*.

Receive SG-DMA

The Altera receive SG-DMA controller core facilitates high speed Ethernet data transfer between the Triple Speed Ethernet MegaCore function receive FIFO and DDR2 SDRAM High Performance MegaCore function. The processor software creates a set of descriptors to specify where to receive the data from and then the SG-DMA performs the series of DMA reads specified by the descriptors. Table 17 shows the receive SG-DMA parameters and corresponding parameter values.

Table 17. Receive SG-DMA Parameters

Parameter	Value
Name	`sgdma_rx`
Transfer mode	Stream to memory
Data width	32
Sink error width	6
FIFO depth	2

For more information about this core, refer to the *Scatter-Gather DMA Controller Core* chapter in volume 5 of the *Quartus II Handbook*.

NicheStack TCP/IP Network Stack, Nios II Edition

The NicheStack TCP/IP Network Stack, Nios II Edition is a small code footprint implementation of the transmission control protocol / internet protocol (TCP/IP) suite. Altera provides the NicheStack as a software core that you can add to your system library or board support package.

For more details about the network stack, refer to the *Ethernet and the NicheStack TCP/IP Stack - Nios II Edition* chapter in the *Nios II Software Developer's Handbook*.

RS-232 UART Serial Port

The Altera UART core is used for serial communication via the RS-232 protocol. Table 18 shows the RS-232 UART serial port parameters and corresponding parameter values.

Table 18. RS-232 UART Serial Port Parameters

Parameter	Value
Name	`uart1`
Baud rate	115,200 bps
Parity	None
Data bits	8
Stop bits	1

For more information about this core, refer to the *UART Core* chapter in volume 5 of the *Quartus II Handbook*.

JTAG UART

The Altera JTAG UART core provides serial character stream communication between a PC host and the SOPC Builder system. The JTAG UART core uses the JTAG circuitry built into Altera FPGAs and provides host access via the JTAG pins on the FPGA. The host PC can connect to the FPGA via any Altera JTAG download cable, such as the USB-Blaster cable. You can use the JTAG link to download and run the Nios II software executable file, debug the software program using the Nios II integrated development environment (IDE) debugger, or debug the hardware system using the SignalTap® II Logic Analyzer. Table 19 shows the JTAG UART parameters and corresponding parameter values.

Table 19. JTAG UART Parameters

Parameter	Value
Name	`jtag_uart`
Write FIFO (data from Avalon to JTAG)	
Buffer depth	8 bytes
IRQ threshold	4
Read FIFO (data from JTAG to Avalon)	
Buffer depth	8 bytes
IRQ threshold	4

For more information about this core, refer to the *JTAG UART Core* chapter in volume 5 of the *Quartus II Handbook*.

Video Subsystem

The video subsystem consists of the following parts:

- LCD controller interface—interfaces with the LCD controller IC to configure the LCD panel for brightness, resolution, gamma curves, etc.
- LCD touch panel interface—interfaces with the touch panel controller IC
- Video pipeline—feeds a multiplexed RGB video stream to the LCD controller

LCD Controller Interface

The LCD controller interface is a 3-bit interface constructed with three Altera PIO cores. Table 20 shows the LCD I2C clock PIO parameters and corresponding parameter values.

Table 20. LCD I2C Clock PIO Parameters

Parameter	Value
Name	`lcd_i2c_scl`
Width (1-32 bits)	1
Direction	Output ports only
Output port reset value	0x0

Table 21 shows the LCD I2C data PIO parameters and corresponding parameter values.

Table 21. LCD I2C Data PIO Parameters

Parameter	Value
Name	`lcd_i2c_sdat`
Width (1-32 bits)	1
Direction	Bidirectional (tri-state) ports
Output port reset value	0x0

Table 22 shows the LCD I2C enable PIO parameters and corresponding parameter values.

Table 22. LCD I2C Enable PIO Parameters

Parameter	Value
Name	`lcd_i2c_en`
Width (1-32 bits)	1
Direction	Output ports only
Output port reset value	0x0

For information about these cores, refer to the *PIO Core* chapter in volume 5 of the *Quartus II Handbook*.

LCD Controller Software API

The LCD controller software application program interface (API) provides a high-level initialization function and a set of low-level functions for communicating with the LCD module registers. During normal operation, you only need to communicate with the LCD module during system configuration. This communication occurs through the software API.

For more information about the software API, refer to *AN 527: Implementing an LCD Controller*.

LCD Touch Panel Interface

The LCD touch panel interface is a 4-bit interface consisting of a 3-wire SPI to communicate with the touch panel controller and a PIO core to capture the interrupt generated when the pen moves on the touch panel. Table 23 shows the LCD touch panel SPI parameters and corresponding parameter values.

Table 23. LCD Touch Panel SPI Parameters (Part 1 of 2)

Parameter	Value
Name	`touch_panel_spi`
Master/Slave	Master
Number of select (`SS_n`) signals	1
SPI clock rate (`SCLK`)	32 KHz
Data register width	8 bits

Table 23. LCD Touch Panel SPI Parameters (Part 2 of 2)

Parameter	Value
Shift direction	MSB first
Timing clock polarity	0
Timing clock phase	0

For information about the SPI core, refer to the *SPI Core* chapter in volume 5 of the *Quartus II Handbook*.

Table 24 shows the LCD touch panel pen interrupt PIO parameters and corresponding parameter values.

Table 24. LCD Touch Panel Pen Interrupt PIO Parameters

Parameter	Value
Name	`touch_panel_pen_irq_n`
Width (1–32 bits)	1
Direction	Input ports only
Input options	■ Edge capture register: synchronous capture, falling edge ■ Interrupt generated: edge based

For information about the PIO core, refer to the *PIO Core* chapter in volume 5 of the *Quartus II Handbook*.

LCD Touch Panel Software API

The LCD touch panel software API provides your application with an abstract pen interface consisting of x and y coordinates, and pen state (up or down).

API operations are available to perform the following distinct actions on the touch panel:

- Initialize
- Calibrate
- Operate
- Stop

For more information about the API software, refer to *AN 527: Implementing an LCD Controller*.

Video Pipeline

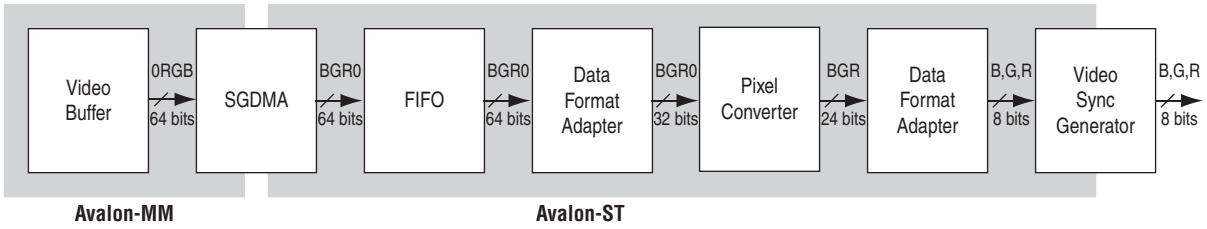
The video pipeline components produce the appropriate pixel data and sync signals to LCD touch panel. The video subsystem consists of the following operational components:

- Video frame buffer—resides in DDR memory.
- Memory-to-stream DMA controller—reads memory 64 bits at a time and produces a stream of 64-bit data values.

- Width (data format adapter)—breaks the 64-bit stream into sequential 32-bit pixel values.
- FIFO—rate matches video stream.
- Pixel format converter—converts the 32-bit BGR0 stream to a 24-bit BGR stream.
- Data format adapter—produces a stream of 8-bit values.
- Video sync-generator—takes the RBG values and produces HSYNC and VSYNC timing signals for the LCD display.

Figure 2 shows the video pipeline.

Figure 2. Video Pipeline



Video Frame Buffer

The image to display resides in a video buffer as 32-bit unpacked pixel data (0RGB). The video buffer is part of the DDR SDRAM memory.

LCD SG-DMA Controller

For minimal interruption during the display of the video stream, the processor sets up an SG-DMA that accesses the pixel data stored in the video buffer on the DDR2 SDRAM and sends the pixel data through the video pipeline. The processor loads up a series of transfers in a descriptor table and the SG-DMA services the descriptor table and thus the stream video. Table 25 shows the LCD SG-DMA controller parameters and corresponding parameter values.

Table 25. LCD SG-DMA Controller Parameters

Parameter	Value
Name	`lcd_sgdma`
Transfer options	Memory to stream
Data width	64 bits
Source error width	0
Data transfer FIFO depth	2

For more information about this core, refer to the *Scatter-Gather DMA Controller Core* chapter in volume 5 of the *Quartus II Handbook*.

LCD Pixel FIFO

The LCD pixel FIFO is a dual clock on-chip FIFO used to buffer video packets in the event that the rate of pixel display is slower that the rate of retrieving video packets from the video buffer. Table 26 shows the LCD pixel FIFO parameters and corresponding parameter values.

Table 26. LCD Pixel FIFO Parameters

Parameter	Value
Name	`lcd_pixel_fifo`
Depth	128
Clock setting	Dual clock mode
Input	Avalon Streaming (Avalon-ST)
Output	Avalon-ST

For more information about this core, refer to the *On-Chip FIFO Memory Core* chapter in volume 5 of the *Quartus II Handbook*.

Pixel Format Converter

The video pipeline outputs a multiplexed stream of R, G, and B channels of 8 bits each. The Altera pixel format converter core simply takes the 32 bits and discards 8 bits to produce the 24 bits that represent the RGB stream. Table 27 shows the pixel format converter parameters and corresponding parameter values.

Table 27. Pixel Format Converter Parameters

Parameter	Value
Name	`lcd_pixel_converter`
Source symbols per beat	3

For more information about this core, refer to the *Video Sync Generator and Pixel Converter Cores* chapter in volume 5 of the *Quartus II Handbook*.

Data Format Adapter

The `lcd_64_to_32_bits_dfa` data format adapter converts 64-bit video frame data to 32-bit data. Table 28 shows the data format adapter (64 to 32) parameters and corresponding parameter values.

Table 28. Data Format Adapter (64 to 32) Parameters (Part 1 of 2)

Parameter	Value
Name	`lcd_64_to_32_bits_dfa`
Input interface	
Data symbols per beat	8
Output interface	
Data symbols per beat	4
Common to Input and Output	

Table 28. Data Format Adapter (64 to 32) Parameters (Part 2 of 2)

Parameter	Value
Channel signal width	0
Max channel	0
Include packet support	Enabled
Error signal width	0
Data bits per symbol	8

The `lcd_32_to_8_bits_dfa` data format adapter converts the 24-bit stream representing RGB to three streams of 8 bit data, with one stream each respectfully representing the R, G, and B pixels. Table 29 shows the data format adapter (32 to 8) parameters and corresponding parameter values.

Table 29. Data Format Adapter (32 to 8) Parameters

Parameter	Value
Name	`lcd_32_to_8_bits_dfa`
Input interface	
Data symbols per beat	3
Output interface	
Data symbols per beat	1
Common to Input and Output	
Channel signal width	0
Max channel	0
Include packet support	Enabled
Error signal width	0
Data bits per symbol	8

For more information about this core, refer to the *Avalon Streaming Interconnect Components* chapter in volume 4 of the *Quartus II Handbook*.

Video Sync Generator

The Altera video sync generator core generates horizontal and vertical synchronization signals to the pixel data and outputs the data to an off-chip display controller. Horizontal synchronization signals synchronize the RGB pixels with the clock to produce a single line of the image. Vertical synchronization signals align a given number of lines to produce individual frames in an image. Table 30 shows the video sync generator parameters and corresponding parameter values. The parameters are user-configurable and define the timing characteristics of the horizontal and vertical synchronization signals.

Table 30. Video Sync Generator Parameters (Part 1 of 2)

Parameter	Value
Name	`lcd_sync_generator`
Data stream bit width	8

Table 30. Video Sync Generator Parameters (Part 2 of 2)

Parameter	Value
Beats per pixel	3
Number of columns	800
Number of rows	480
Horizontal blank pixels	216
Horizontal front porch pixels	40
Horizontal sync pulse pixels	1
Horizontal sync pulse polarity	0
Vertical blank lines	35
Vertical front porch lines	10
Vertical sync pulse lines	1
Vertical sync pulse polarity	0
Total horizontal scan pixels	1,056
Total vertical scan lines	525

Figure 3 shows the horizontal synchronization timing when the parameters are 8 bit data width and 3 beats (R, G, and B) per pixel.

Figure 3. Horizontal Synchronization Timing

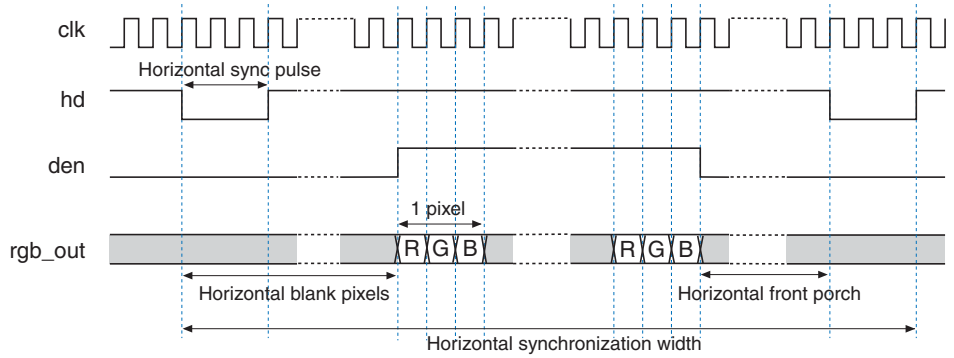
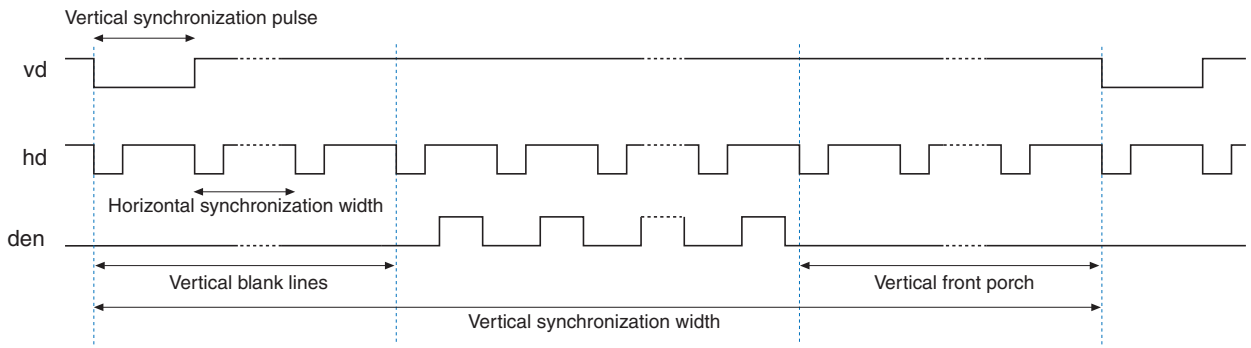


Figure 4 shows the relationship between the horizontal and vertical synchronization signals.

Figure 4. Relationship between Horizontal and Vertical Synchronization Signals



For more information about this core, refer to the *Video Sync Generator and Pixel Converter Cores* chapter in volume 5 of the *Quartus II Handbook*.

Video Pipeline Software API

The video pipeline software API provides a self-contained method to control the video pipeline and manage the graphical frame buffers.

API operations are available to perform the following three distinct actions on the video pipeline:

- Initialize
- Stop
- Manage and manipulate frame buffers

For more information about this software, refer to *AN 527: Implementing an LCD Controller*.

Software Interface

The **system.h** file provides a software description of the microprocessor hardware. The **system.h** file describes each peripheral in the system and provides the following details:

- The hardware configuration of the peripheral
- The base address and offset
- The IRQ priority (if any)
- Accessibility of slave by different masters
- A symbolic name for the peripheral

You can include the **system.h** file in your software application source code to interface with the hardware system.

For more information about **system.h** and the hardware abstraction layer (HAL), refer to the *Hardware Abstraction Layer* section of the *Nios II Software Developer's Handbook*.

Hardware Interface

SDC Timing Constraints

The following Synopsys Design Constraint (**.sdc**) files reside in the project directory:

- **cycloneIII_3c120_niosII_video.sdc**
- **ddr2_sdram_phy_ddr_timing.sdc**
- **ddr2_sdram_1_phy_ddr_timing.sdc**

To add an **.sdc** file to the system, add the following constraint to your Quartus II Settings File (**.qsf**):

```
set_global_assignment -name SDC_FILE <SDC filename>.sdc
```

For further details on your Quartus II settings, refer to the **.qsf** file in the project directory.

Customization

Changing Device and Pin Assignments

This LCD system design is targeted to a Cyclone III 3C120 FPGA. You can target your design to other FPGAs by specifying options on the **Device Settings** page in the Quartus II software.

For more information about specifying a device family, refer to Quartus II Help.

The pin assignments for the microprocessor have to be re-assigned after changing the device. Use the Quartus II software's Assignment Editor to assign the pins according to the printed circuit board (PCB) schematic design for the FPGA and other devices on your board.

For more information about using the Assignment Editor, refer to the *Assignment Editor* chapter in volume 2 of the *Quartus II Handbook*.

Configuring the Processor

The Nios II processor is configurable, soft-core processor. Use the Nios II processor MegaWizard interface in SOPC Builder to reconfigure the Nios II processor settings for the existing LCD system design.

For more information about the Nios II processor MegaWizard interface and settings, refer to the *Instantiating the Nios II Processor in SOPC Builder* chapter in the *Nios II Processor Reference Handbook.*

Adding IP cores

You can add new IP cores into the existing LCD system design in the SOPC Builder.

For more information about adding IP cores in SOPC Builder, refer to *Volume 4: SOPC Builder* of the *Quartus II Handbook*.

Adding Custom Components

You can create your own custom components and add them to the LCD system design. The SOPC Builder's Component Editor helps integrate your custom components into a system.

For more information about the Component Editor, refer to the *Component Editor* chapter in volume 4 of the *Quartus II Handbook.*

You can develop custom device drivers for the custom components added in the system.

For more information about developing device driver, refer to the *Developing Device Drivers for the Hardware Abstraction Layer* chapter in *Nios II Software Developer's Handbook*.

Scaling Performance

There are several ways to improve the system performances. Documents in the following list describe improvement methods:

- To add custom instruction to the Nios II processor, refer to the *Nios II Custom Instruction User Guide*.
- To use C-to-hardware (C2H) acceleration for time critical code, refer to the following documents:
- To create a multiprocessor system, refer to *Creating Multiprocessor Nios II Systems Tutorial*.
- *Accelerating Nios II Systems with the C2H Compiler Tutorial*
- *Nios II C2H Compiler User Guide*
- To use tightly coupled memory with the processor, refer to *Using Tightly Coupled Memory with the Nios II Processor Tutorial*.
- To incorporate system level design optimization techniques, refer to *System-Level Design* section of the *Embedded Design Handbook*.
- To accelerate Nios II networking applications, refer to *AN 440: Accelerating Nios II Networking Applications*.

Referenced Documents

This data sheet references the following documents:

- *Accelerating Nios II Systems with the C2H Compiler Tutorial*
- *altpll Megafunction User Guide*
- *AN 440: Accelerating Nios II Networking Applications*
- *AN 483: Triple Speed Ethernet Data Path Reference Design*
- *AN 527: Implementing an LCD Controller*
- *Assignment Editor* chapter in volume 2 of the *Quartus II Handbook*
- *Avalon Streaming Interconnect Components* chapter in volume 4 of the *Quartus II Handbook*
- *Common Flash Interface Controller Core* chapter in volume 5 of the *Quartus II Handbook*
- *Component Editor* chapter in volume 4 of the *Quartus II Handbook*
- *Configuring Cyclone III Devices* chapter in volume 1 of the *Cyclone III Device Handbook*
- *Creating Multiprocessor Nios II Systems Tutorial*
- *Cyclone III Device Handbook*

- *DC and Switching Characteristics* chapter in volume 2 of the *Cyclone III Device Handbook*
- *DDR and DDR2 SDRAM High-Performance Controller User Guide*
- *Developing Device Drivers for the Hardware Abstraction Layer* chapter in *Nios II Software Developer's Handbook*
- *Ethernet and the NicheStack TCP/IP Stack - Nios II Edition* chapter in the *Nios II Software Developer's Handbook*
- *Hardware Abstraction Layer* section of the *Nios II Software Developer's Handbook*
- *Instantiating the Nios II Processor in SOPC Builder* chapter in the *Nios II Processor Reference Handbook*
- *JTAG UART Core* chapter in volume 5 of the *Quartus II Handbook*
- *Nios II C2H Compiler User Guide*
- *Nios II Custom Instruction User Guide*
- *On-Chip FIFO Memory Core* chapter in volume 5 of the *Quartus II Handbook*
- *Performance Counter Core* chapter in volume 5 of the *Quartus II Handbook*
- *Pin Information for the Cyclone III EP3C120 Device*
- *PIO Core* chapter in volume 5 of the *Quartus II Handbook*
- *Remote System Upgrade With Cyclone III Devices* chapter in volume 1 of the *Cyclone III Device Handbook*
- *Scatter-Gather DMA Controller Core* chapter in volume 5 of the *Quartus II Handbook*
- *SPI Core* chapter in volume 5 of the *Quartus II Handbook*
- *System ID Core* chapter in volume 5 of the *Quartus II Handbook*
- *System-Level Design* section of the *Embedded Design Handbook*
- *Timer Core* chapter in volume 5 of the *Quartus II Handbook*
- *Triple Speed Ethernet MegaCore Function User Guide*
- *UART Core* chapter in volume 5 of the *Quartus II Handbook*
- *Using Tightly Coupled Memory with the Nios II Processor Tutorial*
- *Video Sync Generator and Pixel Converter Cores* chapter in volume 5 of the *Quartus II Handbook*
- *Volume 4: SOPC Builder* of the *Quartus II Handbook*

Revision History

Table 31 shows the revision history for this data sheet.

Table 31. Revision History

Date and Revision	Changes Made	Summary of Changes
November 2008 v1.0	Initial Release.	—