EMBEDDED SYSTEMS

BASED ON CORTEX-M4 AND THE RENESAS SYNERGY PLATFORM

2020
PROF. DOUGLAS RENAUX, PHD
PROF. ROBSON LINHARES, DR.
UTFPR / ESYSTECH

RENESAS ELECTRONICS CORPORATION

11 – ETHERNET

- Introduction
- Block Diagram
- Registers
- SW Stack

11.1 – INTRODUCTION

Ethernet is a wired network technology used to interconnect devices in a Local Area Network (LAN). It has been standardized as IEEE802.3, comprising the physical and data link layers of the OSI model.

Main characteristics:

- Packet-based protocol,
- All messages are broadcast and processed by the nodes only if needed,
- Nodes can transmit at any time → Ethernet provides for automatic collision management (electrical or logical),
- Up to 10 Gbps (10GBASE-SR and further).

11.1 – INTRODUCTION

Ethernet standard has evolved across different versions:

- Ethernet (IEEE 802.3) standard → makes use of coaxial connections (named 10BASE2 and 10BASE5) to achieve bandwidths up to 10 Mbps;
- Fast Ethernet (IEEE 802.3u) → evolved from 10BASE-T (4-pair unshielded twisted pair) to 100BASE-TX and 100BASE-FX (fiber-optic cable) to achieve bandwidths up to 100 Mbps;
- Gigabit Ethernet (IEEE 802.3z) → variation of Fast Ethernet (1000BASE-T); that supports full duplex operation to provide higher data rates (up to 1 Gbps);
- 10 Gigabit Ethernet (IEEE 802.3ae) → entirely based on optical fiber (10GBASE-SR), up to 10 Gbps.

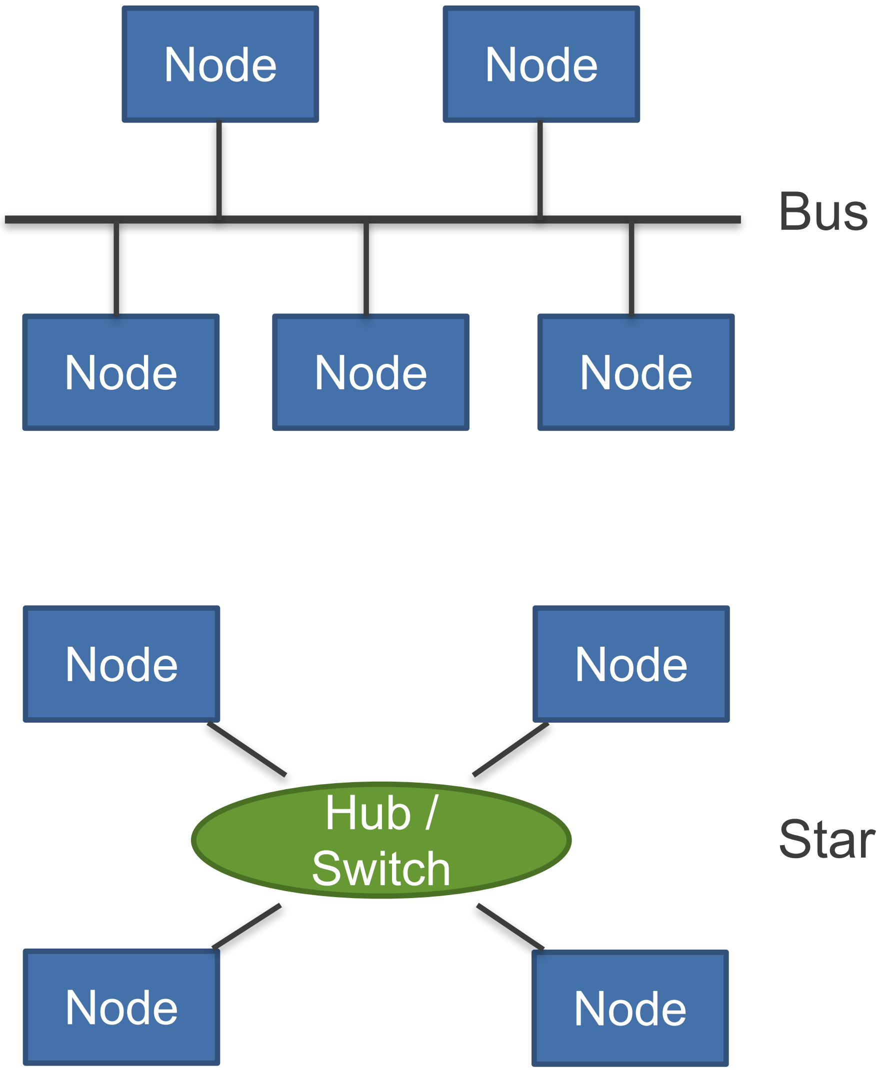
ETHERNET TOPOLOGY

Earlier Ethernet versions (10BASE2 and 10BASE5) used coaxial cables to interconnect nodes in a physical and logical bus topology.

Later versions (10BASE-T) rely on a physical star topology based on **hubs** → connected to nodes with twisted pair cabling.

Current versions (100BASE-TX and further) rely on a physical star topology based on **switches** → physically isolate the nodes so that a packet is delivered solely to its destination node → minimizes the network congestion due to packet collision.

Ethernet is always a bus topology from the logical point-of-view.



source: Authors

ETHERNET CONNECTION INFRASTRUCTURE

A Hub acts as a **repeater**:

- Data received by the hub from an Ethernet node is sent to all other Ethernet nodes connected to the hub.
- Therefore, multiple simultaneous transmissions are mixed and equally propagated to all connected nodes → possibility of collisions.

A Switch acts as a **filtered repeater**:

Destination address of every transmitted Ethernet packet (frame) is checked by the switch.

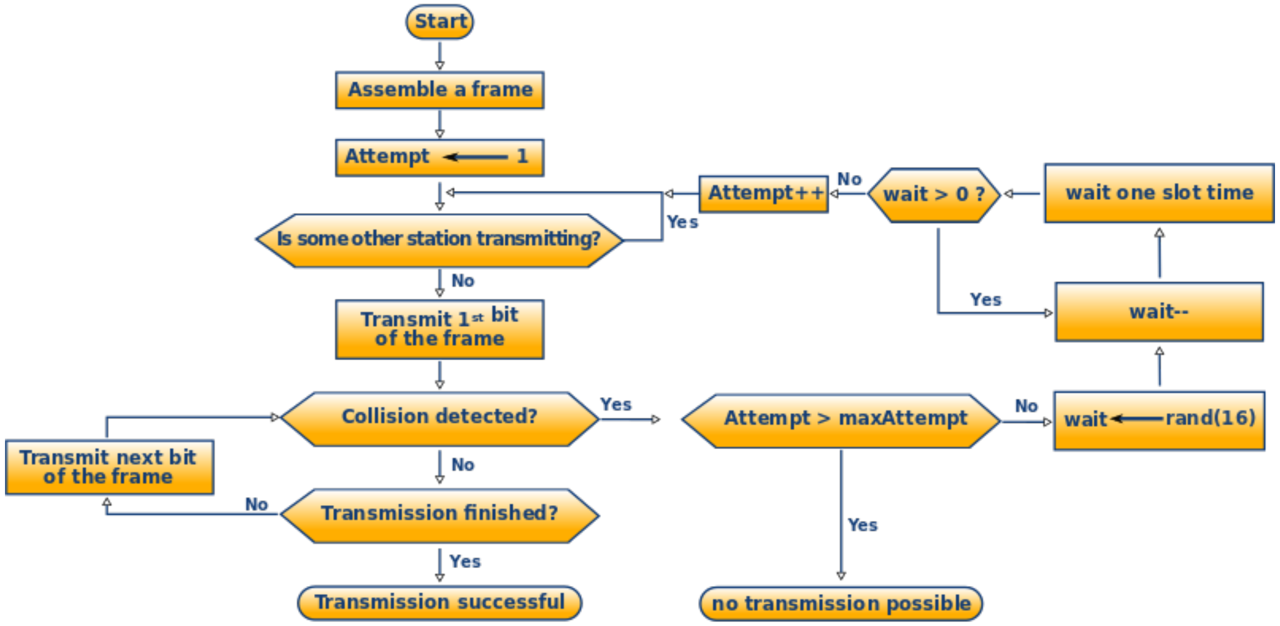
- The frame is forwarded only to the corresponding Ethernet node.
- This allows multiple simultaneous transmissions to succeed, provided that the pair of source-destination nodes for each of the transmissions is different.
- Packet collisions are avoided, as the switch is able to enqueue and serialize multiple frames addressed to the same destination node.

ETHERNET COLLISION MANAGEMENT

Because multiple Ethernet nodes share a logical bus, it is possible that more than one node try to transmit at the same time → collision.

To manage collisions, Ethernet uses the **Carrier Sense Multiple Access / Collision Detection (CSMA/CD)** protocol.

- The sender node starts transmitting a packet (frame) and uses carrier-sensing to detect if other nodes are trying to transmit at the same time.
- While no collision is detected, the sender node keeps on sending the frame bits until the end.
- If a collision is detected, a jam signal is sent to warn the other nodes about the collision, a retransmission counter is incremented, and the frame transmission is restarted after a random amount of time.
- If the retransmission counter reaches the maximum number of attempts, the transmission is aborted.



Source: https://upload.wikimedia.org/wikipedia/commons/3/37/CSMACD-Algorithm.svg

ETHERNET PACKET

An Ethernet frame encapsulates the data packet.

The frame includes addressing information (MAC) and error detection features.

- CRC checking is performed over all fields (except Preamble and SFD) and compared to FCS.

Dest addr defines special values for broadcast (all nodes receive and process the packet) and multicast (a group of nodes receives and processes the packet).

Field	Preamble	Start of Frame Delimiter	Dest MAC addr	Src MAC addr	Length/ type	Data	Frame Check Seq
Bytes	8	1	6	6	2	46 to 1500	4

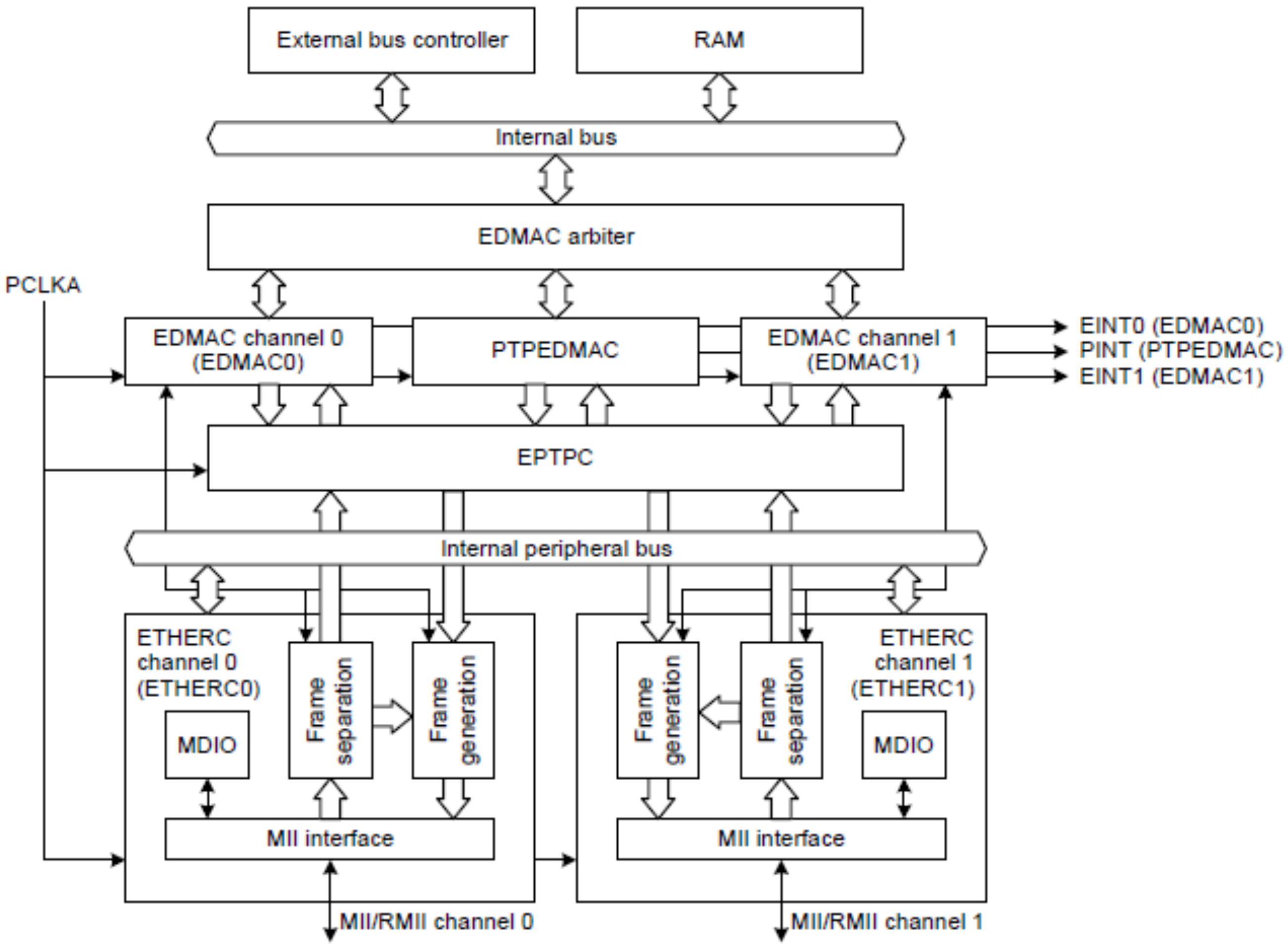
802.3 packet

11.2 – BLOCK DIAGRAM – CASE STUDY

Implementation for the Ethernet Controller Module of the R7FS7G27H3A01CFC Renesas ARM Cortex-M4 MCU.

- Two-channel controller → can operate two independent ETH interfaces.
- Depends on a DMA controller (EDMAC) to handle the TX and RX buffers without CPU intervention.

MII (Media Independent Interface) and RMII (Reduced MMI) are used to connect the ETH controller to the PHY hardware that implements the electrical interface.



Source: Renesas Synergy MCUs User's Manual: Hardware

11.3 – REGISTERS – CASE STUDY

Implementation for the Ethernet Controller Module of the R7FS7G27H3A01CFC Renesas ARM Cortex-M4 MCU:

- ECMR → enable / disable, operation mode configuration
- RFLR → maximum frame length (between 1518 and 2048 bytes)
- ECSR → detection of line events (e. g. false carrier)
- ECSIPR → enable/disable line events interrupt
- PIR → access PHY registers
- PSR → status of PHY
- RDMLR → upper limit for random number generation
- IPGR → sets the interpacket gap (in bit times)

11.3 – REGISTERS – CASE STUDY

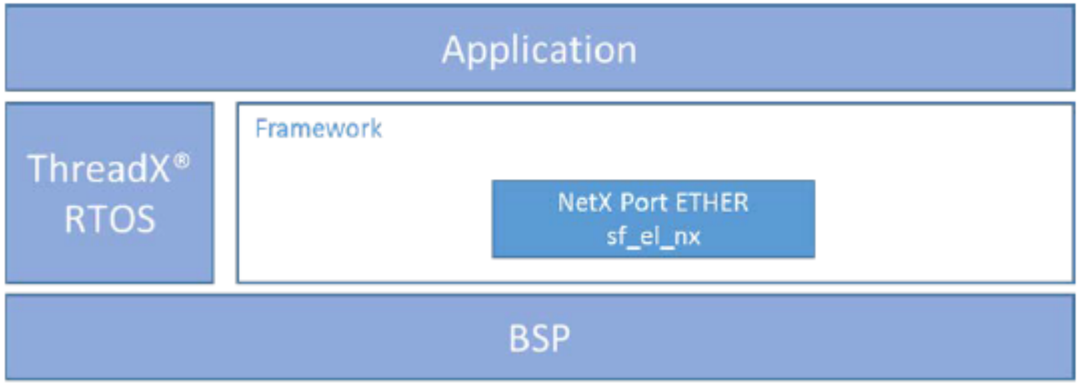
- APR → set pause time for an automatic PAUSE frame (used for flow control)
- MPR → set pause time for a manual PAUSE frame (used for flow control)
- RFCF → number of received PAUSE frames
- TPAUSER → max number of PAUSE frame retransmission
- TPAUSECR → PAUSE retransmit counter
- BCFRR → max number of received broadcast frames
- MAHR → upper bits of MAC address
- MALR → lower bits of MAC address
- TROCR → number of frames that failed to be retransmitted

11.3 – REGISTERS – CASE STUDY

- CDCR → number of late collisions detected
- LCCR → number of losses of carrier detected
- CNDCR → number of times a carrier is not detected
- CEFCR → number of received frames with CRC error
- FRECR → number of times a frame receive error has occurred
- TSFRCR → number of short frames received
- TLFRCR → number of long frames (longer than RFLR value) received
- RFCR → number of frames received with alignment error (not integral number of octets)
- MAFCR → number of multicast frames received

11.4 – SOFTWARE STACK – CASE STUDY

- Example of Ethernet stack for Renesas microcontroller hardware (part of the SSP → Synergy Software Package).
- The Ethernet layer depends on upper layers to manage the network protocols that generate or consume the data encapsulated into Ethernet packets → the “Application” layer.
- (https://www.renesas.com/en-us/software/D6001601.html)



Source: Renesas Synergy NetX Port Module Guide
r11an0218eu0101-synergy-sf-el-nx-mod-guide

11.4 – SOFTWARE STACK – CASE STUDY

- Example of Ethernet API for Renesas microcontroller hardware à part of NetX Framework for Renesas Synergy Software Package (SSP).

void	edmac_eint_isr (void) edmac_eint_isr More...
UINT	nx_synergy_ethernet_init (NX_REC *nx_rec_ptr, sf_el_nx_cfg_t *sf_el_nx_cfg_ptr, bool hw_padding) nx_synergy_ethernet_init More...
void	nx_driver_event_handler (NX_REC *nx_rec_ptr) nx_driver_event_handler More...
void	enet_hw_enable_interrupt (NX_REC *nx_rec_ptr) enet_hw_enable_interrupt More...
UINT	nx_synergy_ethernet_deinit (NX_REC *nx_rec_ptr, sf_el_nx_cfg_t *sf_el_nx_cfg_ptr) nx_synergy_ethernet_deinit More...
ssp_err_t	nx_ether_custom_packet_send (NX_PACKET_POOL *pool_ptr, NX_REC *nx_record_ptr, UCHAR *data, UINT length, USHORT ether_type, nx_mac_address_t dest_mac_address) nx_ether_custom_packet_send More...

	sf_el_nx_cfg_t *sf_el_nx_cfg_ptr) nx_ether_driver More...
void	nx_ether_interrupt (NX_REC *nx_rec_ptr) nx_ether_interrupt More...
ssp_err_t	nx_ethernet_version_get (ssp_version_t *const p_version) Retrieve the API version number. More...

Basic comm API functions

Renesas Synergy Software Package v1.7.5 User's Manual
r11um0140eu0106-synergy-ssp-v175

Renesas.com