



# **Idle Insertion simulation results and state diagram improvements**

---

**Eric Lynskey**

# Update of action item

---

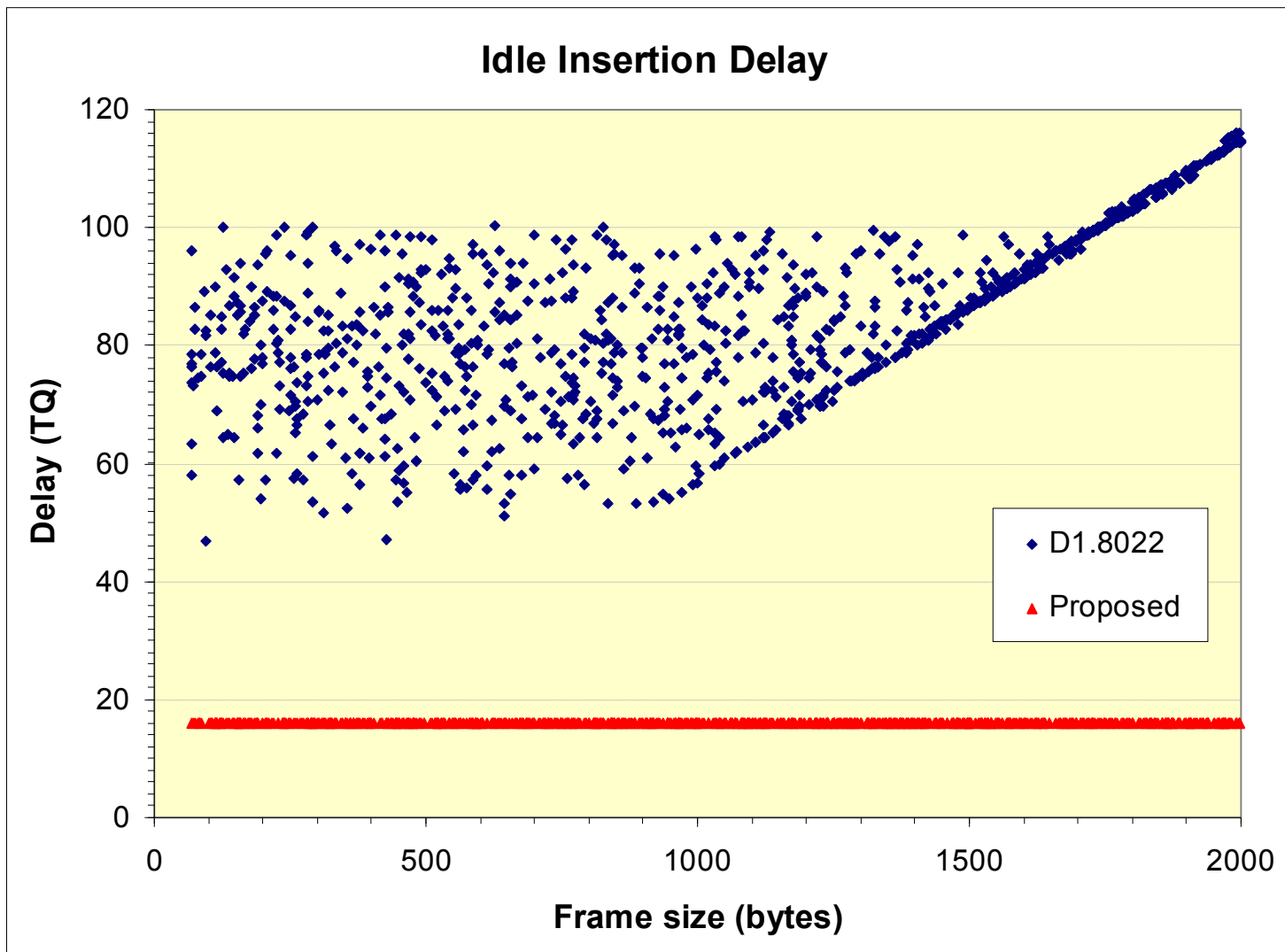
- An action item was taken in the May meeting to model the state diagrams of Clause 92.
  - Prove that state diagrams are functional
  - Prove that state diagrams are technically complete
  - Prove that state diagrams minimize delay variation
- This presentation shows the initial results of that effort as it relates to idle insertion.
- Simulation results are presented and recommendations are made to modify idle insertion state diagram.

# Problem statement

---

- The existing Idle Insertion mechanism in D1.8022 does not maintain a constant delay.
  - Frames of different lengths experience varying delays.
  - Frames of the same length experience varying delays.
  - Maximum delay variation approaches 70 TQ.
- The delay variation through the PMA and PCS must be kept to no more than 1 TQ.
- For MPCP to function properly the delay variation must be kept below 8 TQ.
- **Idle Insertion state diagram must be updated to reduce delay variation.**

# Comparison of idle insertion methods



# Existing state diagrams in D1.8022

- Large latency variation for different frame lengths.
- Large latency variation for frames of same length.
- Two diagrams with 9 total states.
- Complexity of functions and counters maintained by different state diagrams.

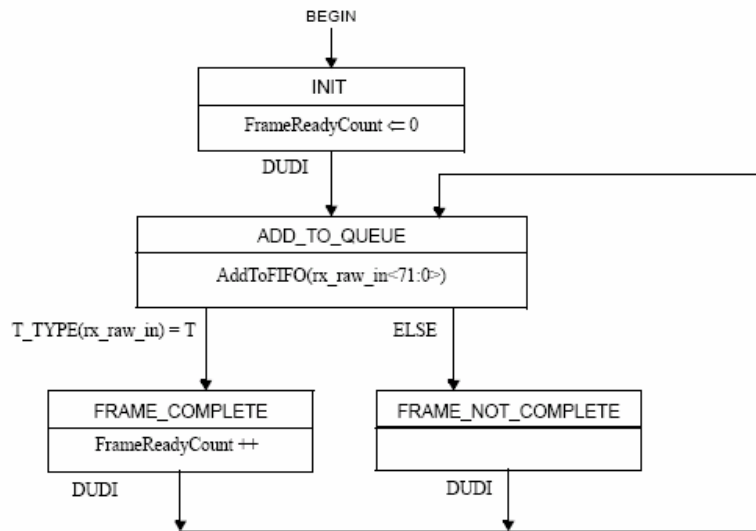


Figure 92-26—PCS Idle Insertion, input process state diagram

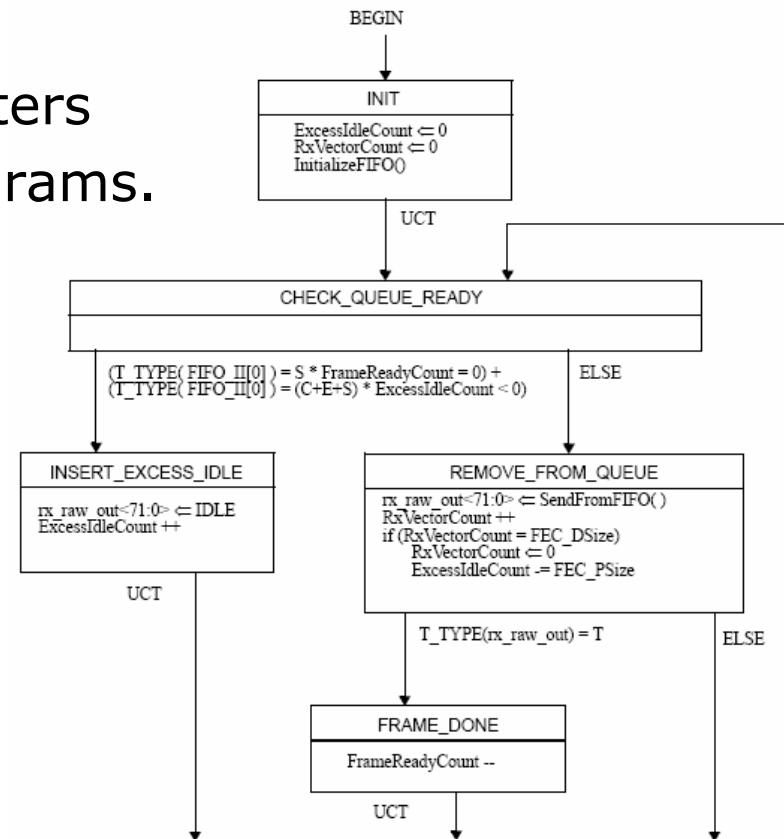


Figure 92-27—PCS Idle Insertion, output process state diagram

# Proposed new state diagram for D2.0

- Constant delay through FIFO for all lengths.
- One diagram with 5 states.
- Reduced number of functions and counters.

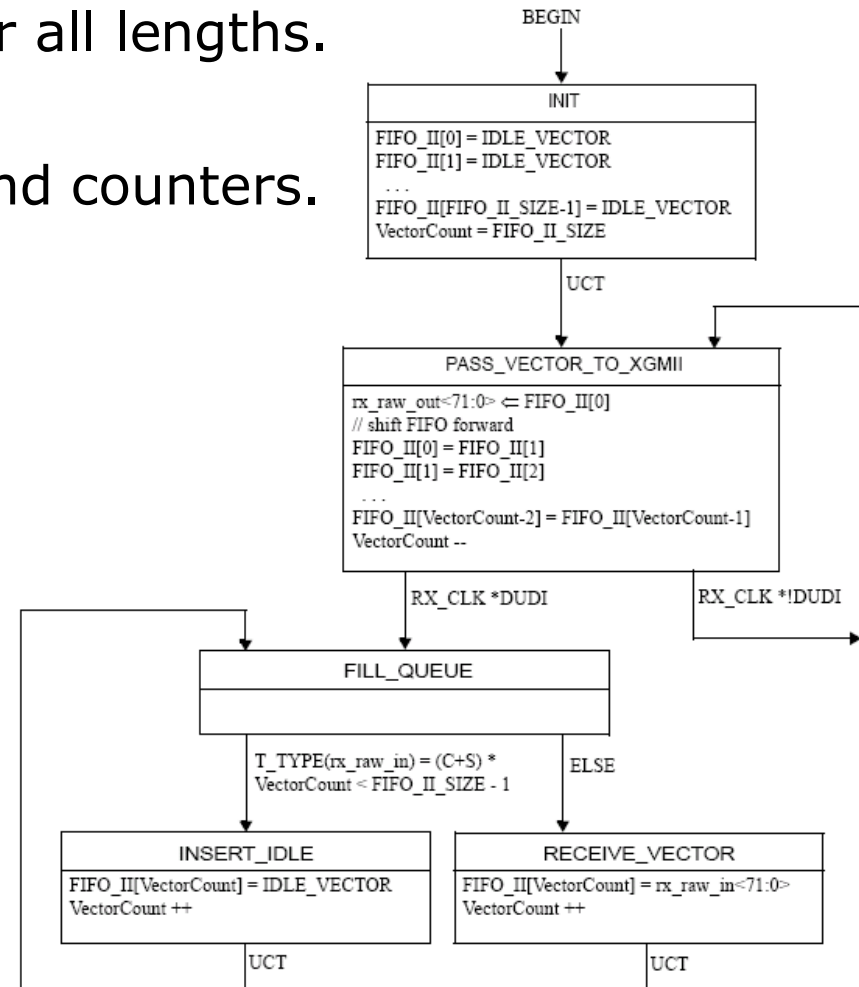


Figure 1-1—PCS Idle Insertion

## Description of state diagram

---

- Data rate leaving FIFO is 31 out of 31 XGMII clocks.
- Data rate entering FIFO is 27 out of 31 XGMII clocks.
- A maximum length frame of 2000 bytes can have up to 10 sets of parity inserted. Thus, up to 40 parity vectors can be removed by FEC decoder from one frame. FIFO needs to pre-buffer enough vectors (>40) in order to prevent run.
- General idea: When /S/ block arrives to the FIFO\_II, make sure it sees 40 vectors buffered ahead of it.
- How to do it: At the end of every frame, refill FIFO with IDLE to guarantee that the FIFO has 40 vectors in it.
- No matter what the length of the frame, it will receive a delay of 40 vectors. This corresponds to 16 TQ.

# Description of new states

---

- **INIT state**
  - Initialize FIFO to all IDLE
- **PASS\_VECTOR\_TO\_XGMII**
  - Send head of FIFO to XGMII at constant rate
  - Advance FIFO
- **FILL\_QUEUE**
  - Determine whether to add data or fill with IDLE
- **INSERT\_IDLE**
  - Fill the rest of the FIFO with IDLE. State diagram will loop through FILL\_QUEUE and INSERT\_IDLE states until FIFO is full and before next DUDI.
- **RECEIVE\_VECTOR**
  - Add received data to FIFO



# Walkthrough of state diagram

---

- After reset, initialize FIFO to be filled with IDLE.
- At every XGMII clock, transfer head of FIFO to XGMII.
- Add IDLE to FIFO until Start of first frame received.
- During frame, add data to FIFO.
- After frame refill FIFO with IDLE.
- FIFO size must be large enough to handle parity inserted with maximum length frame.
- Each Start of frame is inserted at bottom of FIFO and will have constant delay based on FIFO depth.

# Simplification of draft

---

- Figure 92-26 and Figure 92-27 are merged into a single figure with fewer states.
- The following functions are removed:
  - AddToFIFO();
  - InitializeFIFO();
  - SendFromFIFO();
- The following counters are removed:
  - FrameReadyCount
  - ExcessIdleCount

# Conclusion

---

- Simulations show that existing Idle Insertion state diagram can introduce an unacceptable amount of latency variation.
- A new state diagram has been presented.
- Simulations show that proposed Idle Insertion state diagram introduces a fixed amount of latency variation of 16 TQ.
- The Task Force should consider this state diagram as a replacement before heading into Working Group ballot.