1. Introduction:

Ethernet is the basis of LAN networks. The current LAN market is characterized by an, up to now, unknown degree of standardization on Ethernet. Due to its huge market share, Ethernet, despite some disadvantages, scores over all alternative technologies.

* Time-Sensitive Networking (TSN): is a set of standards that provide the ability to synchronize time and manage data traffic on Ethernet networks, allowing network devices to process data in real-time. This ensures the stability and reliability of the network.
* Audio Video Bridging (AVB): is a set of standards that provide the ability to transmit real-time audio and video over Ethernet networks. AVB includes key features such as Stream Reservation Protocol (SRP), Timing and Synchronization for Time-Sensitive Applications (gPTP), and Quality of Service (QoS).

1. Physical layer:

* The Layer 1 of Ethernet AVB (Audio Video Bridging) is the Physical layer, which defines the physical parameters of the network, including cable characteristics, connections, signals, and transmission speeds.
* Specifically, the Physical layer in Ethernet AVB defines the basic requirements for the physical structure of an Ethernet AVB network system.
  1. Bit stream:

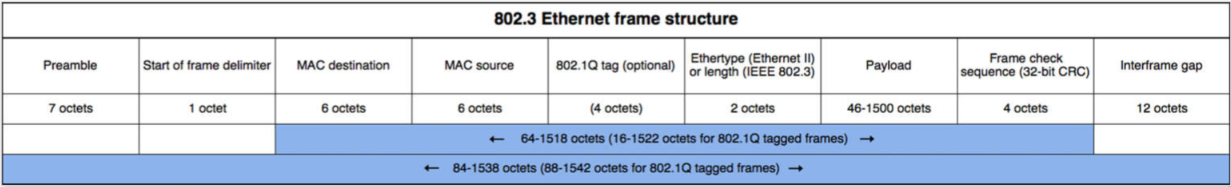
The Physical layer in Ethernet AVB defines the physical parameters of the network, including how data is transmitted over physical media such as coaxial cable, optical fiber, or wireless communication. During data transmission, the data is encoded and converted into a sequence of binary digits 0 and 1, known as a bit stream.

* 1. Physical Medium:

Physical Medium is a part of the Physical layer in Ethernet AVB, which defines the physical transmission media to transmit the bit stream between devices. The physical transmission media can include coaxial cable, optical fiber, or wireless

* 1. Methods of Transferring Data:

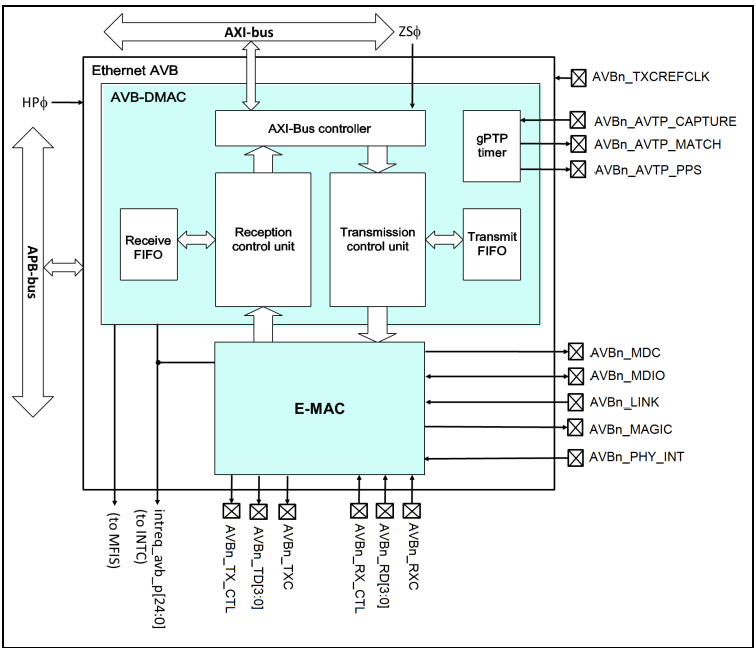
1. Data Link:
   1. Ethernet Frame



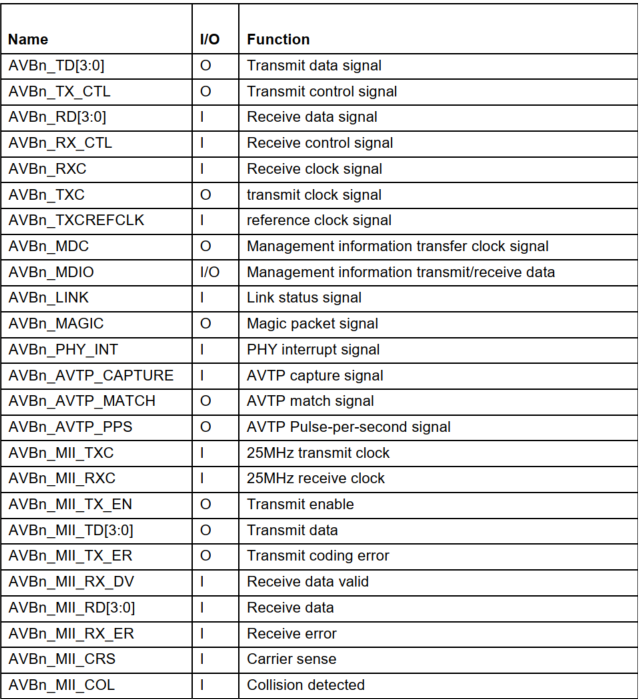
Ethernet Frame: A Frame is the basic unit of data in Ethernet. It contains information such as the MAC addresses of the sender and receiver, error control parameters, and the actual data being transmitted. Each frame consists of the following fields:

* Preamble: 7-byte synchronization sequence used to synchronize data transmission between computers. (7 byte)
* Start Frame Delimiter: 1-byte field following the Preamble, with a value of 10101011, used to synchronize data frames as they are transmitted over the network (1 byte)
* Destination Address: the MAC address of the intended recipient of the frame (6 byte)
* Source Address: the MAC address of the sender of the frame. (6 byte)
* 802.1Q tag: a part of Ethernet Frame used to mark data packets belonging to a specific Virtual LAN (VLAN). The 802.1Q tag consists of 4 bytes (from the 13th byte to the 16th byte in the Ethernet Frame) and includes the following fields: (4 byte)
  + TCI (Tag Control Information): first two bytes of the 802.1Q tag and contains control information, including a 3-bit PRI (Priority Code Point) to determine the priority of the packet in case of network congestion and a 1-bit CFI (Canonical Format Indicator) to determine the format of the MAC address table
  + VLAN ID: the last 2 bytes of the 802.1Q tag and it contains the VLAN identifier, which is used to tag the data packets belonging to a specific VLAN.
* Type/Length: Determines the type of data being transmitted or the length of the frame. (2 byte)
* Payload: Data to be transmitted (46 - 1500 byte)
* Frame Check Sequence (FCS): Evaluates the integrity of a frame. (4 byte)
* Interframe gap: This is the idle time between consecutive Ethernet frames, used to ensure that frames do not collide when transmitted over the network. The interframe gap has a minimum length of 96 bits (equivalent to 9.6 μs at a data rate of 10 Mbps), but is often used with longer lengths to ensure stability and accuracy in data transmission over Ethernet networks (12 byte)
  1. MAC
* MAC Addresses: the unique physical addresses of each device connected to an Ethernet network. They are used to identify the sender and receiver of data transmission. MAC address is a 48-bit string represented in hexadecimal format and is typically divided into two parts: the first 24 bits are the OUI (Organizationally Unique Identifier) to identify the device manufacturer, and the remaining 24 bits are the unique address of the device. Example: 00-1B-63-84-45-E6
* MAC addresses are used in the Data Link Layer to identify devices in a network. When an Ethernet frame is transmitted, it will include the source MAC address and destination MAC address. Devices in the network will use the MAC address to determine if the frame is intended for them or not. If the destination MAC address matches the MAC address of the device, then the frame will be accepted and processed. Otherwise, if the destination MAC address does not match the MAC address of the device, then the frame will be discarded.

1. Ethernet AVB-IF
   1. Diagram:



* **AXI-bus** (Advanced eXtensible Interface bus): is a high-performance bus interface used for transferring data between modules and hardware in an integrated system. In Ethernet AVB-IF, AXI-bus is used to connect blocks such as E-MAC, AVB-DMAC, and FIFO.
* **APB-bus** (Advanced Peripheral Busis a smaller interface bus used to connect modules to smaller components of the system, such as timing and routing modules. In Ethernet AVB-IF, APB-bus is used to connect to modules such as the PTP timing and routing module.
* **E-MAC** (Ethernet Media Access Controller): is a block responsible for controlling the process of transmitting data on the Ethernet network. It performs functions such as error control, packaging data into Ethernet frames, and sending and receiving Ethernet frames on the network. It processes DMA transmission between the receive and transmit FIFO buffer and MII.
* **AVB-DMAC** (DMA transfer controller handles DMA transfers between data storage areas for receive and transmit in URAM and receive and transmit FIFO buffers:
  + **AXI-Bus**: This is a high-level communication bus used to connect the blocks in the AVB-DMAC block to other blocks in the system.
  + **Receive FIFO**: his block is used to store received Ethernet frames before they are processed.
  + **Reception control unit**: This block manages the processing of received Ethernet frames. It checks the destination MAC address and forwards the frame to the corresponding buffer for processing.
  + **Transmission control unit**: This block manages the transmission of Ethernet frames. It checks the transmit buffer and forwards Ethernet frames to the appropriate controller for transmission.
  + **Transmit FIFO**: This block is used to store Ethernet frames before they are transmitted.
  + **gPTP timer**: This block synchronizes the time between different Ethernet AVB-IF devices in the network using the Generalized Precision Time Protocol (gPTP). It allows for highly accurate time synchronization to support real-time applications in automobiles, such as engine control and infotainment systems.



* 1. Register:
* CCC: specifies the operating mode of the AVB-DMAC
* DBAT: register specifies the base address of the descriptor table in the URAM. Writing to this bit is only possible when the current operating mode is configuration mode.
* DLR:
* CSR: register is used to indicate the operating mode in which the AVB-DMAC is running and the individual communications states.
* RFLR: specifies the maximum length (in bytes) of frames that can be received by this LSI (must not be changed while reception is enabled)
* RPC: set padding for received frames
* UFCVs 0 -> 4:
* TGC: make settings related to transmission for the AVB-DMAC
* TCCR: controls transmission by the AVB-DMAC and is used to make related settings.
* UFCS: sets the stop levels for unread frames
* SFO: sets an offset into frames for use by the separation filter
  1. Operation:
     1. AVB-DMAC Operating Modes

Diagram

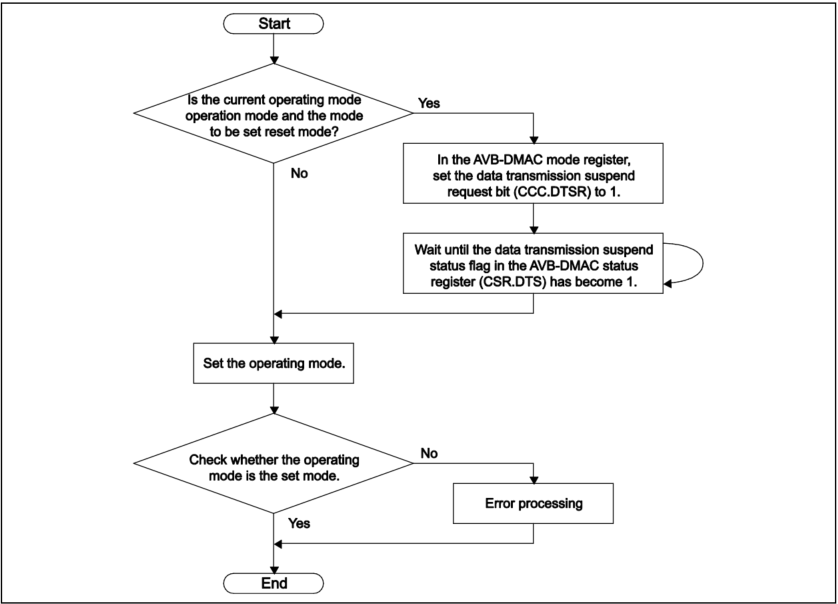
Description automatically generated

* + - 1. Operating Mode
         1. Reset mode
* In reset mode, only the AVB-DMAC operating mode control function is controllable and other functions are all stopped.
* This mode is designed for reduced power when the Ethernet function is not necessary
  + - * 1. Configuration mode
* The operation of most functions is stopped and all status registers are initialized to their reset values.
* By CCC.GAC it is possible to enable gPTP support already in CONFIG mode (pin 7 Register CCC = 1)
  + - * 1. Operation mode
* All functions of the AVB-DMAC can operate
  + - * 1. Standby mode
* The E-MAC can only be used to control the operating mode. Other functions cannot be used.
* By CCC.GAC (pin 7 Register CCC != 1)
  + - 1. How to Set the Operating Mode:

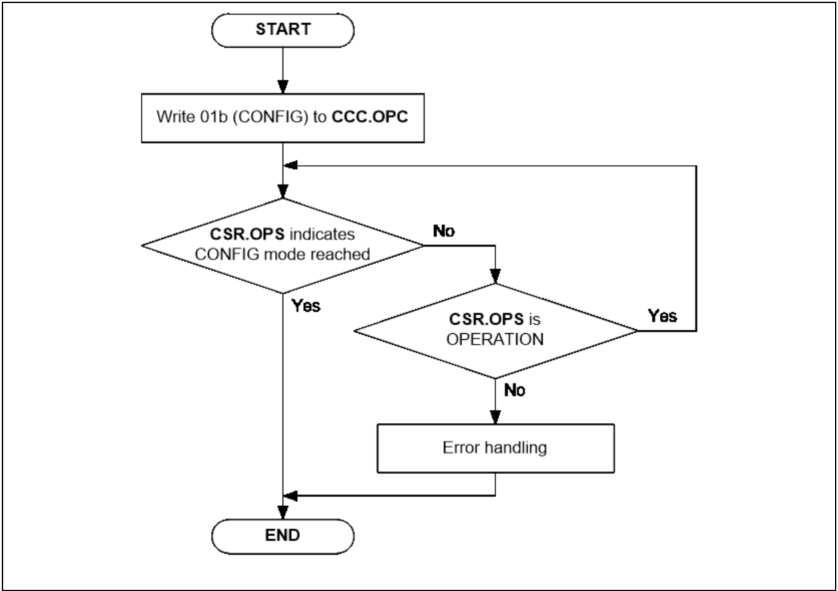
By AVB-DMAC mode register (CCC.OPC bit [1:0])

To check current operating mode => by AVB-DMAC status register (CSR.OPS bit [3:0])

* + - * 1. **Operation Mode to Configuration Mode**



* + - * 1. **Operation Mode to Configuration Mode**



* **Notes:** When the operating mode shifts to configuration mode, all status registers are cleared.

When need status register information, the reception and transmission path can be individually disabled and can check the status before leaving operating mode with follow.

We recommend following the procedure below in the case of this transition.

* Disable reception.
* Since reception actually stopping after being disabled requires time, wait for an interval equivalent to that for
* reception of a maximum length packet.
* Stop the software task that is generating data for transmission.
* Wait until the receive process status bit (CSR.RPO) and the transmit process status bits (CSR.TPO0 to 3) in
* the AVB-DMAC status register are set to 0.
* Capture all of the required status information.
* Set the operating mode configuration bits in the AVB-DMAC mode register (CCC.OPC) to initiate the
* transition to configuration mode.
  + - 1. Leave RESET Mode with Additional Configuration
  + To enable gPTP support in CONFIG mode:
* Write the value 01b to the CCC.OPC register to enter CONFIG mode.
* Use a 32-bit write access to set the corresponding bit for gPTP support in the configuration register (usually the AVB\_GPTP\_CTRL register).
* Set CCC.GAC to 1b to confirm the configuration write.
* Set CCC.CSEL to the intended configuration.
* Finally, write the value 00b to CCC.OPC to exit CONFIG mode and enter operational mode.
* **Note**: enabling gPTP support in CONFIG mode can only be done while Ethernet AVB-IF is in RESET mode and the AVB system is properly configured.
  + - 1. Operating Mode Transitions Due to Hardware
    1. Common Control for Transmission and Reception
       1. Initialization Procedure

Diagram

Description automatically generated

* + - * 1. Initializing the Receiver Section
    - Set the operating mode to configuration mode.
    - Set AVB filtering for network control frames and AVB stream frames to suit the specifications of the product the chip will be used in.
    - Create a descriptor chain for each queue to be used.
    - Set the base address for table address in the descriptor base address table register (DBAT).
    - Specify the maximum frame length with the receive frame length upper limit register (RFLR).
    - Specify whether padding is to be used with the receive padding configuration register (RPC).
    - Set the unread frame counter for each queue with unread frame counter registers (UFCVs) 0 to 4
      * 1. Initializing the Transmitter Section

Diagram

Description automatically generated

* AVB mode: (setting of the TGC.TQP[1:0] bits) being B'01 or B'11
* Non-AVB mode: (setting of the TGC.TQP[1:0] bits) being B'00
  + - * 1. Initializing the E-MAC Section

Diagram

Description automatically generated

* + - * 1. Initialization of the Application Unit

Diagram

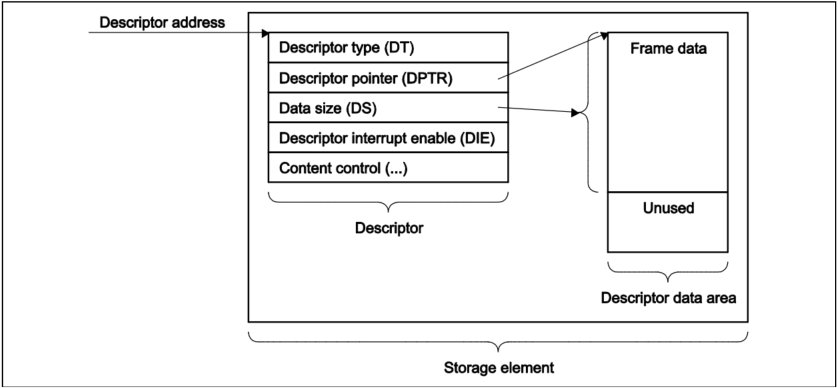
Description automatically generated

* + - * 1. Relationship between Transmission Queue Numbers and Traffic Classes

A picture containing diagram

Description automatically generated

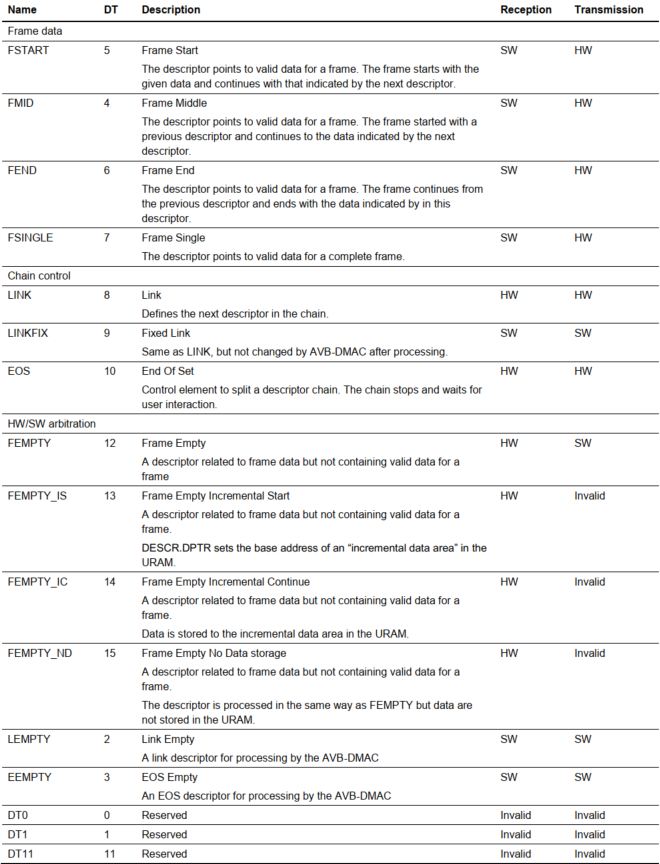
* When the transmit queue priority bits in the transmit configuration register (TGC.TQP) are B'0, the priority order is Q3 → Q 2 → Q1 → Q0.
* When the transmit queue priority bits in the transmit configuration register (TGC.TQP) are B'1, the priority order is Q1→ Q3 → Q2 → Q0
  + - 1. Checking Integrity
         1. Concept of Integrity Checking in Reception
         2. Concept of Integrity Checking in Transmission
         3. Items for Monitoring in Both Reception and Transmission
         4. Items for Monitoring in Reception
         5. Items for Monitoring in Transmission
         6. Items for Monitoring in gPTP
    1. Descriptors
       1. Data Representation in URAM
* The memory in the URAM for use by the AVB-DMAC is configured with control structures referred to as descriptors.
* Example of the memory maps for descriptors and the descriptor data area in the URAM.



* A descriptor consists of its type (DESCR.DT), which controls the descriptor functions
* A descriptor pointer (DESCR.DPTR) indicating the start address for storage of the frame data in the descriptor area.
* The data size field (DESCR.DS), indicating the amount of frame data. Post-processing interrupt generation can be set up for each descriptor.
* Enabling and disabling of the interrupt is controlled by the descriptor interrupt enable bits (DESCR.DIE)
  + - 1. Using Descriptor Chains in Queues
* Descriptor chain: multiple descriptors can be assigned to one queue, combination of multiple descriptors
  + - 1. Descriptor Base Address Table
      2. Descriptor Chain Processing
      3. Descriptor Interrupts

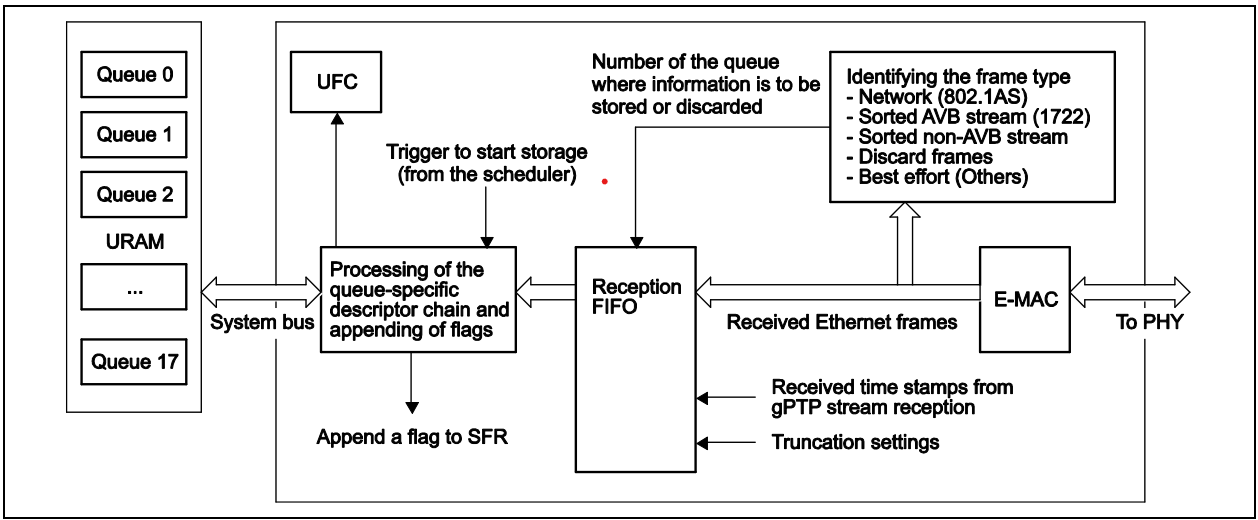
Ethernet AVB-IF provides two kinds of descriptor interrupts:

* Queue specific descriptor interrupt (1 per receive/transmit queue)
* Universal descriptor interrupt (15 shared between all receive/transmit queues)
  + - 1. Descriptor Type

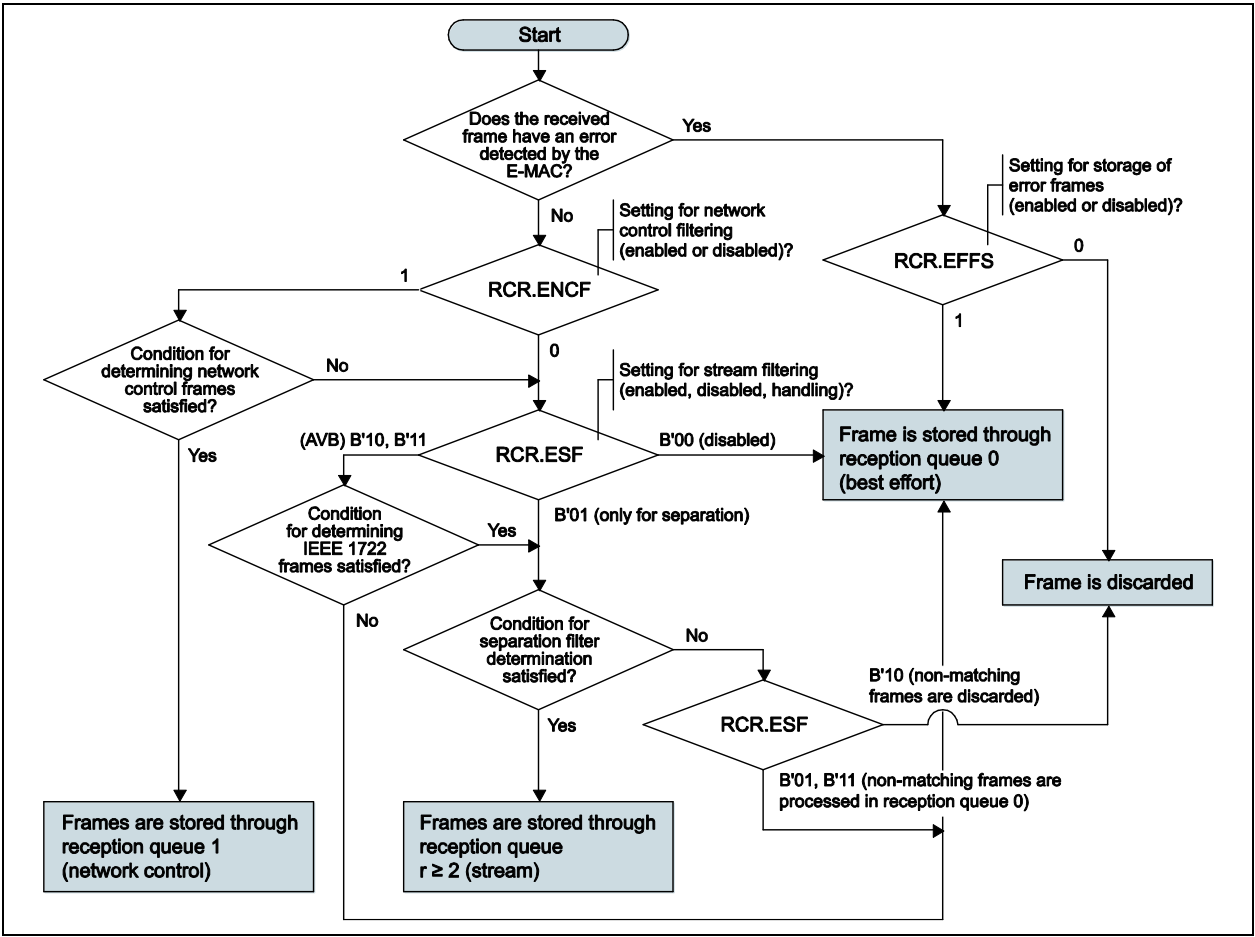


* + - 1. Layout of General Descriptors in the URAM
      2. How to Use Frame Data Descriptors
      3. How to Use Chain Control Descriptors
      4. How to Use Hardware and Software Arbitration Descriptors
      5. Synchronization Between Descriptor Access by Hardware and Software
      6. Tips for Optimizing Performance in Handling Descriptors
    1. Control in Reception

The point of the AVB-DMAC is to transfer data between the E-MAC and URAM without intervention by the CPU

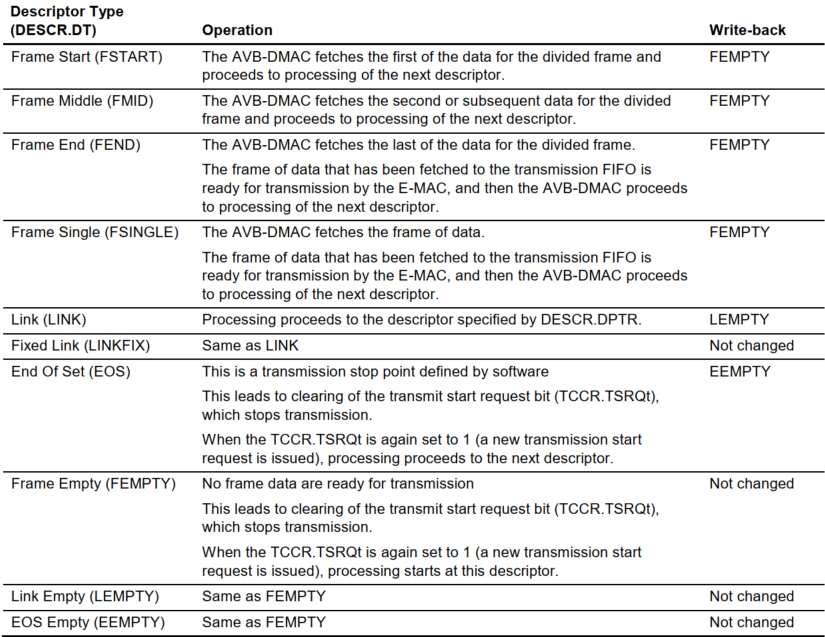


* + - 1. Reception Queues

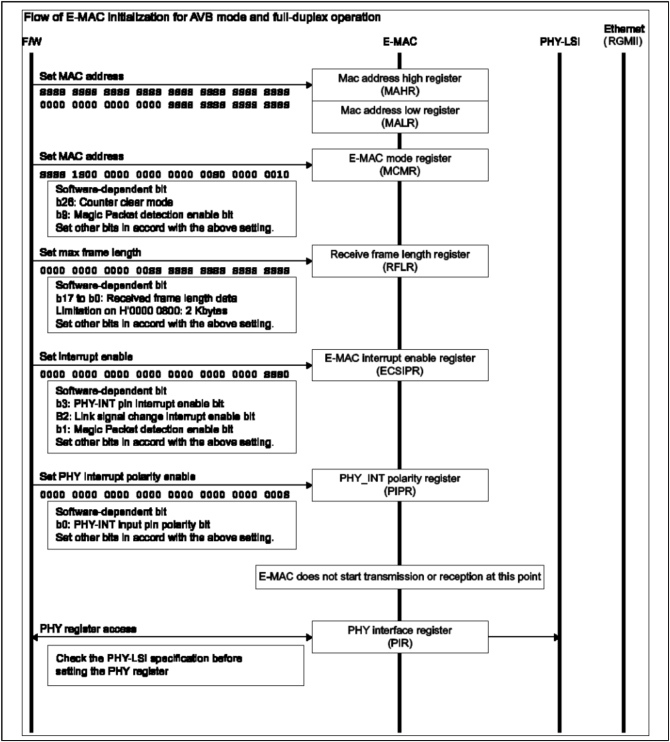


Separation Filtering:

* + - 1. Setting Up Reception Descriptors
    1. Transmission Control
       1. Transmission Modes
          1. AVB Transmission Mode
* Four transmission queues (Q3, Q2, Q1, and Q0) are available.
* Q3 and Q2 are for SR streams (one each for **class A** and **class B**).
* Q1 is for low-bandwidth network control (**NC**) traffic (**gPTP frames**)
* Q0 is for other types of traffic (**MSRPDU\*1**, **MVRPDU\*2**, best effort (**BE**), etc.)
* Notes:
  + MSRPDU: Multiple Stream Registration Protocol Data Unit
  + MVRPDU: Multiple VLAN Registration Protocol Data Unit
    - * 1. Non-AVB Transmission Mode
* The SR class is not supported and the CBS algorithm is not used.
* Data is fetched for transmission in a strict order of priority (Q3 > Q2 > Q1 > Q0)
  + - 1. Setting Up Transmission Descriptors
         1. Transmission Descriptor Type



* + - * 1. Configuration of Transmission Frame Data Descriptors
      1. Transmission
         1. Transmitting Frames
         2. Examples of Descriptor Usage
    1. CBS (Credit-Based Shaping)
    2. Flow Control
    3. Magic Packet Detection
    4. Interrupts
    5. Flows of Operations
       1. Flow of E-MAC Initialization



* + - 1. Flow of AVB-DMAC Initialization

Diagram, schematic

Description automatically generated

* + - 1. Flow for the AVB-DMAC in Reception

Diagram, schematic

Description automatically generated

* + - 1. Flow for the AVB-DMAC in Transmission

Schematic

Description automatically generated with low confidence

* + - 1. Flow for Stopping AVB-DMAC Operation in Reception

Diagram

Description automatically generated

* + - 1. Flow for Stopping AVB-DMAC Operation in Transmission

Table

Description automatically generated

* + - 1. Flow for Stopping and Resetting the AVB-DMAC

Table

Description automatically generated with medium confidence

* + - 1. Flow for Emergency Stopping the AVB-DMAC

A picture containing diagram

Description automatically generated

* + - 1. Flow of gPTP Initialization

Diagram

Description automatically generated with low confidence

* + - 1. Flow of gPTP Time Stamping in Transmission

A picture containing diagram

Description automatically generated

* + - 1. Flow of gPTP Time Stamping and Synchronization in Reception

Diagram

Description automatically generated

* + - 1. Flow of Capturing gPTP Presentation Times

Diagram

Description automatically generated with medium confidence

* + - 1. Flow of AVTP Presentation Time Comparison

Diagram

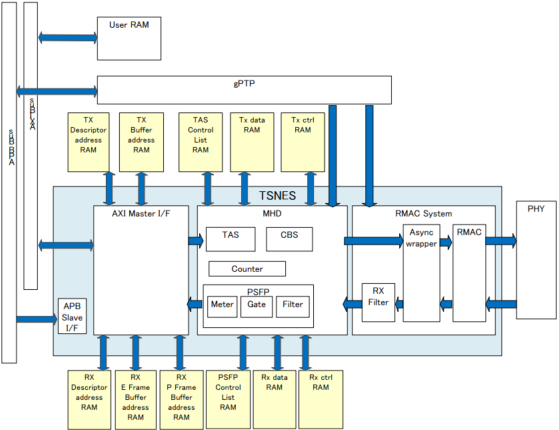
Description automatically generated

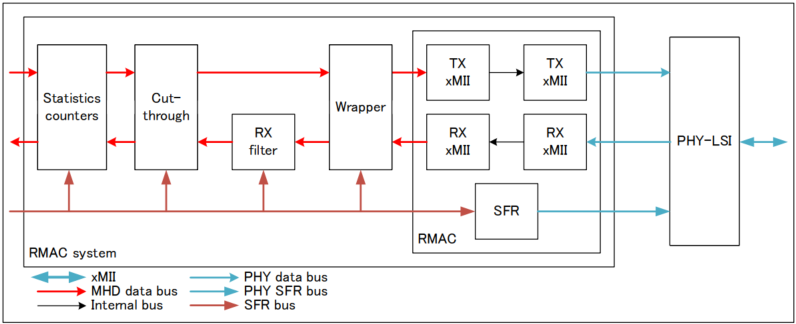
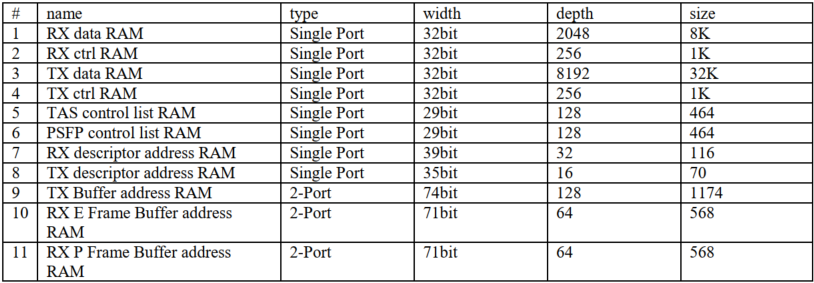
* + - 1. Flow of Loopback Mode Operation

A picture containing diagram

Description automatically generated

1. Ethernet TSN
   1. Block Diagram



* User RAM: the local RAM
* **gPTP** (Generalized Precision Time Protocol): As a form of the PTP protocol, gPTP is used to synchronize time over TSN Ethernet networks. gPTP helps ensure higher time accuracy, suitable for applications that require it.
  + **PTP** (Precision Time Protocol) is a network protocol used to synchronize the clocks of devices on a network. It is designed to provide accurate and precise time synchronization, particularly for industrial control systems, multimedia applications, and other scenarios where timing is critical. PTP uses a master-slave architecture, where one device acts as the master and other devices act as slaves, and exchanges timing information to synchronize their clocks. PTP can operate over Ethernet and other network protocols, and supports both unicast and multicast messaging. It is commonly used in applications such as audio and video production, industrial automation, and telecommunications
* PHY(Ethernet physical layer): is the layer of the Ethernet network that ensures the transmission of signals between devices on the network.
* **TSNES** (Time-Sensitive Networking Endpoint Subsystem): the communication control block on the TSN Ethernet network, responsible for managing data packets and other functions related to time synchronization on the TSN Ethernet network. TSNES includes components such as APB Slave I/F, AXI Master I/F, MHD, RMAC System to connect with other devices on the TSN Ethernet network.
  + **APB Slave I/F** (Advanced Peripheral Bus Slave Interface): This is the interface that connects TSNES to other peripheral devices, facilitating the transmission of data and control commands between them.
  + **AXI Master I/F** (Advanced eXtensible Interface Master Interface): This is the interface that connects TSNES to other devices on the TSN Ethernet network, facilitating the transmission of data and control commands between them.
  + **MHD** (Media Independent Interface with Data Interface): provides a communication interface between PHY and TSNES through the MII or RMII protocol. It is used to transmit data between the physical layer and other functional elements of the system.
  + **RMAC System** (Renesas MAC System): This IP is an Ethernet controller that conforms to the definition of the MAC (Media Access Control) layer for Ethernet in the IEEE 802.3 standard.
    - 
      * RX filtering: Frame filtering
        + Rejection of unknown packets for security purposes
        + Selection of CPU sub destination
      * Cut- through: MAC to MAC data forwarding depending on MAC destination address
        + E-Frame cut-through
        + P-frame cut-through
        + TAS based cut-through
      * Wrapper: Asynchronous modules for synchronization between PHY clocks and bus clock
      * Statistics counters: Counter aiming at monitoring the system.
* 
  1. Register
     1. GPOUT
     2. AXIBMI
     3. MHD
     4. RMAC
  2. Operation