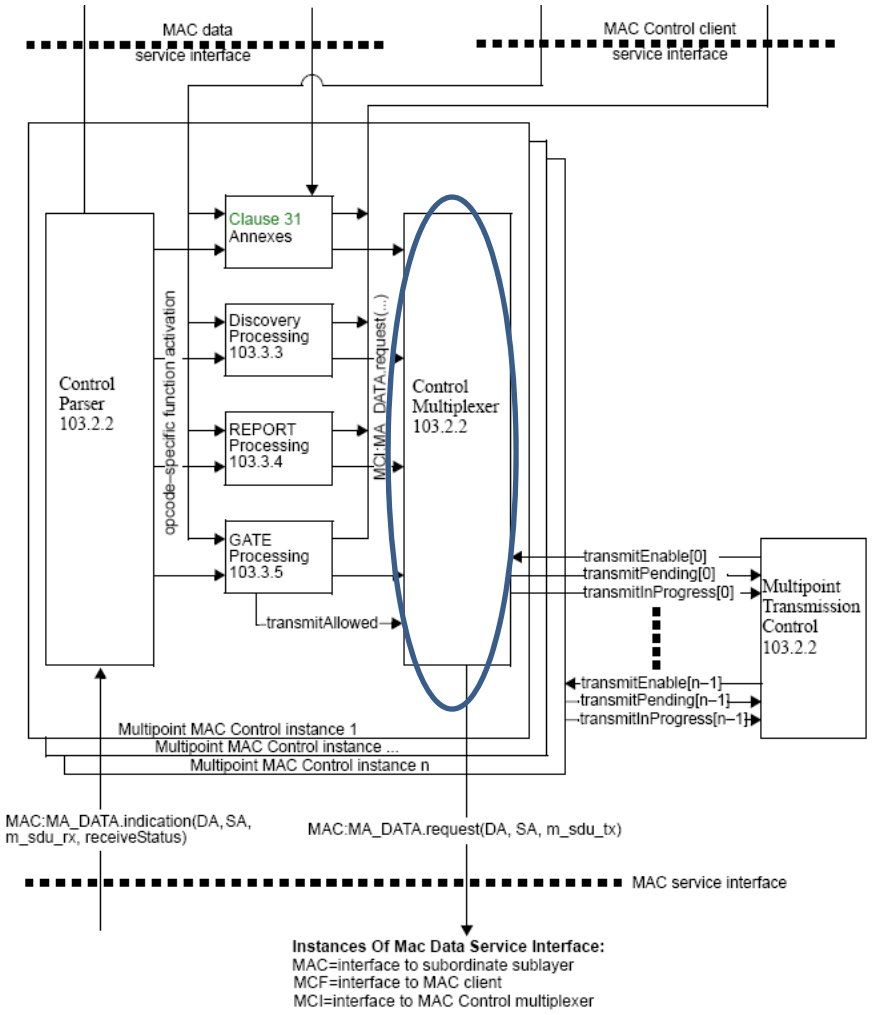
Control Multiplexer, Idle Deletion, Data Detector, and GearBox for Downstream CLT Transmission

Jin Zhang, Beiyu Rong

Marvell

Control Multiplexer in MPCP



Functions of Control Multiplexer

- Common for CLT and CNU control multiplexer
  - Select which data unit to send
    - For example, always control unit before data unit.
    - Implementation dependent
  - Force idle period after each data unit for:
    - Insertion of FEC parity
    - Insertion blank period to adapt to PMD rate.
- CNU only:
  - Identify gap of idles and decide formation of bursts
  - Check if the remaining time is enough for the incoming MAC frame

Current CLT Control Multiplexer

Introduce transmitAllowed for TDD mode

No grant size check for CLT

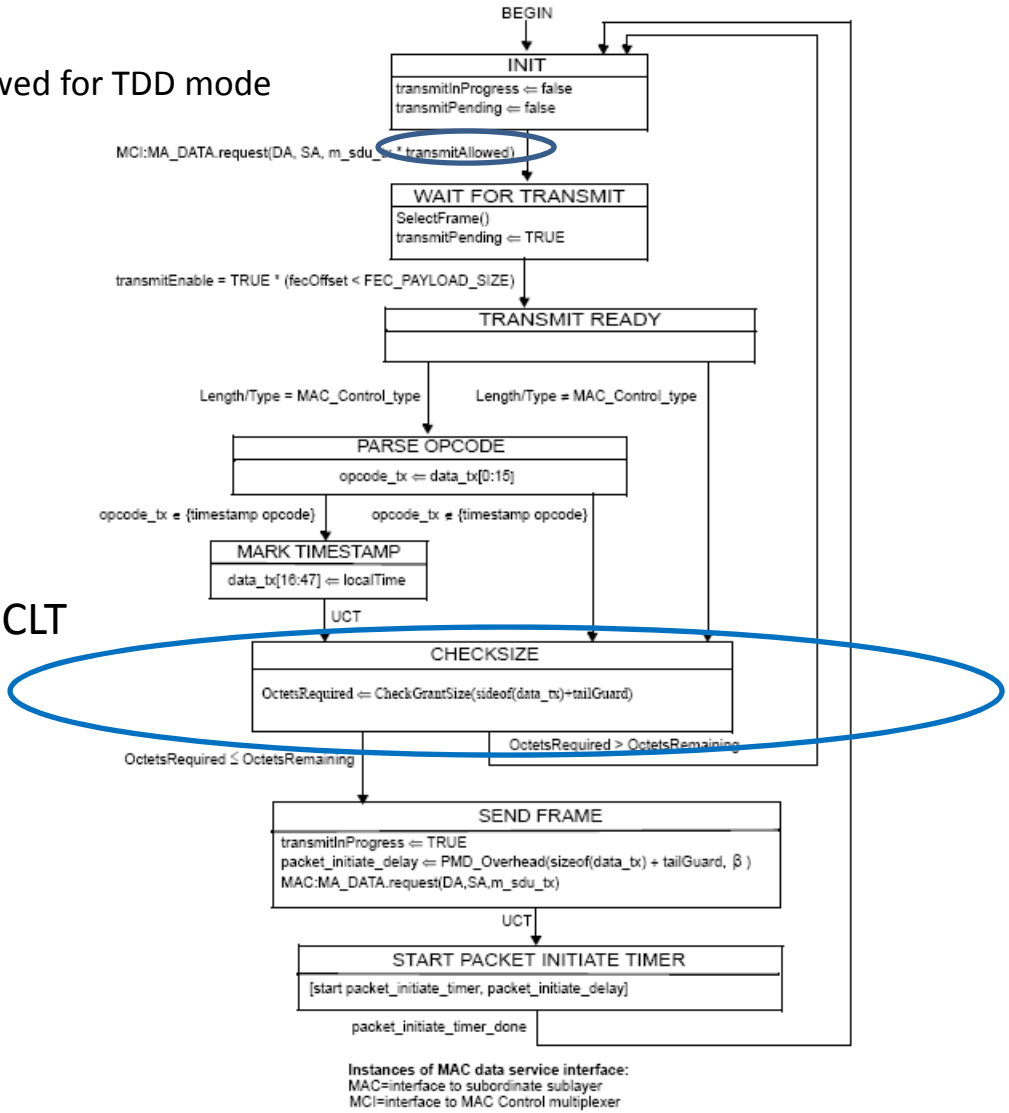


Figure 103–12—CLT Control Multiplexer state diagram

Current PMD Overhead Function

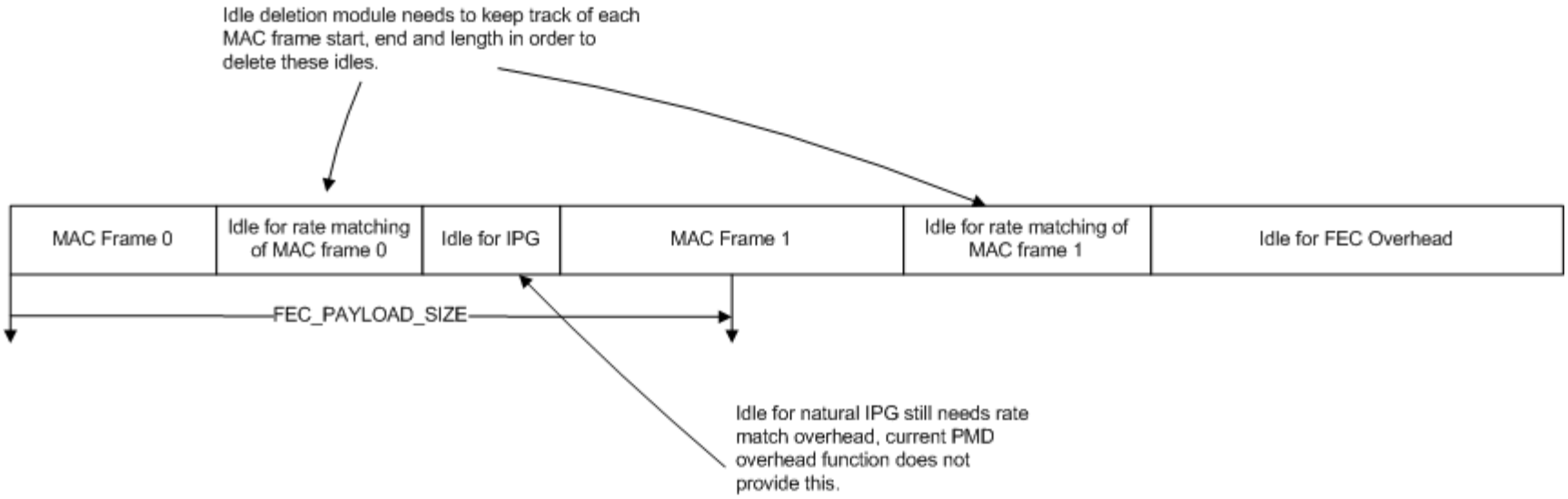
$$PMD\_Overhead((length), \beta) =$$

$$\left\lceil (\beta - 1) \times length + \beta \times (FEC\_PARITY\_SIZE \times \left\lfloor \frac{fecOffset + length}{FEC\_PAYLOAD\_SIZE} \right\rfloor) \right\rceil$$

$$\beta = \frac{XGMII\_Rate}{PCS\_Rate} = \frac{PhyInDataSize}{PhyOutDataSize}$$

- Current PMD overhead function does not work with Idle deletion.
- Idle deletion needs to keep track of each MAC frame, including start, end and length, leading to complex state machine
- Examples next slide

Problem of Current PMD Overhead Function

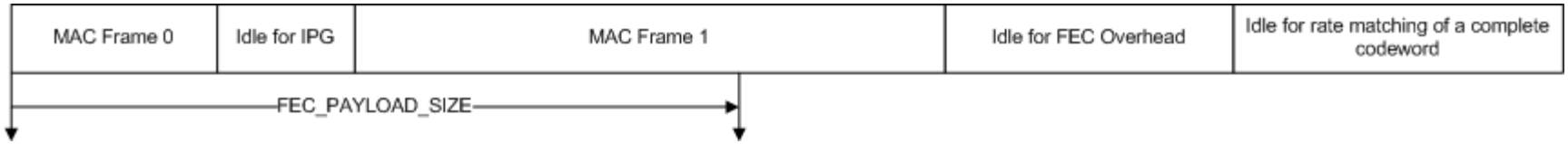


Modify PMD Overhead Function

$$PMD\_Overhead(length, \beta)$$
$$= 12 + FEC\_PARITY\_SIZE \times \left\lfloor \frac{fecOffset + length}{FEC\_PAYLOAD\_SIZE} \right\rfloor$$
$$+ \left\lceil (\beta - 1) \times FEC\_CODEWORD\_SIZE \times \left\lfloor \frac{fecOffset + length}{FEC\_PAYLOAD\_SIZE} \right\rfloor \right\rceil$$

- Rate matching overhead only shows up after a complete codeword.
- Idle deletion module does not need to track the start, end and length of each MAC frame.
- Idle of natural frame gap is also considered in the rate matching overhead.

Example of Modified PMD Overhead Function

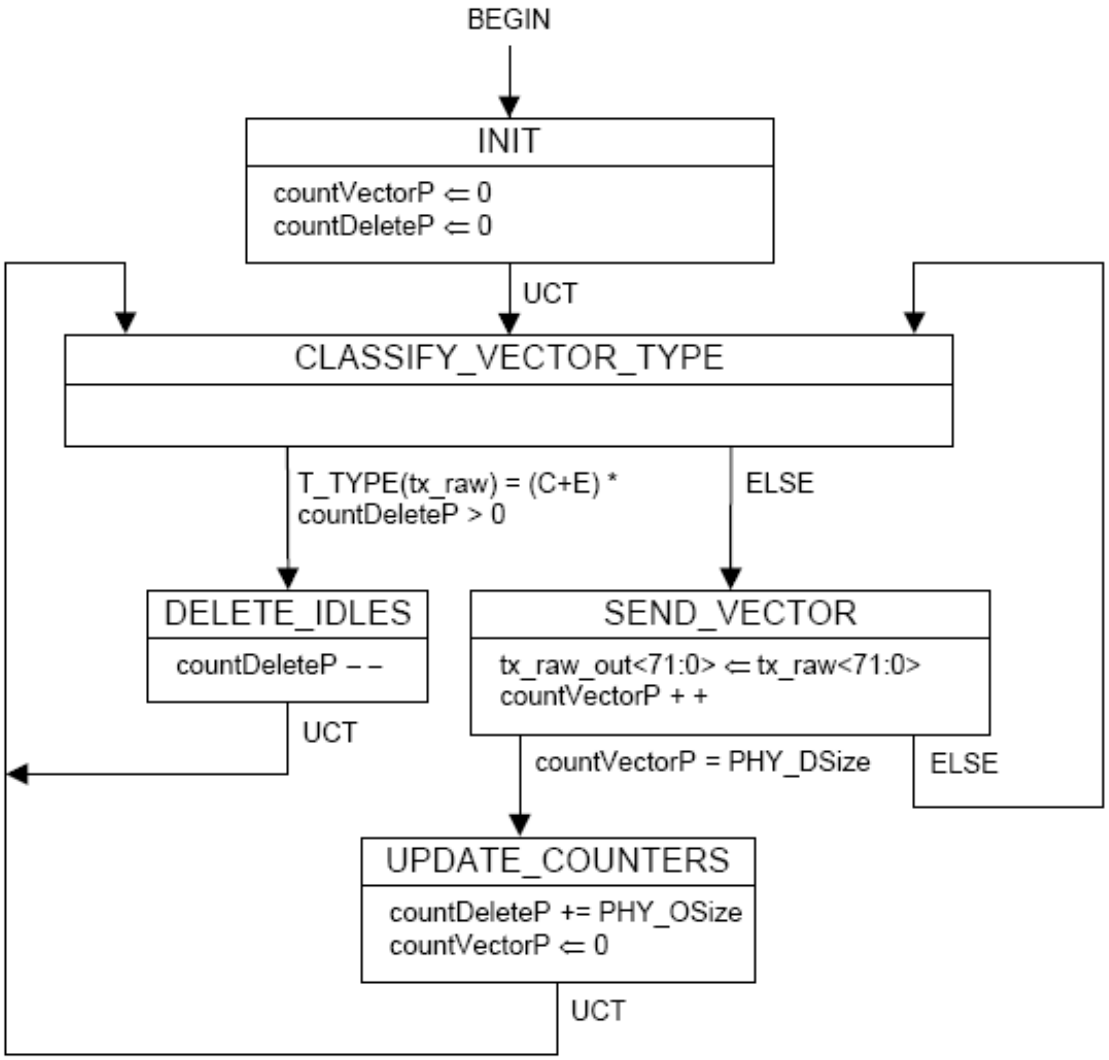


FEC Overhead is a fixed number, so is the rate matching overhead

Downstream Idle Deletion

- Only delete 72-bit idle vectors inserted from control multiplexer.
  - Delete idle vectors for FEC overhead
  - Delete idle vectors for rate adaptation overhead
- Should not delete idle vectors for inter-packet gap.
- In the long run, should supply exact amount of vectors for PMD rate.

Current Idle Deletion Diagram



Current Idle Deletion Diagram

PHY_DSize

TYPE: 16-bit unsigned integer

The number of 72-bit vectors constituting (together with PHY_OSize) the denominator in the EPoC PCS de-rating Equation (1). To normalize the effective PCS data rate, the Idle control character deletion process removes PHY_OSize vectors per every PHY_DSize vectors transferred to the FEC overhead compensation sub-process.

Value: TBD

$$PCS\_Rate = XGMII\_Rate \times \frac{PHY\_DSize}{PHY\_DSize + PHY\_OSize} \quad (1)$$

PHY_OSize

TYPE: 16-bit unsigned integer

The number of 72-bit vectors constituting the numerator in the EPoC PCS de-rating Equation (1). To normalize the effective PCS data rate, the Idle control character deletion process removes PHY_OSize vectors per every PHY_DSize vectors transferred to the FEC overhead compensation sub-process.

Value: TBD

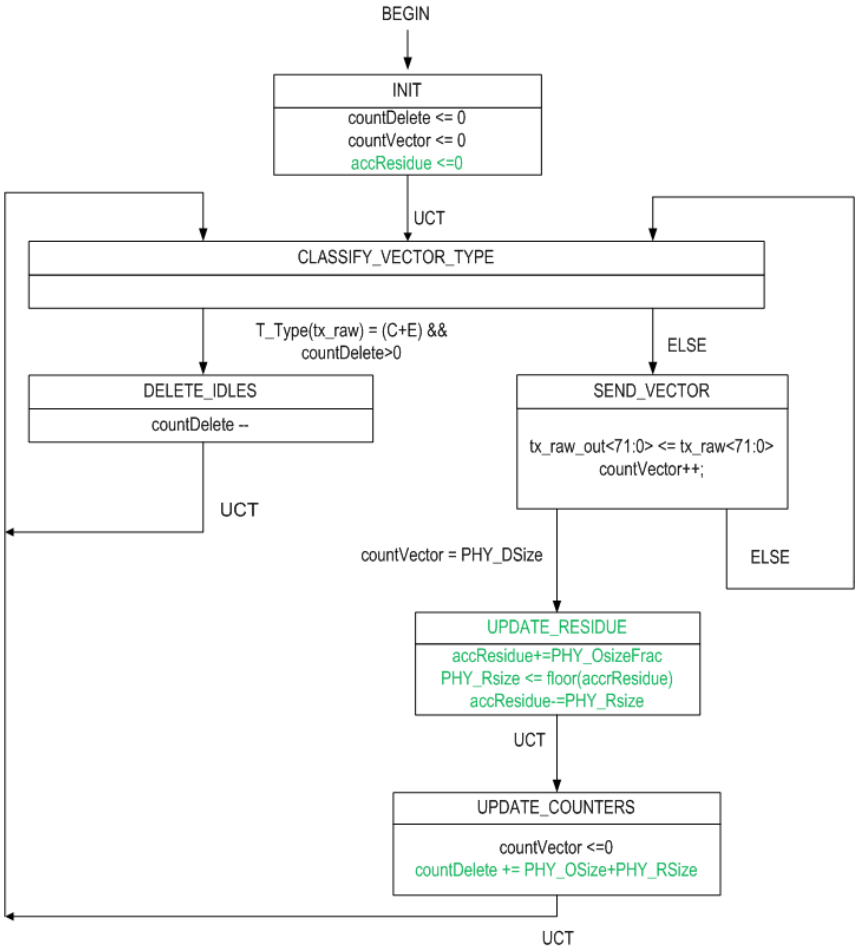
Problem

- Idles for FEC overhead and de-rating overhead are deleted in separated process.
  - Can combine the two processes into single one
- In general, PHY_OSize is a fractional number.
  - PCS_Rate = PMD_Rate*64/65;
  - PMD_Rate = PlcTotalBits/PlcTotalCycles*F_{PMD}

$$\text{PHY\_OSize} = \frac{\text{XGMII\_Rate} - \text{PCS\_Rate}}{\text{PCS\_Rate}} \times \text{PHY\_DSize}$$

  - Should use an accumulator to track residue.

Modified Idle Deletion



- accResidue: a fractional number to accumulate the residue.
- PHY_RSize: binary number, 1 or 0
- PHY_DSize: integer 220
- FEC_OSize: integer 29
- PHY_OSize:

$$\mathrm{PHY\_OSize} = \left\lfloor \frac{\mathrm{XGMII\_Rate} - \mathrm{PCS\_Rate}}{\mathrm{PCS\_Rate}} \times (\mathrm{PHY\_DSize} + \mathrm{FEC\_OSize}) \right\rfloor + \mathrm{FEC\_OSize}$$

- PHY_OsizeFrac: fractional part of the actual PHY_Osize. Its precision is implementation dependent.

$$\mathrm{PHY\_OSizeFrac} = \frac{\mathrm{XGMII\_Rate} - \mathrm{PCS\_Rate}}{\mathrm{PCS\_Rate}} \times (\mathrm{PHY\_DSize} + \mathrm{FEC\_OSize}) + \mathrm{FEC\_OSize} - \mathrm{PHY\_OSize}$$

$$\mathrm{PCS\_Rate} = \mathrm{PMD\_Rate} \times \frac{64}{65}$$

$$\mathrm{PMD\_Rate} = \frac{\mathrm{PLCTotalBits}}{\mathrm{PLCTotalCycles}} \times \mathrm{OFDM\_SampleFreq}$$

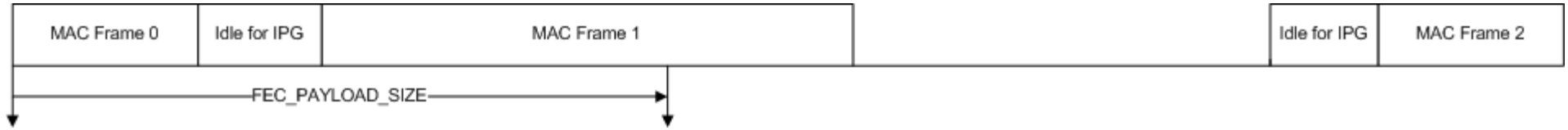
CLT Data Detector

- Downstream CLT data detector is mainly used to control FEC encoder and insert parity.
- The current data detector input and output process can meet the requirement.

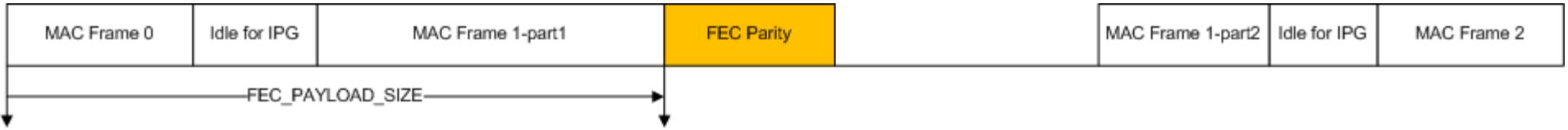
Example

@Control Multiplexer

@Idle Deletion



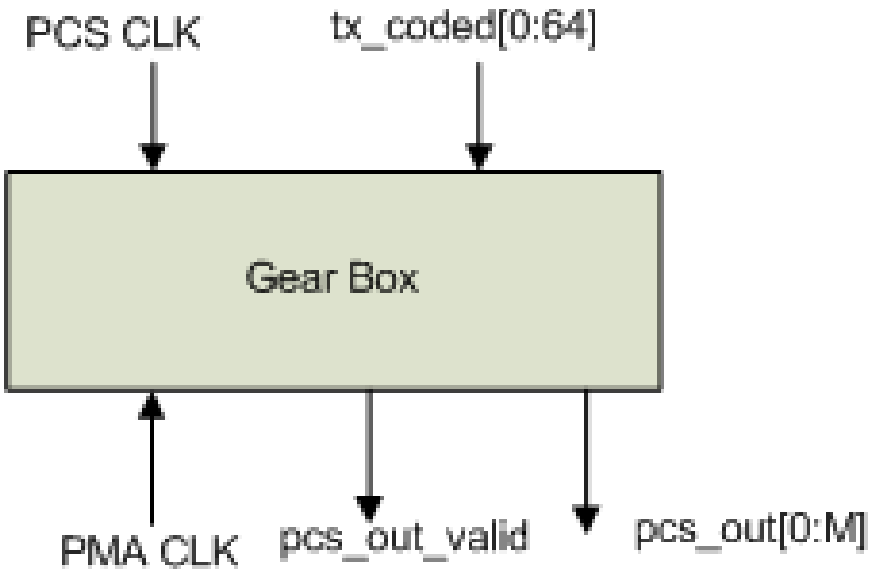
@data detector



Gear Box

- Data detector output can be directly used by PMA
  - Store data detector output in a buffer.
  - PMA fetch data whenever needed.
- Benefit of defining an optional Gear Box layer in EPoC
  - Standard boundary interface between PCS and PMA
  - Convenience for operation and management. Operator can monitor and probe the data through interface.

Gear Box Input and Output



Gear Box Output Process