

# Idle Deletion / Start Alignment / Error Checking State Machine

---

**Eric Lynskey, Teknovus (author)**

**Frank Effenberger, Huawei (supporter)**

# Problem Statement

---

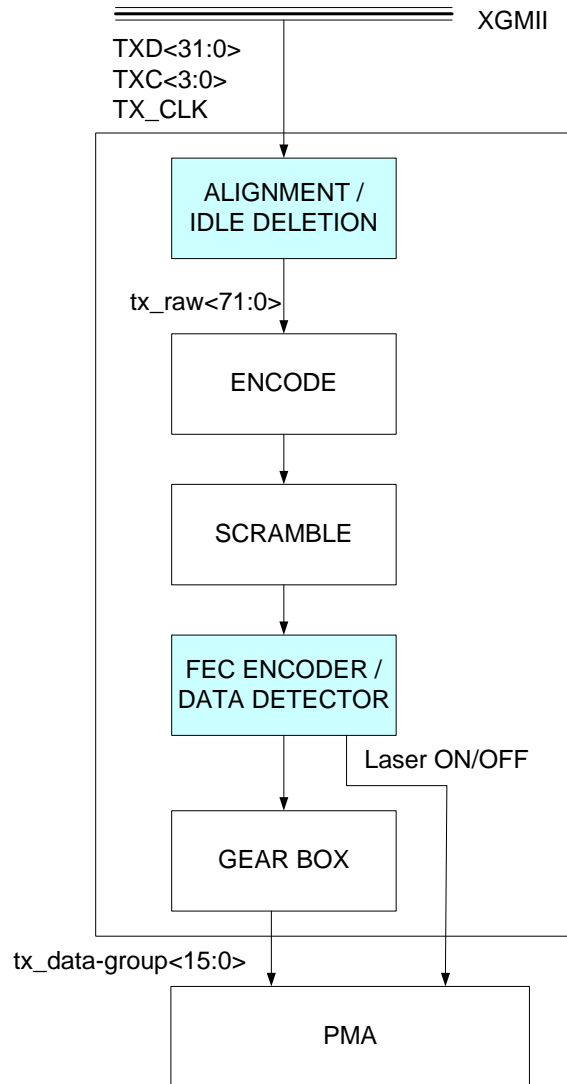
- There are several open issues
  - Idle deletion needs to be defined
  - PCS needs to be able to handle errors and reserved codes
  - Alignment of start character needs to be defined
- This presentation proposes a PCS state machine to handle all of these issues.

# Alignment / Idle Deletion

---

- Align start of burst to Lane 0
  - Allows ONU to be more efficient in filling up entire grant slot.
  - If not aligned, the extra 4 bytes may occasionally force the ONU to not transmit an extra frame at the end of the burst.
- Delete IDLE codes
  - IDLE codes need to be deleted before encoder / scrambler so that parity codes can be inserted by FEC encoder.
  - For this presentation, it is assumed that for every 28 vectors that are transmitted, 2 parity vectors need to be inserted. Other values can easily be applied.

# Functional block diagram



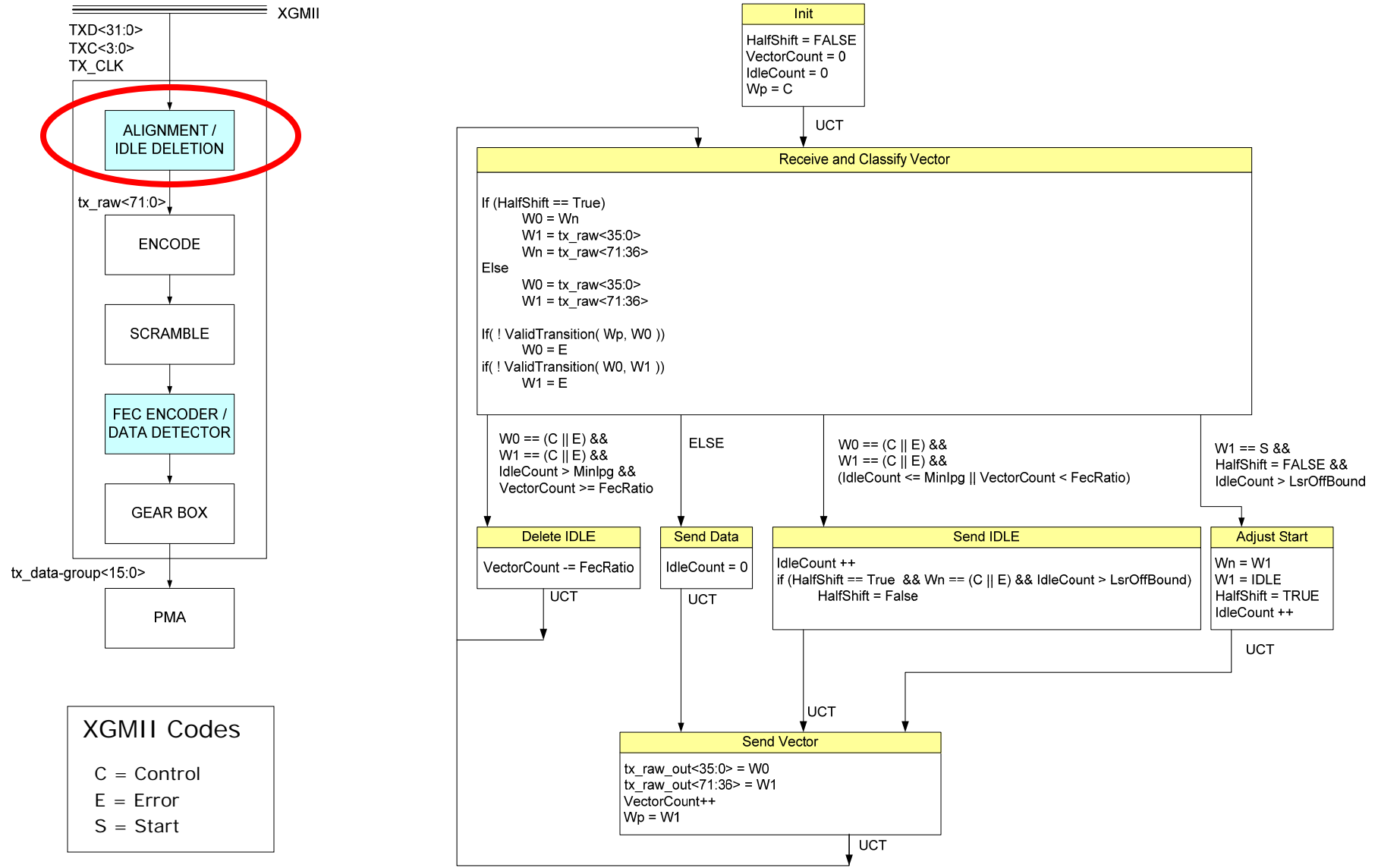
- Shaded blocks show additions to Clause 49.
- No changes to existing Clause 49 functionality.
- Need to maintain XGMII and PMA interfaces and rates.

# Moving data from RS to PCS

---

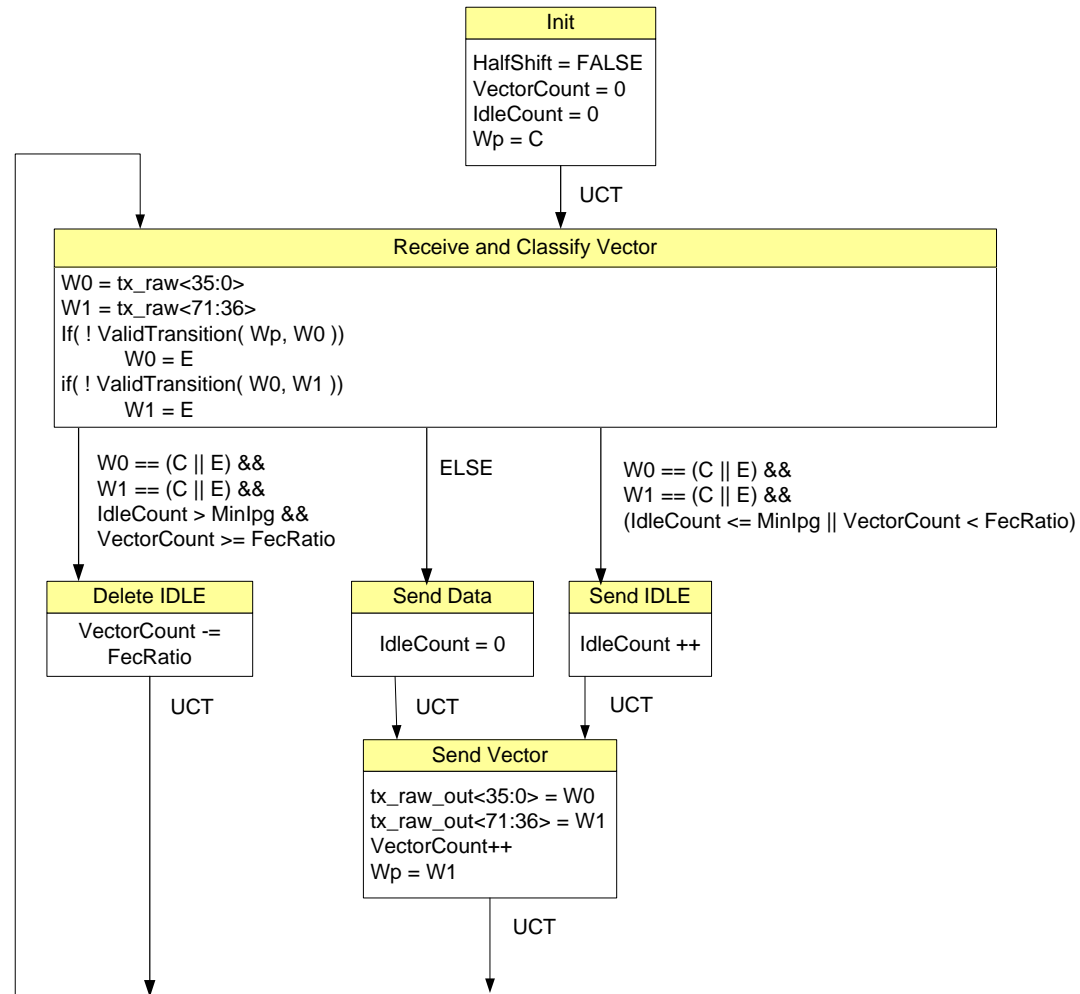
- RS transfers blocks across the XGMII
  - Each transfer contains 32 data bits and 4 control bits
  - Each transfer happens at a rate of 312.5 MHz
- PCS takes action after every second XGMII transfer
  - Two consecutive XGMII transfers provide eight characters that are encoded into one 66-bit transmission block. See Figure 49-5.
  - `tx_raw<71:0>` variable, used in state diagrams, is a “vector containing two successive XGMII transfers.” All transmit operations rely on `tx_raw` variable.
- Alignment / IDLE deletion block
  - Takes in `tx_raw<71:0>`
  - Modify it if necessary (delete IDLE or align START)
  - Output a new `tx_raw<71:0>`.

# State Diagram with Alignment



# State Diagram without Alignment

- If no start alignment is needed, the state machine is reduced.



# Description of variables

---

- **Halfshift**
  - Boolean, True if data is currently shifted by one word. False if data is not currently shifted.
- **VectorCount**
  - Counter, counts number of transmitted 72-bit vectors.
- **IdleCount**
  - Counter, counts number of 72-bit vectors containing idle or other control vectors.
- **Wp**
  - Variable, holds the previously transmitted XGMII word.
- **Wn**
  - Variable, holds the next XGMII word to transmit (what is leftover from most recent tx\_raw).
- **W0, W1**
  - Variables, hold a single XGMII transfer that is written into tx\_raw before being passed to encoder.
- **MinIpg**
  - Constant, number of 72-bit vectors consisting of IDLE that constitute minimum IPG. Value = 2.
- **FecRatio**
  - Constant, number of 72-bit vectors to transmit before deleting idle. Value = 14.
- **LsrOffBound**
  - Constant, number of 72-bit vectors needed in FIFO before laser is turned off. Value = TBD.
- **S, C, E**
  - XGMII T\_WORD\_TYPE /S/, /C/, /E/



# Description of states (1)

---

- Init
  - Initialize variables, and come to this state after reset
- Receive and Classify Vector
  - Take in two consecutive XGMII transfers
  - Assign one  $tx\_Raw<71:0>$  to two intermediate variables:
    - If data is not currently shifted
      - $W0 = tx\_raw<35:0>$ ,
      - $W1 = tx\_raw<71:36>$
    - If data is currently shifted
      - $W0 = Wn$ ,
      - $W1 = tx\_raw<35:0>$ ,
      - $Wn = tx\_raw<71:36>$
  - Check that the transition between XGMII codes is valid
    - If valid transition, do nothing
    - If invalid transition, replace with error codes

## Description of states (2)

---

- **Receive and Classify Vector (continued)**
  - There are 4 possible classifications and results:
    - **W0 and W1 are IDLE and should be deleted.**
      - Delete IDLE
    - **W0 and W1 are IDLE and should not be deleted.**
      - Transmit W0 and W1 as they are
      - If data is shifted and burst is complete, remove the shift.
    - **W0 is IDLE, W1 is START and the data is not shifted.**
      - Shift data by inserting IDLE
      - $W0 \leftarrow W1$ ,  $W1 \leftarrow \text{IDLE}$ ,  $Wn \leftarrow W1$
      - Transmit W0 and W1
    - **Everything else**
      - Transmit W0 and W1 as they are

## Description of states (3)

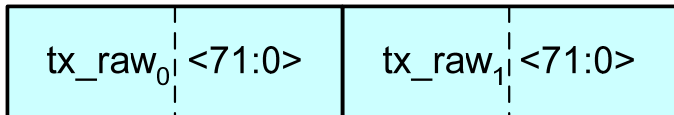
---

- Delete Idle
  - Decrement VectorCount by FecRatio
  - Send nothing, and go get new tx\_raw vector from RS
- Send Idle
  - Increment IdleCount
  - Send Idle without modification
- Send Data
  - Set IdleCount to 0
  - Send Data without modification
- Send Vector
  - Increment VectorCount
  - Transfer complete tx\_raw<71:0> to PCS encoder
  - Go get new tx\_raw vector from RS

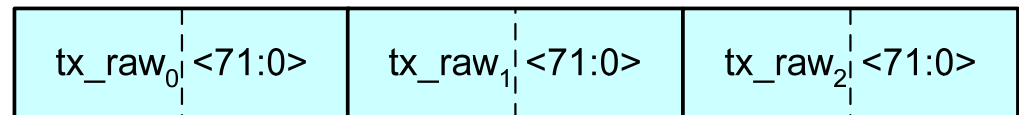
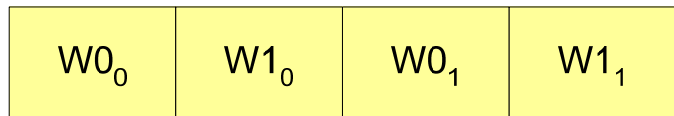
## Description of states (4)

---

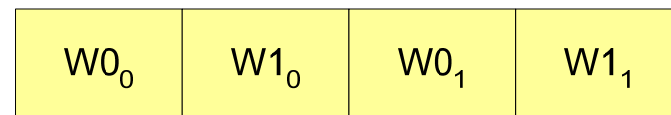
- Adjust Start
  - Shift data by inserting IDLE
  - Set HalfShift to TRUE



Data when HalfShift = FALSE



Data when HalfShift = TRUE



# Recap of when to modify XGMI I data

---

- Align START if
  - Start of burst, and START is in lane 4 of `tx_raw_in<71:0>`.
  - For each burst, the entire burst will either be shifted or not shifted. Shift will be removed at end of burst.
- Delete IDLE if
  - Min IPG has been transmitted after frame, and if we have had 14 `tx_raw<71:0>` transfers without deleting IDLE.
  - We need to delete 2 IDLE vectors for every 28 vectors transmitted.

# Conclusion

---

- A state machine has been presented that:
  - Deletes Idle to allow for FEC parity insertion
  - Performs error checking by looking at transition between consecutive XGMII words
  - Aligns start of burst to beginning of vector