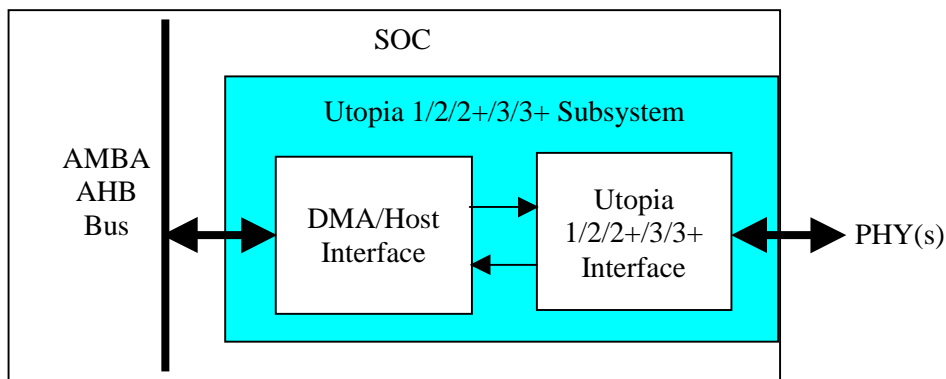


AU-NB9003: Utopia 1/2/2+/3/3+ AMBA Subsystem Core **AMBA AHB Bus Utopia 1/2/2+/3/3+ with DMA**

The AU-NB9003 Utopia 1/2/2+/3/3+ AMBA Subsystem provides a Utopia 1/2/2+/3/3+ peripheral subsystem for AMBA based SOCs. It contains a Utopia 1/2/2+/3/3+ Interface between UTOPIA/POS-PHY compliant PHY Chips and the application layers such as ATM. This Utopia Interface connects seamlessly to the AMBA AHB Bus. A DMA Engine is included to move the packet/cell data. The figure below shows the Utopia 1/2/2+/3/3+ Subsystem used within an SOC. The Utopia 1/2/2+/3/3+ AMBA Subsystem Core is available as a synthesizable Verilog model from Aurora VLSI, Inc. Contact CustomerService@auroravlsi.com.



The Utopia 1/2/2+/3/3+ Interface operates in full duplex mode with PHY devices and supports up to 31 devices with a maximum data rate of 3.2 Gb/s. It is compliant to UTOPIA Level 3, Level 2, Level 1 Standards and POS-PHY Level 2 and 3 standards. Its mode can be switched between U1, U2 and U3.

The Utopia 1/2/2+/3/3+ Subsystem is the Utopia master. An application, such as a host or embedded processor, initiates transmit operations to the Utopia 1/2/2+/3/3+ Subsystem that then sends the transmit data to the PHY. As the master, the Utopia 1/2/2+/3/3+ Subsystem polls ports to find receive data, and then brings the receive data into its Receive Data FIFO.

Internal to an SOC, the Utopia 1/2/2+/3/3+ Subsystem is a peripheral on the AMBA AHB Bus. Direct Memory Access- DMA, between the DMA/Host Interface block of the Utopia 1/2/2+/3/3+ Subsystem, and AMBA Bus targets, is used to transfer the packet/cell data to/from the Utopia 1/2/2+/3/3+ Interface block. When doing DMA, the Utopia 1/2/2+/3/3+ Subsystem is an AMBA Bus master. The Utopia 1/2/2+/3/3+ Subsystem is an AMBA Bus slave for control, address, and status register/FIFO accesses. All control, address, and status register/FIFO accesses originate in an AMBA Bus master outside of the Utopia 1/2/2+/3/3+ Subsystem, such as a host processor.

Utopia 1/2/2+/3/3+ AMBA Subsystem features are summarized:

Utopia 1/2/2+/3/3+

- Supports both ATM Forum's standards and Saturn Group's POS-PHY Interface standards.
- Programmable option for U1, U2, U2+(or POS-PHY Level 2), U3, and U3+ standards.
- Supports both packet and cell level transfers.
- Number of ports for polling can be programmed.
- Receive Ports can be enabled or disabled dynamically.
- Transfer (PHY/core transfer length for every transfer) count can be independently programmed for each port with a maximum of 64 bytes.
- Round robin mechanism of polling on the receive side.
- Parity generation/checking with a programmable polarity option.
- User defined (through Verilog parameters) FIFO sizes for all transmit and receive FIFOs.
- Supports synchronous memories for FIFOs.
- Synthesis options for receive only, transmit only, or both receive and transmit.

DMA/Host Interface

- AMBA AHB Bus interface
- 2 channel DMA Engine
 - transmit data DMA from data source to Transmit Data FIFO
 - receive data DMA from Receive Data FIFO to data destination
- Physical DMA addresses
- Programmable DMA starting address
- Programmable DMA transfer count- up to 64 Kbytes
- Programmable DMA AMBA Bus interface transaction size- 8 to 1024 bytes
- Programmable DMA AMBA Bus data transfer size- 4 or 8 bytes
- Locked DMA operation optional (software programmable)
- Direct software writes or information extracted from descriptors in memory, to program DMA control information
- Dedicated AMBA Bus master interface for each DMA channel
- AMBA Bus slave interface for
 - register reads and writes
 - writes to the Transmit Address FIFO
 - reads of the Transmit Status FIFO
- Interrupts:
 - transmit data DMA completed
 - receive data DMA completed
 - Transmit Data FIFO at empty threshold
 - Receive Data FIFO at full threshold

The core is delivered as a synthesizable RTL Verilog model. Deliverables include:

- RTL Verilog source code model of the core
- Verilog testbench and test cases
- Synthesis scripts examples
- Complete detailed documentation and training class notes

Utopia 1/2/2+/3/3+ Interface

The Utopia 1/2/2+/3/3+ Subsystem includes the Stargate SSN9003 Utopia 1/2/2+/3/3+ Interface Core. Additional logic at the application interface of the Utopia 1/2/2+/3/3+ Interface provides a DMA Engine, AMBA Bus interface, and host processor interrupts for the Utopia 1/2/2+/3/3+ Subsystem.

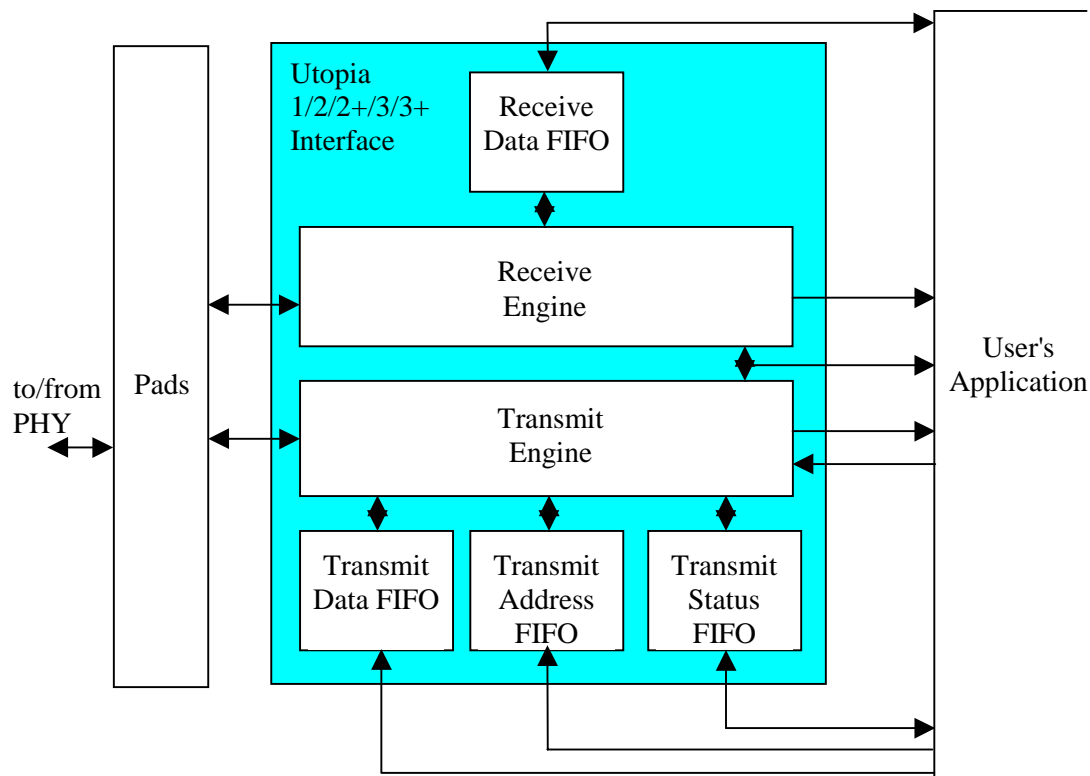
A block diagram of the Utopia 1/2/2+/3/3+ Interface is shown below.

The Utopia 1/2/2+/3/3+ Interface is a shell that holds a Receive Engine and Transmit Engine. The Receive Engine and Transmit Engine are completely independent of each other, with no interaction between them. The Utopia 1/2/2+/3/3+ Subsystem can be built with the Receive Engine and its FIFOs, or the Transmit Engine and its FIFOs, or both Receive and Transmit Engines and all FIFOs.

The Receive Engine receives data from the PHYs and provides it to the DMA/AMBA Interface. It includes Configuration Registers for receiving data.

The Transmit Engine transmits data to the PHYs from the DMA/AMBA Interface. It polls PHY ports to check if they are available to accept data transmitted to them in multi-PHY mode. (this is not necessary in single PHY mode), and captures this availability status in the Transmit Status FIFO. It also includes Configuration Registers for transmitting data.

The FIFOs are non-synthesizable synchronous SRAM macros. The Receive Data FIFO holds incoming receive data from the PHY. The Transmit Address FIFO holds PHY port numbers to be polled to check if they are available to accept data transmitted from the Utopia 1/2/2+/3/3+ Interface. The Transmit Status FIFO holds the status returned by each PHY when it is polled for its availability to accept data transmitted from the Utopia 1/2/2+/3/3+ Interface. The Transmit Data FIFO holds outgoing transmit data for the PHY.



DMA/Host Interface

The DMA/Host Interface includes a two channel DMA Engine. One channel is used to transfer transmit data from the data's source, over the AMBA Bus, to the Transmit Data FIFO. Receive data is sent from the Receive Data FIFO, over the AMBA Bus, to the received data's destination, by the second DMA channel.

Block moves of up to 64K bytes are supported by the DMA Engine. The exact transfer count of each DMA operation is the size of the data block that is being transferred, and is set by software. DMA operations are done as a series of Transmit Data FIFO or Receive Data FIFO accesses, and bus transactions to move the data block. The length of each bus transaction is software programmable so that it can be optimized according to system characteristics. Each individual bus transaction is from 8 bytes to 1024 bytes according to a value programmed into a DMA channel's control register. Additionally, the data size of each data transfer on the bus is software programmable to be four or eight bytes.

The series of accesses that make up a complete DMA operation may be locked together so that no other device gets the AMBA Bus until the DMA operation is finished. This is under software control.

The DMA starting address is set by software. This is the transmit data source starting address in memory for transmit data, and the receive data destination starting address in memory for receive data. These starting addresses are incremented by the DMA Engine to form the AMBA Bus transaction addresses as the DMA operation progresses. All addresses are physical addresses.

The DMA control information that is set by software- starting address, transfer count, bus transaction size, data transfer size, and lock flag, can be set by direct software writes to Utopia 1/2/2+/3/3+ Subsystem registers that hold this DMA control information. Alternatively, this DMA control information can be set from descriptors in memory that hold the DMA control information. Scatter/gather DMA is done using a chained descriptor list as the DMA control information source. When using descriptors to set the DMA control information, the descriptors are initialized by software. To support DMA configuration from descriptors, the DMA/Host Interface contains logic to read the descriptors from memory, and load the appropriate DMA/Host Interface registers with the DMA control information.

A DMA operation begins when the DMA channel is enabled, after the starting address, transfer count, bus transaction size, data transfer size, and lock flag are configured. The DMA channel moves the data block, and the DMA operation ends when the entire frame has been moved.

The DMA/Host Interface generates interrupts to notify the host processor of events that are important to driver software. These interrupts include:

- Interrupt upon a completed DMA operation between the transmit data source and Transmit Data FIFO
- Interrupt upon a completed DMA operation between the Receive Data FIFO and receive data destination
- Interrupt upon the data level in the Transmit Data FIFO falling below the software programmable transmit empty threshold
- Interrupt upon the data level in the Receive Data FIFO going above the software programmable receive full threshold