# Reduced MII

- Low pin count MII replacement for Ethernet switches
  - Pin count down from 16 to 7 per port
  - Cost reduction in high density switches
  - Single synchronous clock
  - Easy change from existing MII silicon

# Features of RMII

- Single 50MHz ref clock for Tx, Rx and control
- Elastic buffer
- 2-bit wide transmit and receive paths
- Carrier_sense & Receive_data_valid combined into CRS_DV
- No TX_ER to force violations on wire
- No Collision signal
- Data replacement on receive error

# RMII/HSTR - issues so far

- Issues specific to RMII
  - EB size for 18K frames
  - Inability to transmit abort sequence
  - Detecting start of frame without an SFD
  - False carrier event
- Issues common to MII, including:
  - Auto-negotiation compatibility
  - Bring link down with “hard reset”.

# Elastic Buffer size

- Ethernet EB too small for 18K TR frame
- Possible solutions
  - Persuade PHY vendors to increase EB size
  - Specify tighter tolerance clocks
    - Cost implications for adapters, less so for switches
    - Optionally adapters could use loop timing - currently not supported by all MII PHYs
  - Specify adapters to always use loop timing

# TX_AB without TX_ER

- Unable to generate code violations without TX_ER signal
- Can use “invalid FCS” & ‘E’ bit as per existing state 4210
- Would need to define E bit(s) in ET field

# Detecting Start of Frame Data

- RMII converts JK SSD to 4 ‘00’ di-bits
  - May be 0 or more ‘00’ di-bits before this
  - Hard to find AC of frame, as PPP undefined
  - Options to easily detect frame start:
    - Fix PPP bits in AC (‘11x’ good, ‘000’ bad as emulates false carrier)
    - Replace AC entirely with an Ethernet SFD (MAC will need to patch it back)
    - Insert Ethernet SFD before AC (safe option)

# False carrier

- A false carrier event is a badly formed SSD (JK pair) - converted to ‘00’ on reception
- Rest of “frame” filled with “10” di-bits until carrier event finishes (idles received)
- Must be distinguishable from start of frame event (any of previous options OK)
- Note RX_ER still operates per 802.3u MII

# RMII silicon

- AMD, Broadcom, NatSemi, TI all signed up
- Very preliminary info available now
- Derivatives of existing MII devices
- Quad devices or bigger (up to octal?)
- Larger EB possible, but impacts layout
- Window for including HSTR-specific changes is closing

# Suggested solutions

- Elastic Buffer:
  - Get friendly with the PHY vendors!
- Transmit abort with no TX_ER signal
  - Use invalid FCS and define E bit in ET
- Detecting frame data and False Carrier
  - Specify SFD in frame as per Ethernet