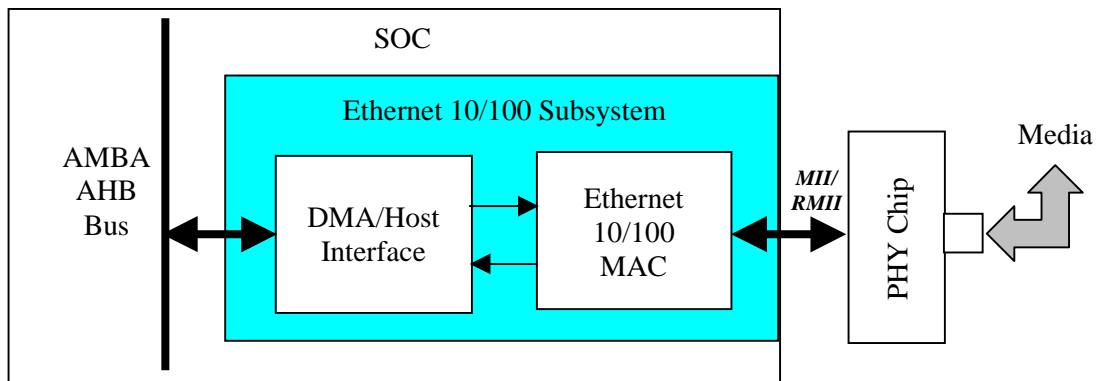


## **AU-NB8006: Ethernet 10/100 AMBA Subsystem Core**

### **AMBA AHB Bus Ethernet 10/100 MAC with DMA**

The AU-NB8006 Ethernet 10/100 AMBA Subsystem provides an Ethernet 10/100 peripheral subsystem for AMBA based SOCs. It contains an Ethernet 10/100 MAC that connects seamlessly to the AMBA AHB Bus. A DMA Engine is included to move Ethernet frame data. The figure below shows its use within an SOC. The Ethernet 10/100 AMBA Subsystem Core is available as a synthesizable Verilog model from Aurora VLSI, Inc. Contact [CustomerService@auroravlsi.com](mailto:CustomerService@auroravlsi.com).



Ethernet line speeds of 10mb/s and 100mb/s are supported. The Ethernet 10/100 MAC block of the Ethernet 10/100 Subsystem interfaces to an external Ethernet PHY through the industry standard Media Independent Interface (MII) or Reduced Media Independent Interface (RMII), as determined by a configuration register.

Internal to an SOC, the Ethernet 10/100 Subsystem is a peripheral on the AMBA AHB Bus. Direct Memory Access- DMA, between the DMA/Host Interface block of the Ethernet 10/100 Subsystem, and AMBA Bus targets, is used to transfer the Ethernet transmit and receive data. When doing DMA, the Ethernet 10/100 Subsystem is an AMBA Bus master. The Ethernet 10/100 Subsystem is an AMBA Bus slave for control and status register accesses. All control and status register accesses originate in an AMBA Bus master outside of the Ethernet 10/100 Subsystem, such as a host processor.

Ethernet 10/100 AMBA Subsystem features are summarized:

Ethernet 10/100 MAC

- 10mb/s or 100mb/s Ethernet line speeds
- IEEE 802.3 compatible, supports IEEE 802.3x and IEEE 802.3Q for flow control
- Industry standard MII or RMI interface to the PHY
- Optional two-part deferral implemented
- Full and half duplex
- Automatic retries after collision, programmable retry counter
- Automatic padding and removal of PAD bytes to meet minimum frame size
- FCS generation for transmitted packets, and checking on received packets
- MDIO interface for PHY management
- Programmable address filtering
- Frame status captured in 2 status FIFOs
  - Transmit Status FIFO- 4 entries
  - Receive Status FIFO- 4 entries
- 128 to 2048 byte Transmit FIFO
- 128 to 2048 byte Receive FIFO

DMA/Host Interface

- AMBA AHB Bus interface
- 2 channel DMA Engine
  - transmit data DMA from data source to Transmit FIFO
  - receive data DMA from Receive 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
- Interrupts:
  - transmit frame DMA completed
  - receive frame DMA completed
  - frame transmit completed
  - frame receive completed

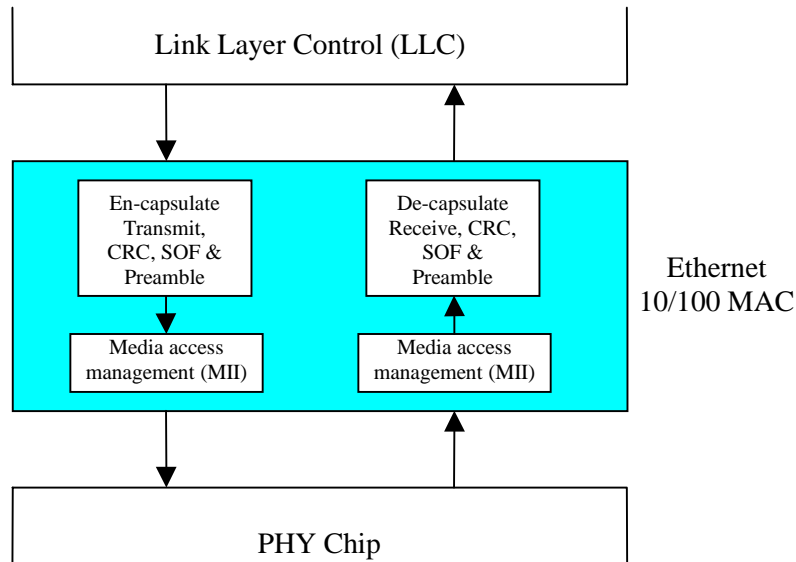
The core is delivered as a synthesizable RTL Verilog model. Deliverables include:

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

### **Ethernet 10/100 MAC**

The Ethernet 10/100 Subsystem includes the Stargate SSN8006 Ethernet 10/100 MAC Core. Additional logic at the application interface of the Ethernet 10/100 MAC provides a DMA Engine, AMBA Bus interface, and host processor interrupts for the Ethernet 10/100 Subsystem.

The Ethernet 10/100 MAC is an Ethernet 10/100mb/s Media Access Controller. It performs the IEEE 802.3 recommended service specification for Media Access Control. A block representation of Ethernet 10/100 MAC as per the IEEE standard specifications is shown below.



Transmit data is transferred from the LLC to the Ethernet 10/100 MAC. The Ethernet 10/100 MAC packages the data, including Preamble, SOF, PAD bytes as needed, and FCS. When the transmitter is not in the deferred mode, packets are transmitted to the PHY. The industry standard MII or RMII interface is used to transfer the packet to the PHY chip.

Received data is checked for size and FCS correctness, and its destination address is filtered. Then if it passes address filtering, the received data is transferred to the LLC. If the length field in the frame is less than the minimum frame size, the PAD bytes are stripped from the frame. All of these operations can be enabled or disabled through the Ethernet 10/100 MAC configuration registers.

Transmit and receive status are captured for each transmitted and received frame, respectively. Transmit frame status is captured in a four entry Transmit Status FIFO. The four entry Receive Status FIFO captures receive frame status. The host processor may read these status FIFOs or optionally, the Ethernet 10/100 Subsystem automatically reports the status by sending it out over the AMBA Bus.

The Ethernet 10/100 MAC has a Transmit FIFO and Receive FIFO to hold transmit data from the AMBA Bus and receive data to be driven over the AMBA Bus, respectively. The size of each FIFO is user configurable from 128 bytes to 2048 bytes.

## **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 FIFO. Receive data is sent from the Receive 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 frame that is being transferred, and is set by software. DMA operations are done as a series of Transmit FIFO or Receive 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 Ethernet 10/100 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 FIFO
- Interrupt upon a completed DMA operation between the Receive FIFO and receive data destination
- Interrupt upon completing one or multiple frame transmissions- typically used to check transmit status
- Interrupt upon one or multiple complete frames received and sent out over the AMBA Bus - typically used to start processing the received frame