SoC FPGA Dedicated Peripherals Advance Information Brief

This document describes the dedicated peripherals that support Altera® SoC FPGAs, featuring a microprocessor unit (MPU) with a dual-core ARM® Cortex™-A9 MPCore™ processor. Altera's 28-nm Cyclone® V and Arria® V SoC FPGAs include a rich set of system peripherals to help you create SoC FPGA designs rapidly.

Peripheral System Overview

The SoC FPGA hard processor system (HPS) includes the MPU subsystem and the following dedicated system peripherals:

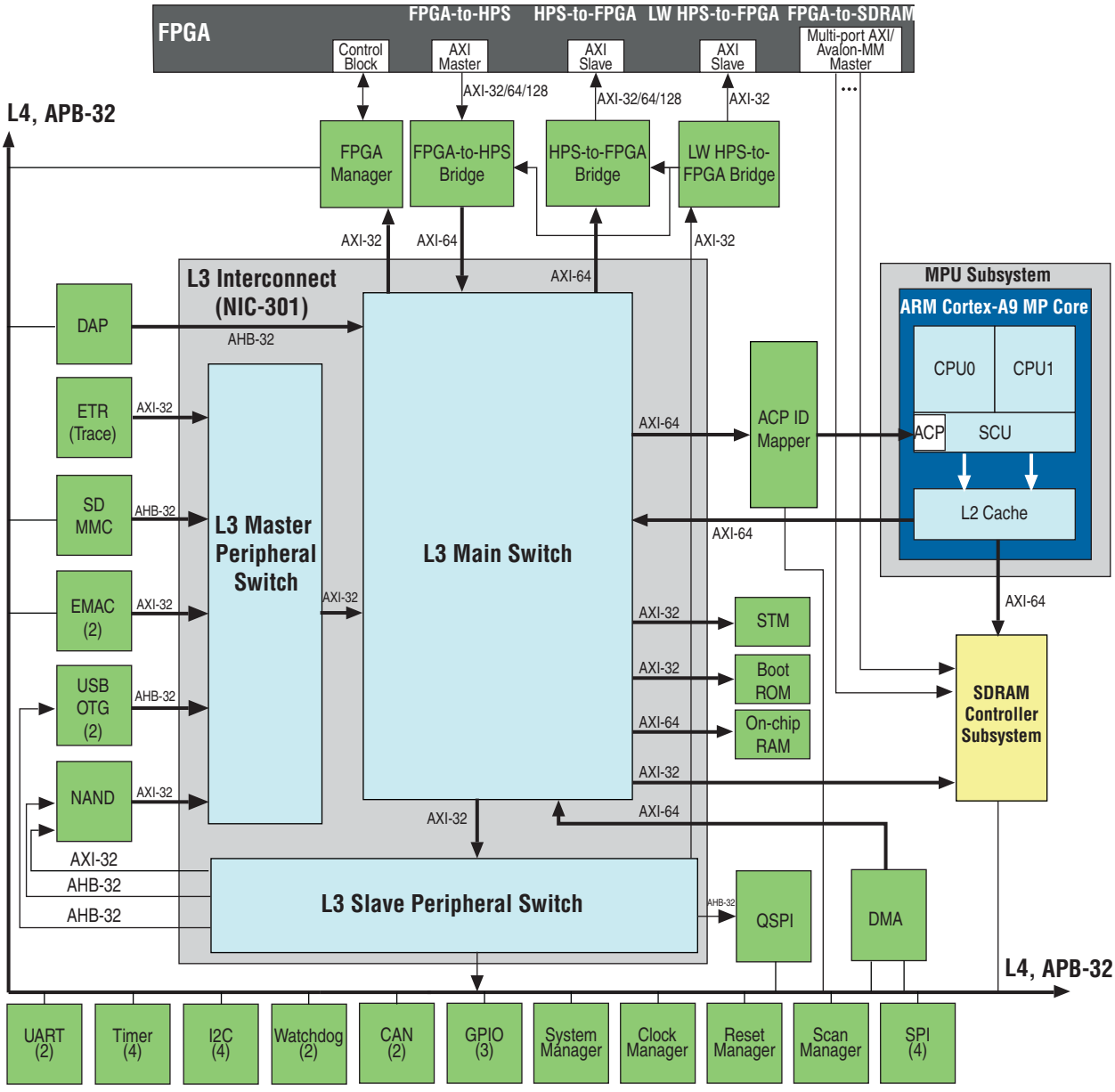
- Two 10/100/1000 Ethernet media access controllers (EMACs)
- Two USB 2.0 On-The-Go (OTG) controllers with direct memory access (DMA)
- Two controller area network (CAN) controllers
- Multiport SDRAM controller subsystem
  - DDR2/3
  - LPDDR1/2
- Micro-programmable DMA controller with instruction set
- NAND flash controller with DMA and optional error correction code (ECC)
- Quad serial peripheral interface (SPI) NOR flash controller with optional ECC
- SD/SDIO/MMC flash controller with integrated DMA and optional ECC
- Two 16550-compliant UARTs
- Four 32-bit general-purpose timers
- Two 32-bit watchdog timers
- Four I^2C serial ports
- Two serial peripheral interface (SPI) masters and two SPI slaves
- Three general-purpose I/O (GPIO) ports, each with 29 GPIO pins per port; 86 GPIO pins total
- System manager
- Scan manager
- 64 kilobytes (KB) of on-chip RAM

The following sections describe the SoC FPGA dedicated peripherals in greater detail.

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Figure 1 shows an HPS block diagram including the MPU and dedicated system peripherals.

Figure 1. SoC FPGA Dedicated Peripherals and Debug Modules



10/100/1000 EMACs

The HPS peripheral set includes two Ethernet media access controllers (EMACs), each supporting 10/100/1000 megabits per second (Mbps) connections over a standard Ethernet link.

EMAC Features

10/100/1000 Mbps MAC

- IEEE 802.3x flow control in full-duplex mode
- Back-pressure support for half-duplex mode
- IEEE 1588-2002 and 2008 precision networked clock synchronization
- IEEE 802.3-az, version D2.0 for Energy Efficient Ethernet (EEE)
- Preamble and start-of-frame data (SFD) insertion in transmit and deletion in receive paths
- Automatic cyclic redundancy check (CRC) and pad generation controllable on a per-frame basis
- Options for automatic pad/CRC stripping on receive frames
- Programmable frame length supporting standard and jumbo Ethernet frames (with sizes up to 9.6 KB)

Physical Interface

- An external Ethernet physical layer (PHY) interface device is required for industry-standard network interfaces
- Dedicated hardware support for reduced pin-count PHY interfaces
  - Reduced Media Independent Interface (RMII) for 10/100
  - Reduced Gigabit Independent Interface (RGMII) for 10/100/1000
- FPGA-based support for alternate PHY interfaces
  - Gigabit Media Independent Interface (GMII)
  - Serial Gigabit Media Independent Interface (SGMII)
- Management data input/output (MDIO) or I^2C PHY management interface

Integrated DMA Support

- Integrated, dedicated DMA engine to offload the host processor
- Programmable burst size for optimal bus utilization
- Single-channel mode transmit and receive engines
- Byte-aligned addressing mode for data buffer support
- Dual-buffer (ring) or linked-list (chained) descriptor chaining
- Descriptors can each transfer up to 8 KB of data

Management Interface

- Separate MDIO ports, one per EMAC
- Comprehensive status reporting for normal operation and transfers with errors
- Configurable interrupt options for different operational conditions
- Per-frame transmit/receive complete interrupt control
- Separate status returned for transmission and reception packets

Hardware Acceleration

- Offload transmit and receive checksum generation for TCP, user datagram protocol (UDP), or Internet control message protocol (ICMP) over IP

Other Features

- Supports flexible address filtering modes
- Up to 31 additional 48-bit perfect destination address filters with masks for each byte
- Up to 31 48-bit source address comparison checks with masks for each byte
- 256-bit hash filter (optional) for multicast and uni-cast destination addresses
- Option to pass all multicast addressed frames
- Promiscuous mode support to passing all frames without any filtering for network monitoring
- Passes all incoming packets (as per filter) with a status report

External PHY Support

An Ethernet PHY integrates all the specialized physical-layer functions required to send and receive Ethernet-compliant data. Because of the special analog circuitry and electrostatic discharge (ESD) protection required, the Ethernet PHY is implemented as an external device to the FPGA SoC.

Each HPS EMAC has direct interfaces that support the following PHY interfaces. These "reduced" interfaces use fewer dedicated pins.

- RMII for 10/100 Mbps
- RGMII for 10/100/1000 Mbps

The following additional PHY interfaces require connections using FPGA fabric resources:

- GMII
- SGMII

The GMII interface is exposed to the FPGA fabric. You can the implement and interface from FPGA fabric logic to a GMII or SGMII PHY. Alternatively, the GMII interface can connect to an external PHY through FPGA I/O pins.

The SGMII interface is a serial 1.25 Gbps bidirectional link to a supporting PHY. The IEEE 802.3z specification describes the physical coding sublayer (PCS) portion of the interface in detail. The interface is source synchronous and is clocked at 625 MHz DDR running over four differential pairs (clock and data for each direction). Interface signals are routed through the FPGA fabric and to a hard transceiver block. FPGA logic implements PCS and clock data recovery (CDR) logic while the transceivers provide device I/O.

The EMAC also includes a control interface to configure and monitor the status of the PHY. There are two choices of control interface:

- Station management interface multifunction interface module (MIM)
- I^2C interface

The MIM interface is part of the EMAC peripheral, while the I^2C interface uses one of the separate HPS I^2C modules. The interfaces are multiplexed externally to the EMAC.

IEEE 1588 Precision Time Protocol Support

The IEEE 1588-2002 standard defines the Precision Time Protocol (PTP), enabling precise clock synchronization of distributed devices over a network. The PTP applies to systems communicating by local area networks that support multicast messaging. This protocol enables heterogeneous systems to synchronize with one another, despite having clocks of varying inherent precision, resolution, and stability. PTP is used in automation systems where a collection of communicating machines such as robots must be synchronized to a common time base. Clocks at individual nodes in the network tend to drift over time. When sensors, actuators and other devices lose synchronization, the functionality of the entire distributed system can be compromised.

The IEEE 1588 protocol forms the basis of almost all real-time automation protocols. It is a message-oriented protocol that operates over Ethernet and other types of network links such as a CAN bus. The PTP protocol exchanges UDP/IP messages between various nodes of the system. The system uses these messages and network latency to resolve differences between the master clock and slave clocks.

In general, PTP easily achieves sub-microsecond synchronization accuracy. In the HPS implementation of PTP, a 50 MHz reference clock produces 20-ns accuracy, updating the system time in 20-ns steps.

The HPS EMAC supports additional features found in the latest version of the specification, IEEE 1588-2008. Peer-to-peer (PTP) transparent clock (TC) messages are supported.

IEEE 802.3az Energy Efficient Ethernet

Each EMAC supports Energy Efficient Ethernet (EEE), as defined in the IEEE 802.3-az standard, version D2.0. The EEE features are available when the EMAC operates in full-duplex mode, using either the GMII or RGMII PHY interfaces. EEE is not supported in half-duplex mode, or when using the RMII or SGMII PHY interfaces.

EEE enables the EMAC to operate in low-power idle (LPI) mode. Either end point of an Ethernet link can disable functionality to save power during periods of low link utilization. The MAC controls whether the system should enter or exit LPI mode and communicates this information to the PHY. The EEE standard specifies the negotiation methods used by the link partners to determine whether EEE is supported for the specific link and, if so, selects the set of parameters common to the devices and both ends of the link.

Remote FPGA Update

Remote FPGA update is possible via Ethernet and other sources. Because the processor can configure and reconfigure the FPGA under program control, the processor can read FPGA configuration files from any of the HPS peripheral interfaces (e.g. Ethernet, USB, SPI, SD/MMC). Additionally, the processor can write the new FPGA image to the configuration flash memory to install a nonvolatile update.

Software Download and Debug via Ethernet

Software developers can debug their application software and download new code using one of the Ethernet ports—a common software development flow for applications running under operating systems (OS) such as Linux and VxWorks®. However, non-OS or *bare metal* applications require additional code, including a debug stub and Ethernet stack, functions typically included as part of the OS.

USB Controllers

The HPS peripheral set includes two independent USB On-The-Go (OTG) controllers. Each controller is a dual-role device (DRD) that supports both USB host and USB device functions, including a range of data rates from low-speed (1.5 Mbps) up to high-speed (480 Mbps), as shown in Table 1. The controller is fully compliant with the OTG supplement to the USB 2.0 specification, and can be programmed to support both device and host functions to enable the data movement over the USB protocol.

USB OTG 2.0 Controller Features

- Complies with USB 2.0 specification, including both revision 1.3 and revision 2.0 of the OTG supplement
- Supports all USB 2.0 speeds:
  - High-speed (HS, 480-Mbps)
  - Full-speed (FS, 12-Mbps)
  - Low-speed (LS, 1.5-Mbps), supported only in host mode
- Supports all types of USB transfers
  - Control
  - Bulk
  - Isochronous
  - Interrupt
  - Automatic ping capability
- Supports software-configurable modes of operation between OTG1.3 and OTG2.0.
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- Supports suspend, resume, and remote wake
- Independent DMA support for all 16 channels
- Supports up to 16 host channels
  - Up to 4,064 endpoints supported by reprogramming each host channel under software control
  - Each channel capable of supporting up to 127 devices, each having 32 endpoints (IN + OUT)
- Supports up to 16 bidirectional endpoints, including control endpoint 0
- Seven periodic device IN endpoints supported
- Generic root hub support
- Transaction scheduling performed in hardware
- External USB PHY support features
  - Single USB port connected to each OTG instance
  - Supports a ULPI interface connection to an off-chip USB transceiver with support for 12-bit SDR, 8-bit DDR and carkit
  - Software-controlled access, supporting vendor specific or optional PHY registers access to ease debug
  - OTG2.0's support for Attach Detection Protocol (ADP) only through an external ADP controller

Table 1. Supported USB Operating Modes and Data Rates

Operating Mode/Data Rate	USB Host	USB Device (Slave)
High-speed (HS, 480 Mbps)	✓	✓
Full-speed (FS, 12 Mbps)	✓	✓
Low-speed (LS, 1.5 Mbps)	✓	—

Each USB OTG controller supports connections for all types of USB devices, including mouse, keyboard, digital camera, network adapters, hard drives, and a generic hub. As a DRD, the USB OTG controller fulfills multiple roles. The USB OTG module can perform as a USB slave device. Example USB slave devices include a digital audio player or a mobile phone. The USB OTG module can also perform as a USB host, similar to the role played by the USB port on desktop or notebook computer.

Each USB OTG controller supports a single USB port connected via an external PHY device. Because of the voltages and ESD protection required, the USB PHY is implemented as an external device, connected to the SoC FPGA with a USB 2.0 Transceiver Macrocell Interface (UTMI) Low Pin Interface (ULPI)-compliant USB transceiver using either 12-bit SDR, 8-bit DDR, or three-wire carkit operation.

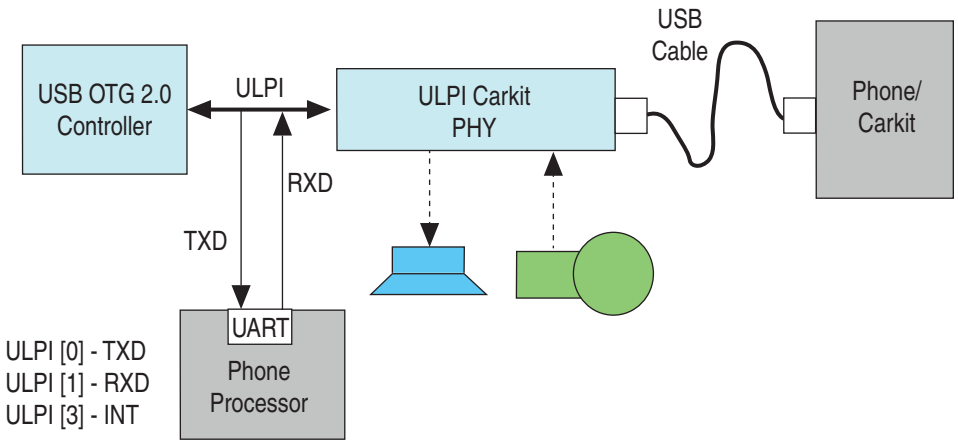
The following USB PHYs are compatible with the USB OTG controller:

- Texas Instruments TUSB1210
- NXP ISP1504
- Cypress CY7C68003
- SMSC USB3300

Carkit Support

The carkit (CEA-936-A) option allows the link to communicate through the PHY to a remote device, such as mobile phone or other RS-232 devices, through UART-style signaling over a mini-USB connector, as shown in Figure 2

Figure 2. ULPI Carkit and USB/Carkit PHY



CAN Bus Controller

The HPS peripheral set includes two CAN interface controllers to serially communicate with other CAN controllers using the CAN protocol. The CAN two-wire serial interface uses a multimaster communication protocol that efficiently supports distributed real-time applications. The HPS CAN controllers are created from IP supplied directly by Bosch®. These modules are fully compliant with CAN protocol specification 2.0 parts A and B, and support bit rates up to 1 Mbps. External CAN transceivers are required to establish the CAN connection.

The HPS CAN controller handles all the ISO 11898-1 CAN protocol functions. A message handler controls data transfers between the message RAM and the CAN module's transmit and receive shift register. The message handler also handles acceptance filtering and the interrupt.

CAN Controller Features

The HPS CAN controller supports the following features:

- CAN protocol version 2.0 parts A and B
- Bit rates up to 1 Mbps
- Supports 128 message objects
  - Each message object has its own identifier mask
  - Programmable FIFO mode for message objects
- Programmable loop-back modes for self-test operation

- Two interrupt lines per CAN module
  - Status interrupt
  - Message object interrupt
- Optional automatic retransmission (ISO11898-1, 6.3.3)

Interoperability

Table 2 lists a sampling of the CAN transceivers that are compatible with the HPS CAN controller. Each transceiver requires only two connections to the HPS CAN peripheral—serial transmit and receive data using 3.3V CMOS signal levels.

Table 2. CAN Transceivers Compatible with HPS CAN Controller

Vendor	Part Number
Infineon	TLE6250GV33
Maxim	MAX13041
NXP	TJA1041A TJA1043
ON Semi	NCV7341
Texas Instruments	SN65HVD230 SN65HVD231 SN65HVD232 SN65HVD233 SN65HVD234 SN65HVD235

DMA Controller

The HPS DMA controller is micro-programmable with its own instruction set, and provides a rich set of features and flexible DMA transfer types. This DMA is used to accesses the following modules:

- Memory to memory
- Quad SPI
- SPI
- I^2C
- UART
- CAN
- System Trace Macrocell (STM)
- FPGA fabric

The DMA controller is separate and independent of the DMA controllers integrated with other HPS peripherals, including the EMACs, USB controllers, and the flash memory controllers (SD/SDIO/MMC and NAND). However, modules such as the UARTs and quad SPI rely on this DMA because they lack an integrated DMA. The DMA's program code is stored in system memory, but it includes an efficient instruction caching system to optimize performance and minimize bus traffic.

The DMA controller supports up to eight logical channels with up to 32 handshake peripheral requests. The peripheral request interface includes hardware flow control to prevent slow peripherals from degrading higher-bandwidth transfers.

DMA Controller Features

The HPS DMA controller supports the following features:

- Micro-programmable transfer operations
  - Small instruction set provides a flexible method to specify DMA operations
  - Greater flexibility than the fixed capabilities of a linked-list DMA controller
- Multiple transfer types
  - Memory-to-memory
  - Memory-to-peripheral
  - Peripheral-to-memory
  - Scatter-gather
- Eight DMA channels
- Up to eight outstanding bus read and eight outstanding bus write transactions
- Scheduling capabilities of up to 16 outstanding read and 16 outstanding write instructions
- Eleven interrupt lines
  - Eight lines for external events (one per DMA channel)
  - One line for DMA thread abort
  - Two lines for MFIFO ECC, single-bit corrected and double-bit corrected
- Up to 32 peripheral request interfaces
- Secure and non-secure transfers

GPIO Ports

The HPS peripheral set includes GPIO ports. Each port consists of 29 GPIO pins per port, or 86 GPIO pins in total. Of these 86 pins, 15 are for input only. The GPIO pins include synchronous debounce circuitry and are capable of generating an interrupt to wake up the system.

Serial Communication Interfaces

The dedicated HPS peripheral set supports the following popular serial communication interfaces:

- I^2C, including Two-Wire Interface (TWI)
- SPI
- RS-232 UART

I^2C Bus Controller

The HPS provides four I^2C controllers, enabling the system to serially communicate with other external I^2C-compatible devices. The I^2C controllers operate in either master or slave mode, and support both 100 kilobits per second (Kbps) and 400 Kbps transfer rates. You can optionally use two of the I^2C controllers to configure Ethernet PHYs connect to the EMACs.

In order to support the small formfactor plugable (SFP) Ethernet transceiver, the HPS provides two additional I^2C controllers to emulate the two-wire serial interface defined for the ATMEL AT24C01A/02/04 family. The maximum transfer rate of the two-wire interface is 100 kbps.

The I^2C bus controller supports the following features:

- Two communication speeds
  - Standard mode (100 Kbps)
  - Fast mode (400 Kbps)
- Master or slave I^2C operation
- Optional support for TWI, which is similar but not fully compliant with I^2C
- 7- or 10-bit addressing
- Mixed read and write combined format transactions in both 7-bit and 10-bit
- Bulk transmit mode
- Handles bit and byte waiting at all bus speeds
- Data FIFOs with programmable fill-level for interrupts and DMA control
- DMA handshaking interface

SPI Bus Controller

The HPS peripheral set provides two general-purpose SPI controllers, each capable of operating in either master or slave mode. These SPI controllers are in addition to the "Quad SPI NOR Flash Controller" on page 12. The SPI controller is compliant to the Motorola SPI four-wire, full-duplex, serial protocol.

The HPS SPI interface supports data rates up to 60 Mbps when operating in master mode, and 50 Mbps when operating in slave mode.

Frame data is stored in an associated FIFO, which supports up to 256 continuous frames of between 4 to16 bits each. The SPI controllers have the following features:

- High-speed SPI master operation
  - Programmable serial receive data sample time
  - Enables programmable control of routing delays resulting in higher serial data-bit rates
  - Dynamic control of serial clock bit-rate
- Selectable data item size
  - From 4 to 16 bits per item
  - Size of each data transfer under program control
- Supports up to four slave select outputs when operating as a serial master
- Programmable interrupt or DMA threshold level for transmit and receive FIFOs
- SPI baud rate generator
- Each SPI master contains four slave selects, enabling connection with 4 slaves

UART Controllers

The HPS peripherals includes two UART controllers to provide asynchronous serial communication. The UART controller is modeled after the industry-standard 16550 UART, but with control registers aligned to 32-bit boundaries. The UART controller supports 16750 compatible auto-flow control (RTS and CTS cross-connected). The UART controllers have the following features:

- Programmable character transfer properties
  - Number of data bits per character (5 to 8 bits per character)
  - Optional parity bit with odd or even parity select
  - Number of stop bits (1, 1.5, or 2)
- Line-break generation and detection
- DMA request signaling with two programmable modes (FIFO, non-FIFO)
- Prioritized interrupt identification
- Programmable baud rate generator. Using a 100 MHz reference clock, generate baud rates from 95 baud to 6.25 Mbaud
- False start bit detection
- Auto-flow control mode, as specified in the 16750 standard. RTS and CTS signals connected to FPGA fabric
- Internal loopback mode that enables greater testing of modem control and auto-flow control features

The UART does not include IrDA mode.

Flash Memory Interfaces

The HPS peripheral set includes dedicated controllers for the following flash memory types:

- Quad SPI NOR flash memory
- 8-bit NAND flash memory
- SD/SDIO/MMC flash memory cards

These flash memory interfaces are specifically optimized for loading the processor's secondary boot image, usually compressed, and the configuration image(s) for the FPGA. The interfaces also use a minimum number of I/O pins. The quad SPI and NAND flash interfaces support components usually soldered directly on the board. The SD/SDIO/MMC interface supports removable flash media.

Although designed primarily for boot and configuration, the flash interfaces are also available to the FPGA SoC application after boot and configuration. For example, you can use one flash interface to boot and configure the FPGA SoC, while the other flash interfaces are available for bulk data and file storage.

There is no dedicated support within the HPS peripheral set for parallel NOR flash because of the large number of required interface pins. However, you can add a parallel NOR flash interface in the FPGA fabric using the soft logic.

Quad SPI NOR Flash Controller

The HPS peripheral set includes a quad SPI NOR flash controller, designed to rapidly read data stored in SPI flash PROMs over one (x1), two (x2), or four (x4) data lines. The more data lines, the higher the effective read and write bandwidth. The quad SPI NOR flash controller is separate and independent from the "SPI Bus Controller" on page 10.

Quad SPI NOR Flash Controller Features

- Support for x1 (single), x2 (dual), and x4 (quad) SPI flash instructions
- Programmable device sizes
- Programmable clock frequency up to 108 MHz
- Programmable clock polarity
- Up to four external device selects
- Support for execute in place (XIP)
- Memory mapped "direct" mode of operation for performing flash data transfers
- Programmable interrupt generation
- Software triggered "indirect" mode of operation for performing low latency and non processor intensive flash data transfers
- DMA peripheral interface to communicate indirect mode status with external DMA
- Programmable write protected regions
- Programmable delays between transactions
- Legacy mode allows software to directly access the low-level transmit and receive FIFOs

Quad SPI flash memory often provides similar or faster bandwidth than an equivalent-sized parallel NOR flash and also requires far fewer I/O pins (6 pins for quad SPI flash versus 40 or more pins for parallel NOR flash).

The quad SPI controller supports the following SPI flash PROM devices:

- Micron N25Q series
- Numonyx N25Q series
- Spansion S19 series

- Atmel AT25 series

NAND Parallel Flash Controller

The HPS peripheral set includes a NAND parallel flash controller. NAND flash has the advantage of providing higher storage capacity at low cost-per-bit and fast, sustained write performance, making NAND flash ideal for file applications. NAND flash also uses fewer pins than an equivalent parallel NOR-flash interface, primarily because of its multiplexed address/data interface.

NAND Parallel Flash Controller Features

- Supports x8 NAND flash devices available from multiple memory manufacturers
  - Single-layer cell (SLC)
  - Multiple-layer cell (MLC)
- Programmable page sizes
  - 512 bytes per page
  - 2 KB per page
  - 4 KB per page
  - 8 KB per page
- Programmable block size
  - 32 pages per block
  - 64 pages per block
  - 128 pages per block
- Maskable, active-high interrupt
- Programmable access timing
- Supports up to four banks
- Optional ECC with programmable sector size
  - Bose–Chaudhuri–Hocquenghem (BCH) algorithm to detect and correct multiple errors per page
  - 512-byte sector with 4-, 8-, or 16-bit corrections
  - 1024-byte sector with 24-bit corrections
- Support for Open NAND Flash Interface (ONFI) 1.0, Samsung K9 series, and Toshiba TC58 series NAND flash devices
- Pipelined read-ahead and write commands for enhanced read and write throughput
- 128-word-deep read and write FIFOs
- Integrated, dedicated DMA engine to offload the host processor

The NAND flash controller supports x8 NAND flash memories from the following manufacturers:

- Hynix
- Samsung
- Toshiba
- Micron Technology, Inc.
- STMicroelectronics

SD/SDIO/MMC Flash Controller

The HPS peripheral set includes an SD/SDIO/MMC flash controller. The HPS SD/SDIO/MMC flash controller supports removable flash cards. The other supported flash interfaces (SPI flash, NAND flash) are primarily designed for embedded applications, where the flash components are either soldered on the circuit board or available as package-on-package components.

SD/SDIO/MMC Flash Controller Features

- Single-card flash memory interface for a variety of flash card standards
- Secure Digital memory (SD – version 3.0)
- Secure Digital I/O (SDIO - version 3.0, includes SDIO and eSDIO)
- Consumer Electronics Advanced Transport Architecture (CE-ATA – version 1.1)
- MultimediaCard (MMC - version 4.41, x1 and x4 cards only, x8 cards not supported)
- Integrated, dedicated DMA engine to offload the host processor
- Internal 4KB data FIFO to store transmit and receive data

Interoperability Support

- SD
  - SanDisk 64MB SD, 256MB SD, 256MB MiniSD
  - Panasonic 128MB SD
  - PNY 256MB SD
  - Memorex 32MB SD
  - SimpleTech 64MB SD

Interoperability Support (cont.)

- MMC
  - SanDisk 64MB MMC
  - SimpleTech 128MB MMC
- Lexar 32MB MMC
- SDIO
  (Verified only Basic CMD5, CMD52, and CMD53 I/O commands)
  - PALM Bluetooth
  - Toshiba Bluetooth
- HSMMC
  (Verified 1-bit, 4-bit and 8-bit modes; verified CMD6, CMD8, CMD14, CMD19, CMD17, CMD18, CMD24, and CMD25 commands)
  - Pretec
  - Skymedi
- CEATA
  - Hitachi Microdrive 3K8 hard drive

The MultiMediaCard (MMC) is a flash memory card standard. MMC originally used a 1-bit serial interface, but newer versions allow data transfers of 4 or 8 bits at a time. MMC cards have essentially been superseded by Secure Digital (SD) cards, although MMC cards are still used in many applications that primarily support SD cards. Secure Digital Input Output (SDIO) cards are a similar technology—a combination of an SD card and an I/O device. The HPS interface to any of these SD/SDIO/MMC memories requires just 7-9 I/O pins.

The SD/SDIO/MMC flash controller enables the FPGA SoC to boot and configure directly from a removable flash memory card, which is convenient during the system development phase. This method also provides a practical means to upgrade system software and FPGA configuration files for non-networked applications. Likewise, flash memory cards provide sufficient, affordable bulk storage for larger applications or user data.

On-Chip RAM

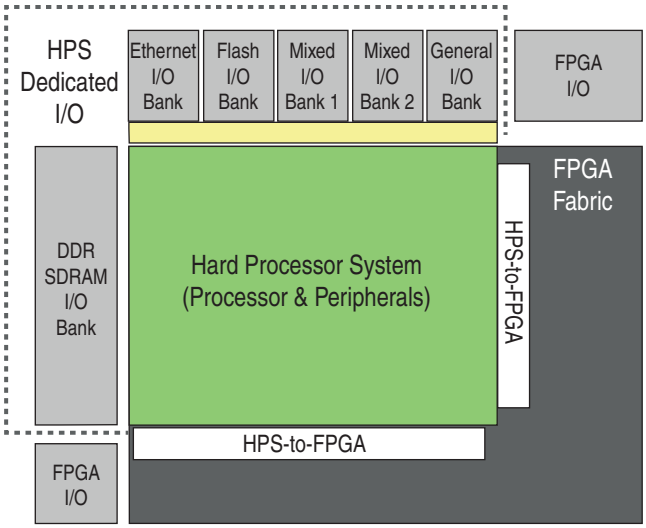
The HPS peripheral set includes 64 KB on-chip RAM. The 64 KB on-chip RAM is used during the initial boot process but can be re-used by the application program once the boot process is complete. The on-chip RAM has various potential uses, including but not limited to the following uses:

- Stores the “warm boot” code to accelerate processor boot on a warm reset
- Stores the DMA controller program
- Buffers data for Ethernet, USB, or FPGA logic

HPS Dedicated I/O Pins

The HPS, including the processor and peripherals, connects to external devices through a number of dedicated I/O pins, arranged into multiple I/O banks, as shown in Figure 3.

Figure 3. HPS Dedicated I/O



The actual number of banks and number of dedicated pins depends on the specific SoC FPGA device and packages.

Supported I/O Standards

The dedicated HPS I/O banks support the following variety of I/O standards:

- 3.3V LVTTL/LVCMOS
- 3.0V LVCMOS
- 2.5V LVCMOS
- 1.8V LVCMOS
- 1.5V LVCMOS
- 1.5V HSTL, Class I and II

The dedicated SDRAM controller subsystem pins support different standards, as described in "SDRAM Controller Subsystem" on page 18.

Pin Multiplexing

As with many highly-integrated, processor-based SoC devices, not every signal from every peripheral can simultaneously connect to external devices, especially in all packages. The interface pins to and from the dedicated peripherals are connected through pin multiplexing, similar to Table 3.

A single HPS pin connects to as many as four different peripherals. In typical usage, each pin connects to only one source. For each set of pins associated HPS I/O bank, the application selects from one of up to five possible options in the associated table row, as indicated in Table 3. Selecting a specific option precludes the other possible options available in that row. For example, from the top row in Table 3, if some of the EMAC I/O pins connect to `RGMII 0`, then those same pins cannot be used for the `USB 1`, for `GPIO[13:0]`.

You can use the Qsys system integration tool available in the Quartus II software to confirm that the configuration of peripherals is valid. The actual pin mapping is controlled by registers in the system manager.

The DDR SDRAM interface and the dedicated pins are not multiplexed.

Table 3. Preliminary HPS I/O Pin Multiplexing

HPS I/O Bank	From Dedicated Peripherals			
	Option 1	**Option 2**	**Option 3**	**Option 4**
Ethernet MAC I/O	RGMII 0	USB 1	—	GPIO[13:0]
	RGMII 1	—	—	GPIO[19:14]
Mixed I/O 2		SPI Master 0	SPI Slave 0	GPIO[54:51]
		SPI Master 1	SPI Slave 1	GPIO[58:55]
Mixed I/O 1	NAND flash	RGMII 1	USB 1	GPIO[28:14]
Flash	Quad SPI flash	—	—	GPIO[35:29]
	USB 0	SD/SDIO/MMC	—	GPIO[47:36]
	SD/SDIO/MMC	Quad SPI flash (extra selects)	—	GPIO[58:48]
General	Trace	SPI Slave 0	SD/SDIO/MMC	GPIO[55:51]
		SPI Master 1	SPI Slave 1	GPIO[59:56]
	SPI Master 0	I^2C 1	UART 1	GPIO[61:60]
		CAN 1	UART 0	GPIO[63:62]
	UART 0	I^2C 1	CAN 0	GPIO[65:64]
	I^2C 0	UART 1	CAN 1	GPIO[67:66]
	CAN 0	UART 0	I^2C 0	GPIO[69:68]
	UART 1 I^2C 1 SPI Slave 0 SPI Slave 1	—	—	GPIO[70:59]
Dedicated	Dedicated	—	—	—
DDR SRAM	SDRAM	—	—	—

System Manager

The HPS peripheral set includes a system manager. The system manager contains logic to control the following system functions and steer the connections of the HPS dedicated I/O pins:

- HPS I/O pin multiplexing
- Memory-mapped control signals to other modules
- Controlling ECC in RAMs and injecting errors

Scan Manager

The HPS peripheral set includes a scan manager. The scan manager configures the HPS I/O. You can also use the scan manager to extend the HPS JTAG connection into the FPGA control block. Finally, the scan manager also connected to the freeze controller to ensure that HPS I/O are frozen prior to reconfiguration.

HPS Peripheral Timers

The HPS peripherals provide additional general-purpose timers and watchdog timers, beyond those already integrated within the MPU subsystem.

General-Purpose Timers

The HPS peripheral set includes the following 32-bit general-purpose timers. These timers are in addition to the private timers built into the MPU subsystem.

- osc1_timer0
- osc1_timer0
- sp_timer0
- sp_timer1

Each 32-bit timer supports the following operating modes. In each case, an interrupt is generated whenever the 32-bit binary down counter reaches zero.

- Free-running mode—decrementing from the maximum value (0xFFFFFFFF). Reloads maximum value upon reaching 0.
- User-defined count mode—effectively generates a periodic interrupt. Decrements from the user-defined count value loaded from the load count register. Reloads the user-defined count upon reaching 0.

A timer supports a timeout period of up to 43 seconds when the timer clock frequency is 100 MHz.

Watchdog Timers

The HPS peripheral set includes two watchdog timers. These timers are in addition to the private watchdog timers built into MPU subsystem. The 32-bit watchdog timers are `watchdog0` and `watchdog1`.

The watchdog supports 16 fixed timeout range values, and software chooses which timeout range is desired. The timeout ranges are from a minimum of 2^{16} clock periods to a maximum of 2^{31} clock periods. Once the counter reaches zero, either an interrupt or reset request is generated. Alternatively, the watchdog can be programmed to assert an interrupt request on a timeout and, if the interrupt is not serviced by software before a second timeout occurs, generate a reset request.

SDRAM Controller Subsystem

The HPS SDRAM controller subsystem enables efficient access to external SDRAM for the MPU, level 3 (L3) interconnect, and the FPGA fabric. The SDRAM controller implements an interface to the FPGA fabric and the HPS that accepts Advanced Microcontroller Bus Architecture (AMBA®) Advanced eXtensible Interface (AXI™) and Avalon Memory-Mapped (Avalon-MM) bus transactions, and converts those transactions to the correct commands for the SDRAM. The SDRAM controller subsystem manages traffic to the SDRAM device to maximize the system memory performance.

SDRAM Controller Subsystem Features

- Support for DDR2, DDR3, LPDDR1 and LPDDR2 SDRAM.
- Address range up to 4 GBytes of memory
- Maximum clock rate up to 400 MHz (800 Mbps data rate) in Cyclone V SoC FPGAs, and 533 MHz (1066 Mbps data rate) in Arria V SoC FPGAs
- Supports data widths of 8-, 16-, and 32-bits
- Integrated 24- and 40-bit ECC for 16- and 32-bit data widths
- Low-voltage 1.35V DDR3L and 1.2V DDR3U support
- Run-time configuration of I/O, size, mode, and memory timing
- Multiport front end (MPFE) provides interfaces to SDRAM for the MPU, L3 interconnect, and FPGA modules
- FPGA fabric interface supports Avalon-MM and AMBA AXI-3
- Power management allowing self refresh, partial array self refresh (PASR), power down and deep power down (LPDDR2 only)
- Memory protection to allow software configurable basic protection of memory from all masters in the system
- AMBA AXI exclusive access support
- Command reordering (look-ahead bank management)
- Data reordering (out of order transactions)
- Deficit round-robin arbitration with aging for bandwidth management
- High-priority bypass for latency sensitive traffic
- Dedicated I/O to external memory

SDRAM Controller Subsystem Interfaces

The SDRAM controller subsystem connects directly to the MPU subsystem, the L3 interconnect, and the FPGA fabric. The memory interface consists of the SDRAM controller, the PHY, control and status registers (CSRs), and their associated interfaces.

Table 4 describes the SDRAM controller subsystem interfaces.

Table 4. SDRAM Controller Subsystem Interfaces

Interface	Description
MPU subsystem interface	SDRAM controller is connected to the MPU with a dedicated 64-bit AMBA AXI port.
L3 interconnect interface	The SDRAM controller is connected to the L3 interconnect with a dedicated 32-bit AMBA AXI port.
CSR interface	The CSR interface provides a dedicated ARM Advanced Peripheral Bus (APB™) slave port driven by the L4 slave peripheral bus to access the CSRs of the SDRAM controller subsystem.

FPGA-to-HPS SDRAM interface	The FPGA-to-HPS SDRAM interface provides the FPGA fabric access to the HPS SDRAM controller subsystem. The interface has three port types for use as AMBA AXI or Avalon-MM ports:
	■ Command port
	  ■ Read/write commands
	  ■ Address
	  ■ Burst commands
	■ 64-bit read data ports
	  ■ Data returned from a read of memory
	  ■ Handshaking signals for the bus protocol
	■ 64-bit write data ports
	  ■ Data to be written to memory
	  ■ Status on the completion of a write for the specified bus protocol

In addition to these interface ports, a sideband port allows commands such as refresh and SDRAM power control. Table 5 shows the number of available ports.

Table 5. Number of Available Ports

Port Type	Number
Command	6
64 bit read data	4
64 bit write data	4
Sideband	1

Table 6 shows the number of ports required to configure different bus protocols, based on type and data width. You can configure the FPGA-to-HPS SDRAM interface with the following characteristics:

- Avalon-MM ports and AMBA AXI ports can be mixed and matched as required by the fabric logic within the bounds of the number of ports provided to the fabric.
- Each Avalon-MM and AMBA AXI port on the FPGA-to-HPS SDRAM interface is clocked independently with both pseudo-synchronous and asynchronous relationships supported relative to the SDRAM controller clock.

- The FPGA-to-HPS SDRAM ports are configured during FPGA configuration.

Table 6. Bus Protocol Ports Required for Protocol Configuration

Bus Protocol	Command	Read Data	Write Data
8-, 16-, 32-, 64-bit AMBA AXI	2	1	1
128-bit AMBA AXI	2	2	2
256-bit AMBA AXI	2	4	4
32-, 64-bit Avalon-MM bidirectional	1	1	1
128-bit Avalon-MM bidirectional	1	2	2
256-bit Avalon-MM bidirectional	1	4	4
32-, 64-bit Avalon-MM write only	1	0	1
128-bit Avalon-MM write only	1	0	2
256-bit Avalon-MM write only	1	0	4
32-, 64-bit Avalon-MM read only	1	1	0
128-bit Avalon-MM read only	1	2	0
256-bit Avalon-MM read only	1	4	0

SDRAM Controller

The SDRAM controller provides high memory bandwidth, low latency, and run-time programmability. The controller can reorder data to reduce row conflicts and bus turn-around time by grouping reads and writes together, allowing for efficient traffic patterns and reduced latency. The SDRAM controller consists of the MPFE and the single-port controller.

MPFE

The MPFE schedules the various pending transactions from the configured ports and sends the scheduled memory transactions to the single-port controller. The MPFE handles all functions related to individual ports.

Single-Port Controller

The single-port controller queues pending SDRAM bursts. The controller chooses the most efficient burst to send next, keeps the SDRAM pipeline full, and ensures that all SDRAM timing parameters are met. Transactions passed to the single-port logic for a single page are executed in order, although transactions can be reordered between pages. Each SDRAM burst read or write is converted to the appropriate Altera PHY interface commands to open a bank on the correct row for the transaction (if required), execute the read or write command, and finally precharage the bank (if required). The single-port controller performs the following tasks:

- Implements command and data reordering.
- Implements command reordering by looking ahead at the command sequence to see what banks can be put into the correct state to allow a read or write command to be executed.
- Implements data reorder by determining which data transactions can be dispatched even if the data transactions are executed in an order different than they were received from the multiport logic.

DDR PHY

The double data rate (DDR) PHY is the interface between the memory controller and the memory devices that perform read and write operations to the memory.

The DDR PHY implements the following functions:

- Calibration—the DDR PHY supports the JEDEC specified steps in synchronizing the memory timing between the controller and the SDRAM devices. The calibration algorithm is implemented in software.
- SDRAM device initialization—the DDR PHY performs the mode-register write operations to initialize the devices. The DDR PHY handles re-initialization after a deep power down.
- Conversion between single and double data rate movement of data.

The DDR PHY has dataflow components and also a calibration controller which handles the sequencing of the calibration for the SDRAM interface timing.

Further Information

- SoC FPGA Overview:
  www.altera.com/devices/processor/soc-fpga/proc-soc-fpga.html
- *SoC FPGA Product Overview Advance Information Brief* (AIB):
  www.altera.com/literature/hb/soc-fpga/aib-01017-soc-fpga-overview.pdf
- Arria V FPGAs: Balance of Cost, Performance, and Power:
  www.altera.com/devices/fpga/arria-fpgas/arria-v/arrv-index.jsp
- Cyclone V FPGAs: Lowest System Cost and Power:
  www.altera.com/devices/fpga/cyclone-v-fpgas/cyv-index.jsp
- Dual-Core ARM Cortex-A9 MPCore Processor
  www.altera.com/devices/processor/arm/cortex-a9/m-arm-cortex-a9.html
- Using the Virtual Target with the ARM Cortex-A9 MPCore Processor:
  www.altera.com/devices/processor/arm/cortex-a9/virtual-target/proc-a9-virtual-target.html
- Processors from Altera and Embedded Alliance Partners:
  www.altera.com/devices/processor/emb-index.html
- Presentation: “ARM NEON Technology Introduction”:
  http://www.arm.com/files/pdf/AT_-_NEON_for_Multimedia_Applications.pdf
- Webcast: “Getting to Know the ARM-Based SoC FPGA”:
  www.altera.com/education/webcasts/all/wc-2011-arm-based-soc-fpga.html
- White Paper: *Strategic Considerations for Emerging SoC FPGAs*
  http://www.altera.com/literature/wp/wp-01157-embedded-soc.pdf
- White Paper: *ARM Cortex-A9 Processors*
  http://www.arm.com/files/pdf/ARMCortexA-9Processors.pdf

Document Revision History

Table 7 shows the revision history for this document.

Table 7. Document Revision History

Date	Version	Changes
October 2011	1.0	Initial release.

: