**🔹 1. eth\_mac\_10g.v – Top-Level 10G Ethernet MAC Module**

**✅ Purpose:**

This is the top-level wrapper module for the 10G Ethernet MAC. It integrates:

* Transmit and receive logic
* Flow control (pause frame support)
* MAC control (optional)
* XGMII interface on one side and AXI-Stream on the other

**📦 Submodules Likely Included:**

* axis\_xgmii\_rx\_64 or axis\_xgmii\_rx\_32 (based on data width)
* axis\_xgmii\_tx\_64 or axis\_xgmii\_tx\_32
* mac\_pause\_ctrl\_rx
* mac\_pause\_ctrl\_tx
* mac\_ctrl\_rx (optional MAC control logic)
* mac\_ctrl\_tx (optional MAC control logic)

**⚙️ Interfaces:**

* **XGMII TX/RX**: 64-bit or 32-bit wide bus, based on mode
* **AXI-Stream TX/RX**: For user logic to transmit/receive Ethernet frames
* **Flow Control**: Optional pause frame support using mac\_pause\_ctrl\_\* modules

**🔹 2. axis\_xgmii\_rx\_64.v and axis\_xgmii\_rx\_32.v – XGMII Receiver to AXI-Stream**

**✅ Purpose:**

* Converts XGMII (64-bit or 32-bit) receive data into AXI-Stream Ethernet frames.
* Detects preambles, Start-of-Packet (SOP), End-of-Packet (EOP)
* Removes inter-frame gaps (IFG)
* Reports errors (e.g., bad FCS, early termination)

**🔁 Differences:**

* \_rx\_64.v: Used in 64-bit XGMII mode (common for 10G)
* \_rx\_32.v: Used in 32-bit mode (e.g., for simulation or resource-constrained designs)

**⚙️ Key Operations:**

* Watches for SFD (Start Frame Delimiter)
* Extracts destination/source MAC and ethertype
* Validates frame structure and optionally CRC
* Sends data via AXI-Stream output ports

**🔹 3. axis\_xgmii\_tx\_64.v and axis\_xgmii\_tx\_32.v – AXI-Stream to XGMII Transmitter**

**✅ Purpose:**

* Takes AXI-Stream Ethernet frames and sends them over XGMII
* Adds preamble, Start Frame Delimiter (SFD), pads frame if needed
* Inserts IFG (inter-frame gap)
* Calculates and appends CRC (FCS)

**🔁 Differences:**

* \_tx\_64.v: Works with 64-bit XGMII (most common)
* \_tx\_32.v: Works with 32-bit XGMII interface

**⚙️ Key Operations:**

* Serializes AXI frame into XGMII data/control characters
* Inserts IDLE characters when not transmitting
* Ensures compliance with IEEE 802.3

**🔹 4. mac\_ctrl\_rx.v – MAC Control RX**

**✅ Purpose:**

* Handles reception of special MAC control frames (e.g., pause frames)
* Detects opcode = 0x0001 which signifies a pause frame
* Extracts pause time value
* Passes control info to the pause control logic

**🔹 5. mac\_ctrl\_tx.v – MAC Control TX**

**✅ Purpose:**

* Builds and sends MAC control frames like pause frames
* Triggered by internal flow control logic or external logic
* Encodes frame with:
  + DA: 01-80-C2-00-00-01 (reserved multicast for pause)
  + Opcode: 0x0001
  + Pause time: 16-bit value

**🔹 6. mac\_pause\_ctrl\_rx.v – Pause Frame Receiver Control**

**✅ Purpose:**

* Handles the logic once a pause frame is received
* Interprets pause time and applies it to internal pause state
* Coordinates pause signal assertion/deassertion based on timers

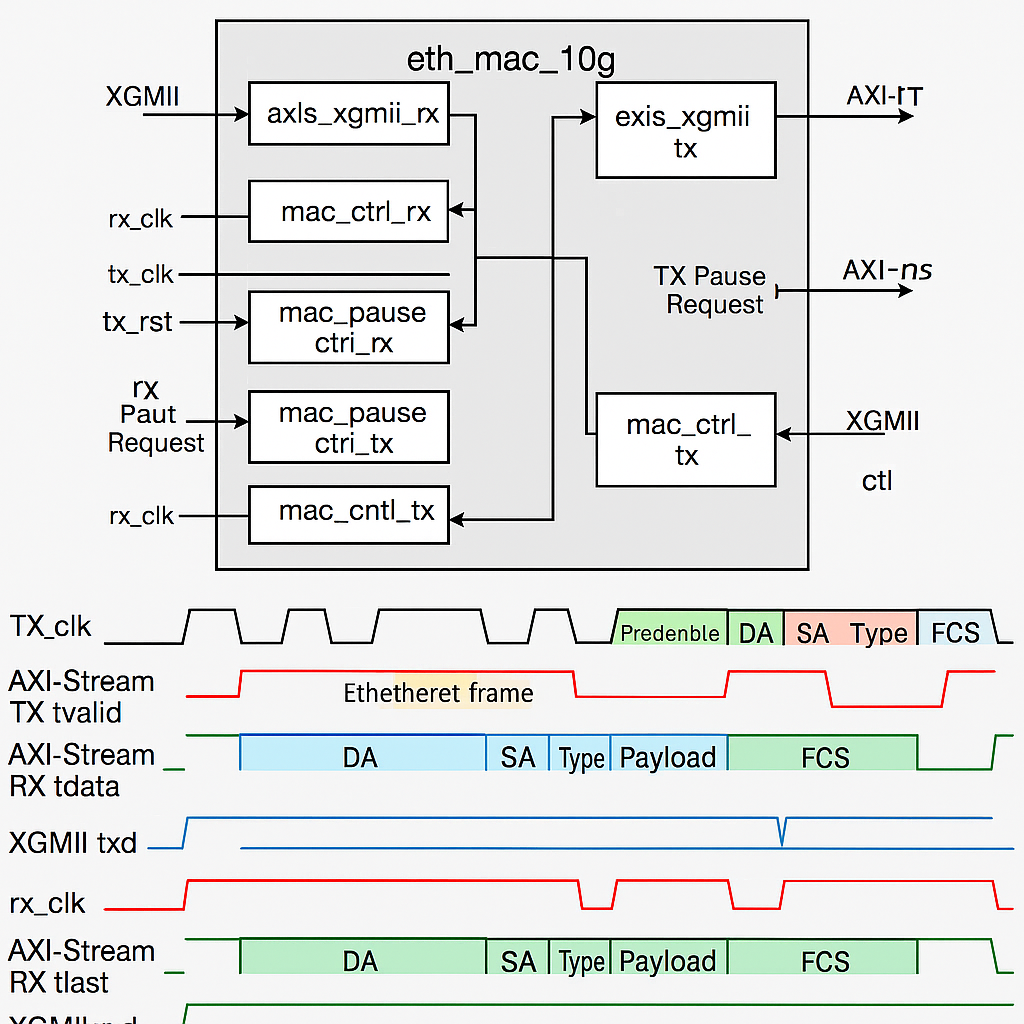
**🔹 7. mac\_pause\_ctrl\_tx.v – Pause Frame Transmitter Control**

**✅ Purpose:**

* Generates pause request logic
* Handles pause frame transmit timing and interval enforcement
* Works in conjunction with mac\_ctrl\_tx.v

| **File** | **Function** |
| --- | --- |
| axis\_xgmii\_rx\_32.v / axis\_xgmii\_rx\_64.v | Converts XGMII RX interface to AXI4-Stream — decodes received Ethernet frames. Supports 32- or 64-bit XGMII data bus width. |
| axis\_xgmii\_tx\_32.v / axis\_xgmii\_tx\_64.v | Converts AXI4-Stream TX data to XGMII TX interface — adds preamble, CRC, and encodes Ethernet frames. |
| eth\_mac\_10g.v | Top-level 10G Ethernet MAC wrapper — connects RX/TX datapaths, pause, and control logic. |
| mac\_ctrl\_rx.v | Handles incoming MAC control frames (like pause frames) from the network. |
| mac\_ctrl\_tx.v | Sends MAC control frames (pause, etc.) to the network. |
| mac\_pause\_ctrl\_rx.v | Processes RX pause frames — extracts pause time and generates flow control. |
| mac\_pause\_ctrl\_tx.v | Implements TX-side pause frame generation — manages pause request timing. |

**Block Diagram – 10G Ethernet MAC TX & RX**



**Hierarchy Flow:**

+----------------------+

| eth\_mac\_10g |

|----------------------|

| |

+--------v--------+ +---------v-------+

| axis\_xgmii\_tx\_32| | axis\_xgmii\_rx\_32|

| axis\_xgmii\_tx\_64| | axis\_xgmii\_rx\_64| <- Optional

+--------+--------+ +---------+-------+

| |

| |

+---v---+ +---v---+

| XGMII | | XGMII |

| PHY | | PHY |

+-------+ +-------+

Control Path:

+--------------------+

| MAC Pause/Control |

| mac\_ctrl\_tx / rx |

| mac\_pause\_ctrl\_tx |

| mac\_pause\_ctrl\_rx |

+--------------------+

**TX Waveform Example (axis\_xgmii\_tx\_64)**

A white background with black lines and white text

AI-generated content may be incorrect.

**Explanation:**

* First 8B: Preamble + SFD
* Next 2 cycles: Ethernet header (14B)
* Payload: Starts after header
* tlast: Indicates end of frame

**RX Waveform Example (axis\_xgmii\_rx\_64)**

A screen shot of a computer

AI-generated content may be incorrect.