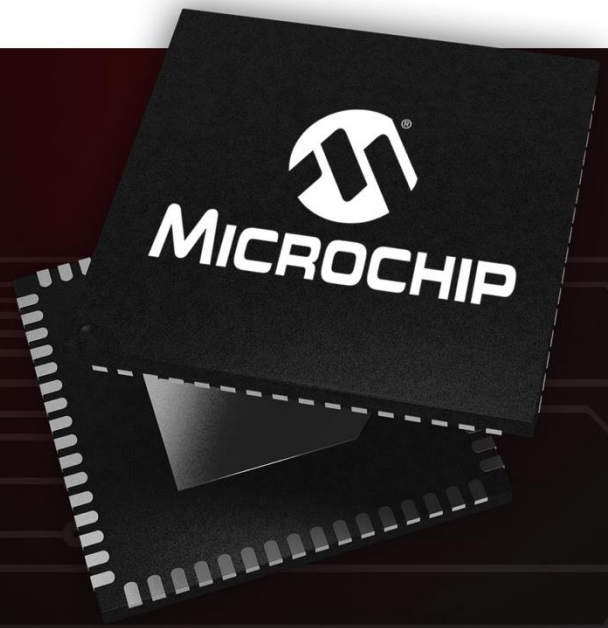
MICROCHIP



A Leading Provider of Microcontroller, Mixed-Signal, Analog & Flash-IP Solutions




IEEE 802.3cg
Priority-Based PLCA
October 2018

Future-proofing PLCA for priority-based transmit ordering

Contributors
Michael Rentschler, Venkat Iyer

Introduction

- **Communicating priority from MAC client to RS is being worked via NEA**
- **Following the suggestion from last ad-hoc meeting to future-proof PLCA for priority**
- **Goals in developing proposal**
  - Minimize wait time for priority packet transmit
    - Wait till end of packet, not end of bus cycle
  - No adverse EMC impact
  - Robust in face of bit errors
  - Focus on 2 levels but extensible to more levels
  - Interoperate with simple nodes which don’t need priority
  - Minimize changes to PLCA (state machines)

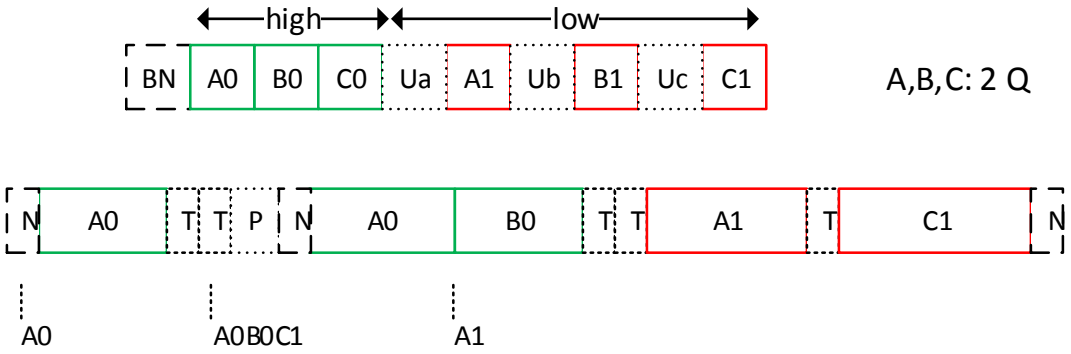
Proposal

- **Fixed priority assignment by node ID**
  - Starting with highest priority at ID=0, descending order
  - Multiple IDs can be assigned per node (one per priority queue)

- **Introduce Priority Request (PRQ) signaling**
  - Each node must be able to signal PRQ if curID > levelID
  - PRQ can occur in front of a TO or alternately at the end of a frame
  - Nodes can prevent use of lower priority TO by issuing PRQ
    - Receiving PRQ will cause the bus master to preempt the running bus cycle by issuing a new Beacon

PLCA Control SM

- **How can this be integrated into current PLCA state machine?**
  - Reserved/Unassigned IDs (Ux) will be inserted for PRQ signaling
  - If PRQs collide
    - PRQ signal can be the scrambled ID, to mitigate risk of phase cancellation
    - Collision detect ‘not’ needed; only required to sense carrier to detect PRQ
  - Very little change to PLCA control state machine
    - Master: WAIT_TO->EARLY_RECEIVE ->RECOVER, sends new beacon
    - Slave: WAIT_TO->EARLY_RECEIVE ->RESYNC, resyncs on new beacon

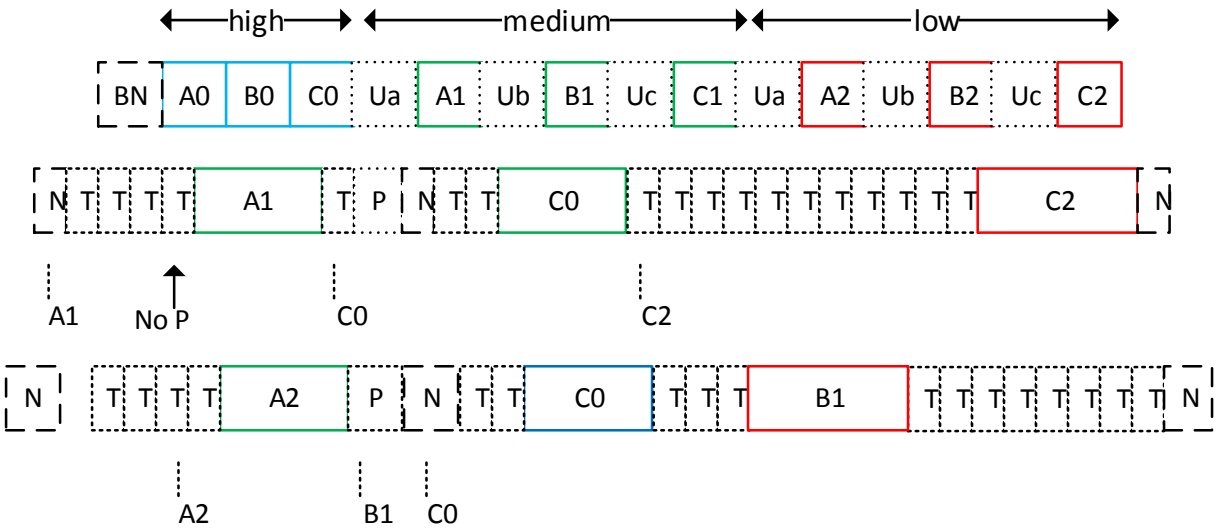


PLCA Data SM

- **If there is no PRQ, PLCA data state machine is traversed as before**
- **During PRQ**
  - receiving=FALSE → PRQ doesn't go to MII RX
  - Transmitters delay line(s) will continue to 'HOLD' TX symbols till TRANSMIT or COLLIDE
- **No changes to PLCA Data state machine necessary**

Extending to more levels

- **Example with more priority levels**
  - 3 nodes (A, B, C)
  - 3 priority levels per node (0, 1, 2), one segment for each
  - All, but highest segment, includes unassigned IDs (Ux)



- **Optional: reduce number of IDs by eliminating unassigned IDs**
  - Increase TO, split into two windows and use earlier window for PRQ
  - In WAIT_TO, CRS=FALSE actions delayed till latter half of TO
  - Alternately, nodes observe a PRQ window at the end of a frame

Prevent unfairness

- **Fixed priority can cause unfairness, which may be undesired**
- **Possible solutions**
  - Grouping priorities together into evenly balanced segments
    - An MLQ scheduler will allow fairness per level
    - Each MLQ level will operate it’s own Round-Robin schedule
    - After a TO is used, that TO is yielded till curID reaches maxID or the ID of the next level
  - Prevent starvation of lower priority queues
    - Master monitors the time since the last full bus cycle
    - If extensive PRQ requests cause time to exceed a given limit, the master is able to ignore further PRQs

Conclusion

- **Future proofing PLCA for priority based transmit ordering is possible by pre-empting bus cycle with new beacon**
- **Proposal meets goals identified earlier**
  - Minimizes wait time for priority packet transmit
    - Limits wait to end of packet, not end of bus cycle
  - No adverse EMC impact
  - Robust in face of bit errors
  - Extensible to 2 or more levels of priority
  - Interoperates with simple nodes which don’t need priority
  - Minimizes changes to PLCA (state machines)

Thank You!