CAN RECEIVER

Adrian Gongora
Angel Sanchez

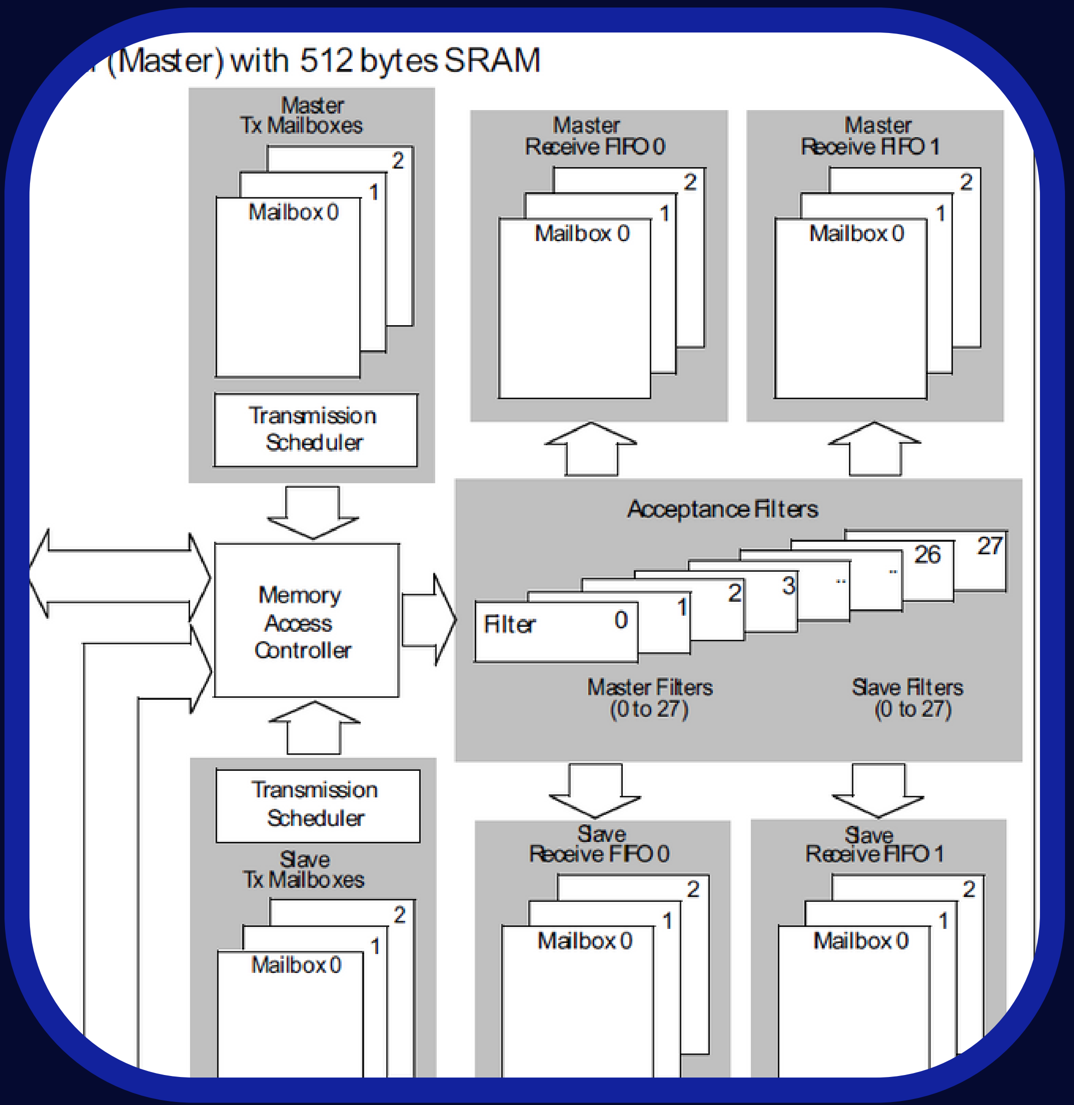
FEATURES

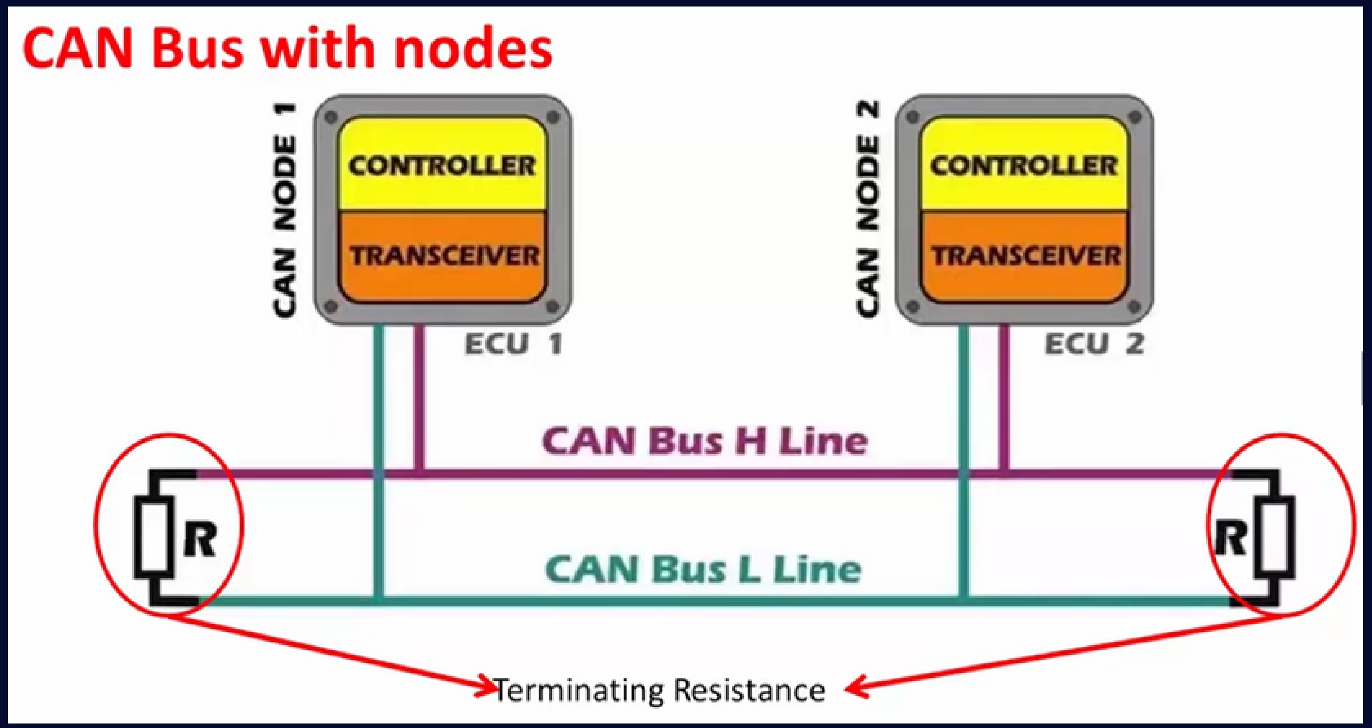
- 2 recieve FIFOs with 3 stages
- Scalable filter banks:
  - 28 filter bakns shared between CAN1 and CAN2
- Identifier list feature
- Configurable FIFO overrun
- Time Stamp on OF reception

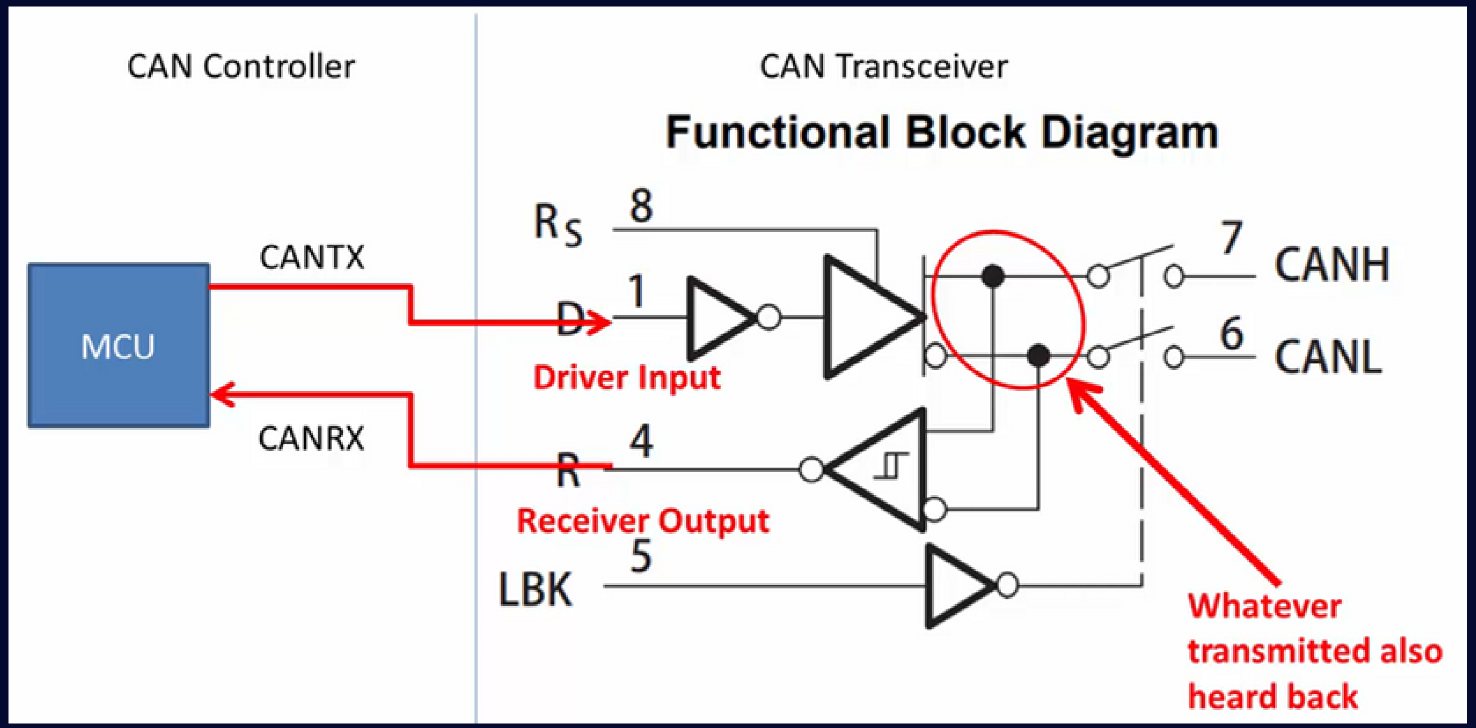
RX PATH

To understand reception we need to know the **PATH** that a message takes from the **BUS** to the **MCU.**

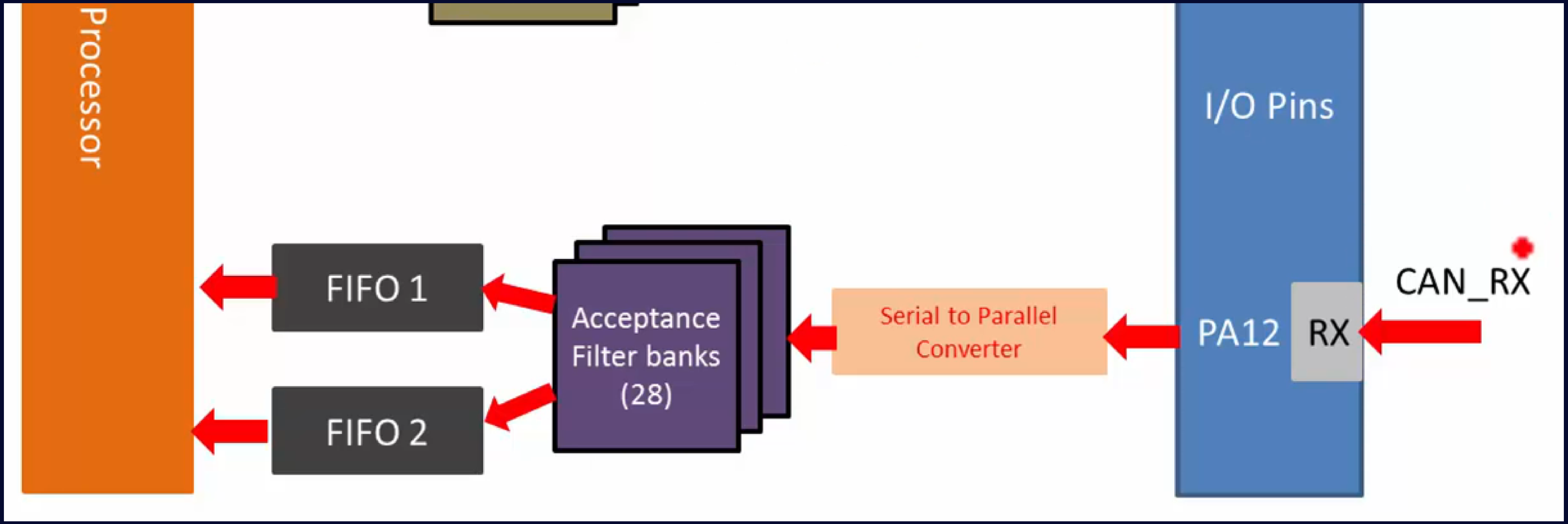
CAN BUS -> TRANSRECIVER -> CAN CONTROLLER (bxCAN)







The bxCAN RX PATH - Filtering

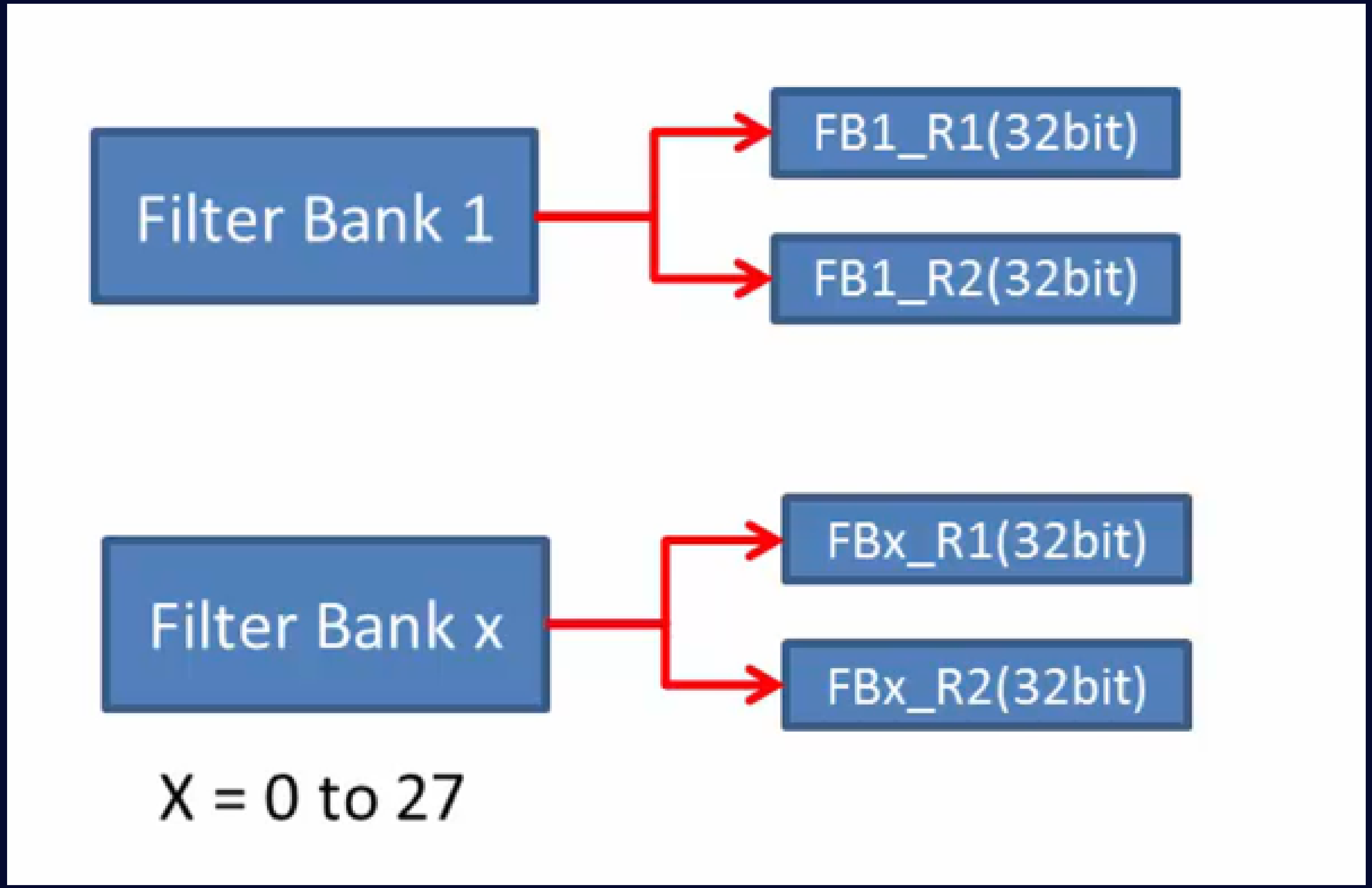


Filter banks is a hardware mechanism that can be programmed to ignore and discard certain CAN frames and allow other to pass on to FIFOs.

There are 28 filter banks for CAN1&2 each with 2 32-bit registers

Filtering Banks are also used to prevent saturation of the CPU

The bxCAN RX PATH - Filtering



There are 28 filter banks for CAN1&2 each with ***two*** *32-bit registers*

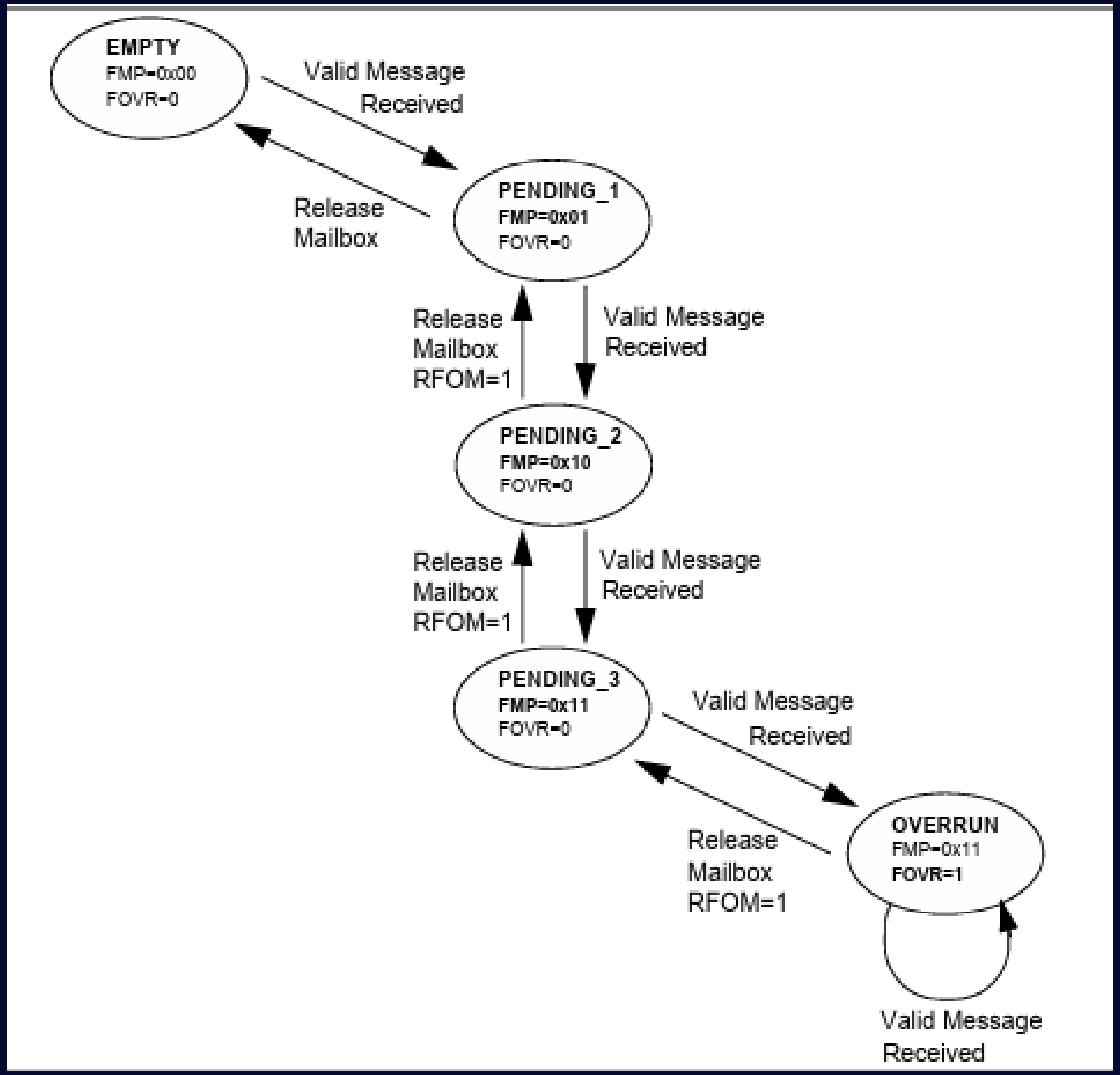
Examples:

- *Accept only extented id frames*
- *Accept only indentifiers with hex value 0x02*

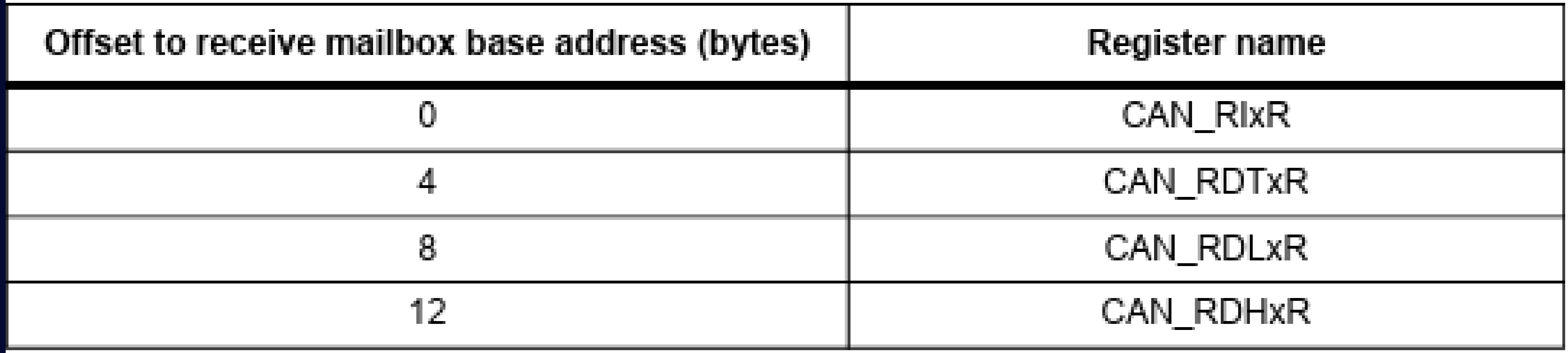
FIFO

- For recieving we have 2 FIFOs (First in-First Out), these are used to store incomming messages.
- 3 complete messages can be stored at each
- Managed completely by hardware.

- We can say that a message is valid when recieved according to the CAN protocol.
- The application accesses the messages stored in the FIFO throught the FIFO output mailbox.
- The important bit is the acknowledge field

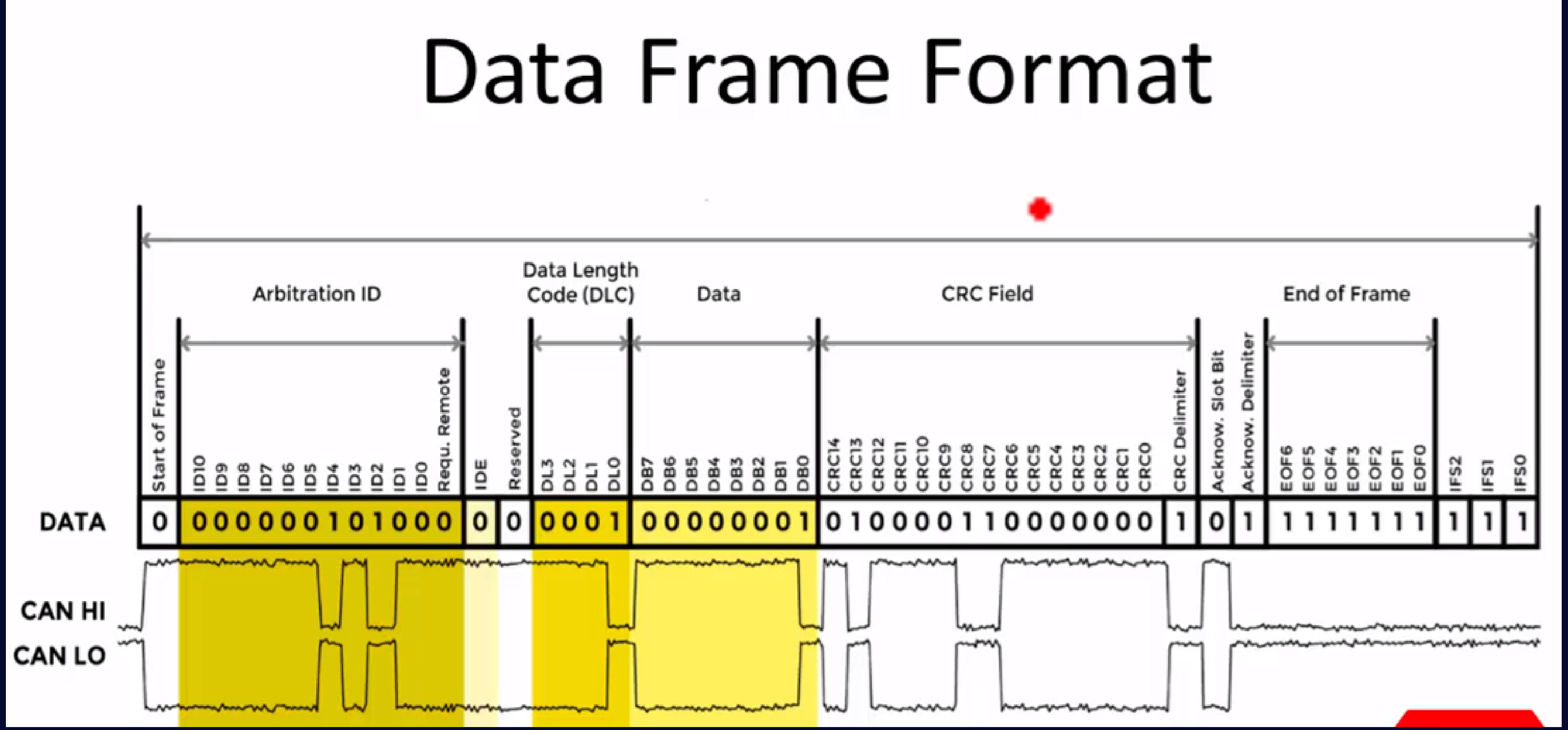


- Reception Handling is more complex than the Transmission, we have the 3 mailboxes organized by the FIFO and the Acceptance filters.
- The acknowledge field is controlled by the recieving node, the state of this field will tell us the outcome.



Offset to receive mailbox base address (bytes)	Register name
0	CAN_RIxR
4	CAN_RDTxR
8	CAN_RDLxR
12	CAN_RDHxR

- When a message is stored, it becomes available in the FIFO output mailbox.
- Once the software handle the message, the software must release the FOM, activating the CAN_RFR Reg to make the incoming message available.



THANK YOU