High Frequency Response Tracking System micro-architecture

Gopalakrishnan Sridhar, Senior Logic Design Engineer/Micro-architect, Intel Technology India Pvt Ltd, Bengaluru, India (*gopalakrishnan.sridhar@intel.com*)

Vadlamuri Venkata Sateesh, Logic Design Engineer, Intel Technology India Pvt Ltd, Bengaluru, India (*vadlamuri.v.sateesh@intel.com*)

accellera SYSTEMS INITIATIVE™

2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

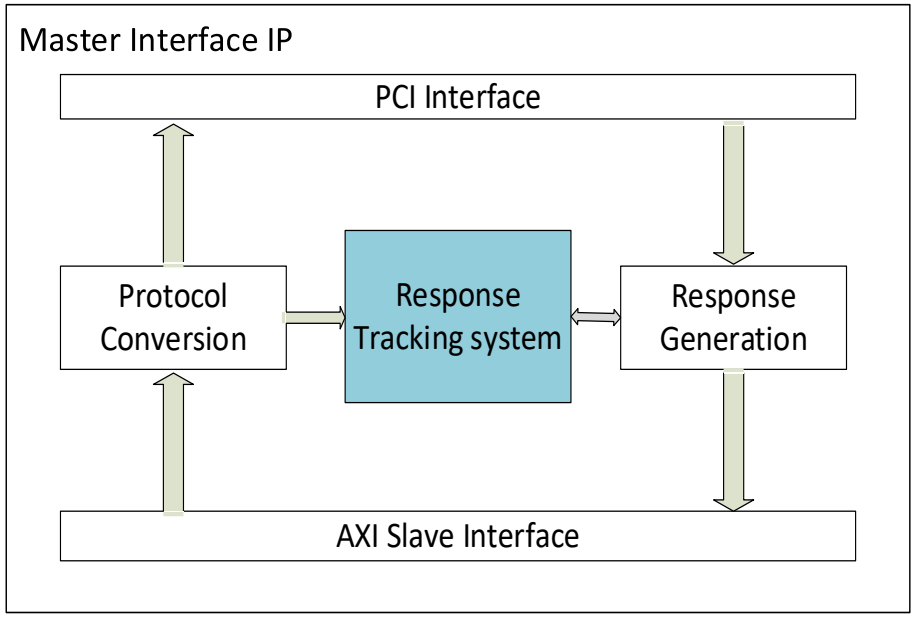
Agenda

1. Introduction
2. Response Tracking System
3. Complexities in meeting Higher Frequencies
   A. push a request entry into the RTS
   B. pop from the RTS during response
4. Summary
5. References

Introduction

Why Response Tracking System is Required ?

- To track the completions of the Requests initiated by the Bus master
- To frame the proper completion format.
- To check for the unexpected completions
- To track the remaining byte count.



Introduction(contd..)

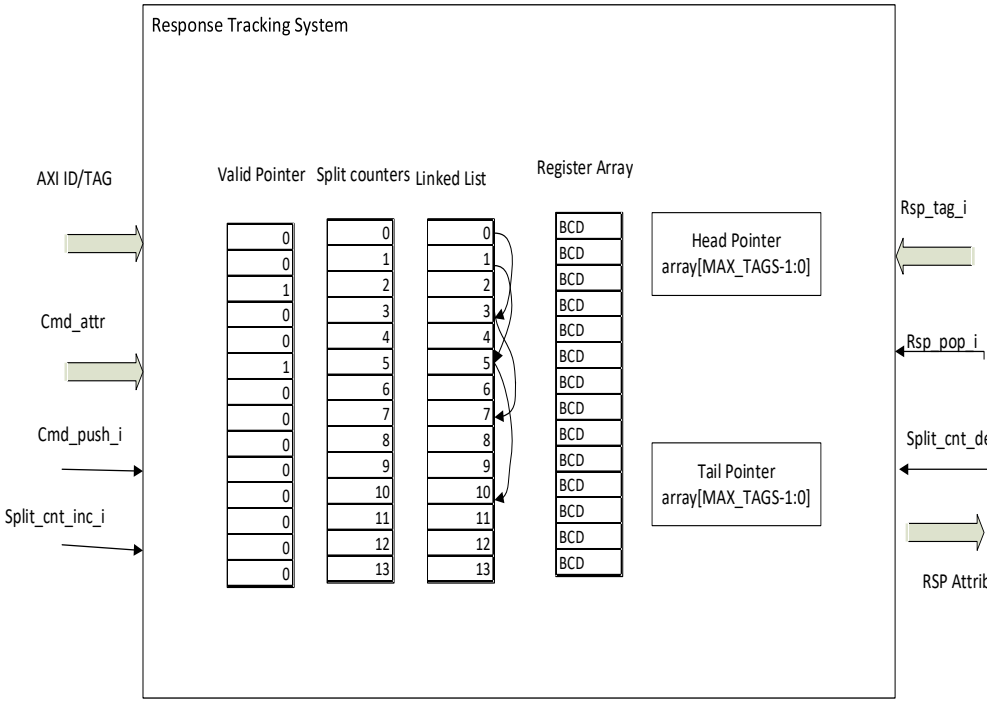
What are the complexities with Response Tracking System ?

- Dynamic buffer management is required, as the static queue allocation for each tag is area prone.
- Response of different tags can come out of order.
- For every new transaction going out of bus master, has to create a link with the existing linked list.
- As response of each tag comes in the same order, so RTS should be capable of removing that entry from the linked list.
- Has to provide the its attributes on a “tag”.

Response Tracking System

1. During Bus Master Request
   - Set the tag valid, head/tail pointer and link list
   - Store command attributes

2. During Response to the bus master
   - Provide the transaction attributes for the given tag
   - Remove the transaction form the cpl tracker on rsp_pop assertion



Response Tracking System Functionality

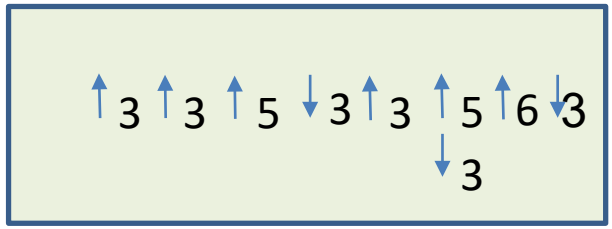
Tag	Valid
2	
3	0
4	
5	1
6	1

Head
9
7
13

Tail
9
11
13

Current ptr	Next ptr
2	3
3	9
7	11
9	
11	

Free ptr
13



Complexities for meeting Higher Frequencies in Response Tracking System

During a entry push into the RTS:

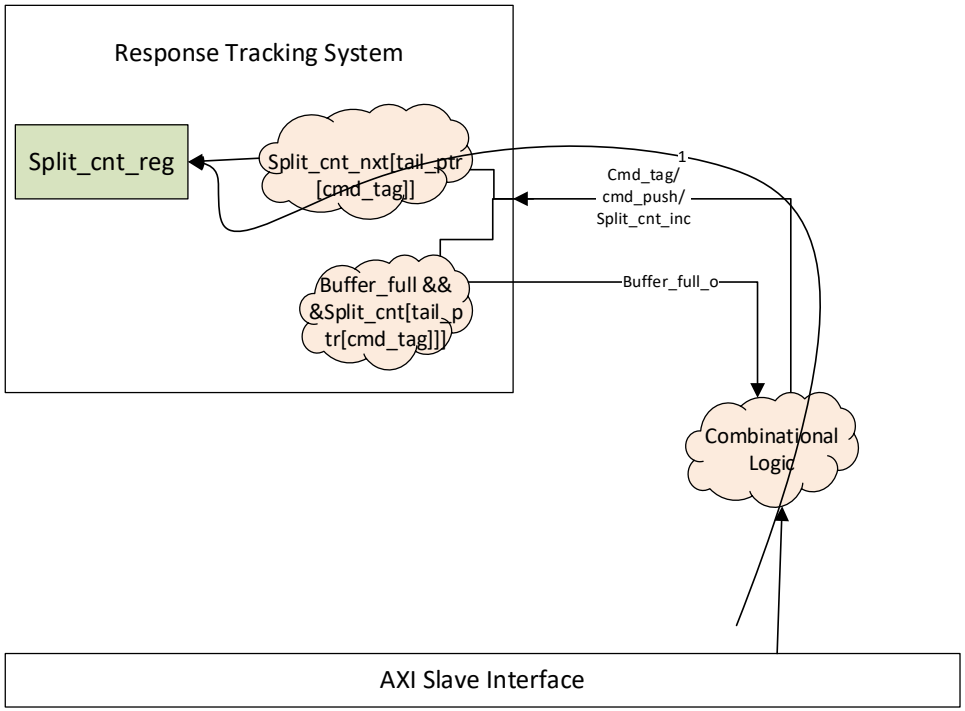
- RTS has to check whether its buffers are full or not and then update tail pointer and split counter.

On arrival of response, evicting an entry from RTS:

- RTS has to find the head of the linked list of a given tag
- And then find the split count and attributes of the head pointer.
- Based on split count and attributes decide to remove the entry or not.

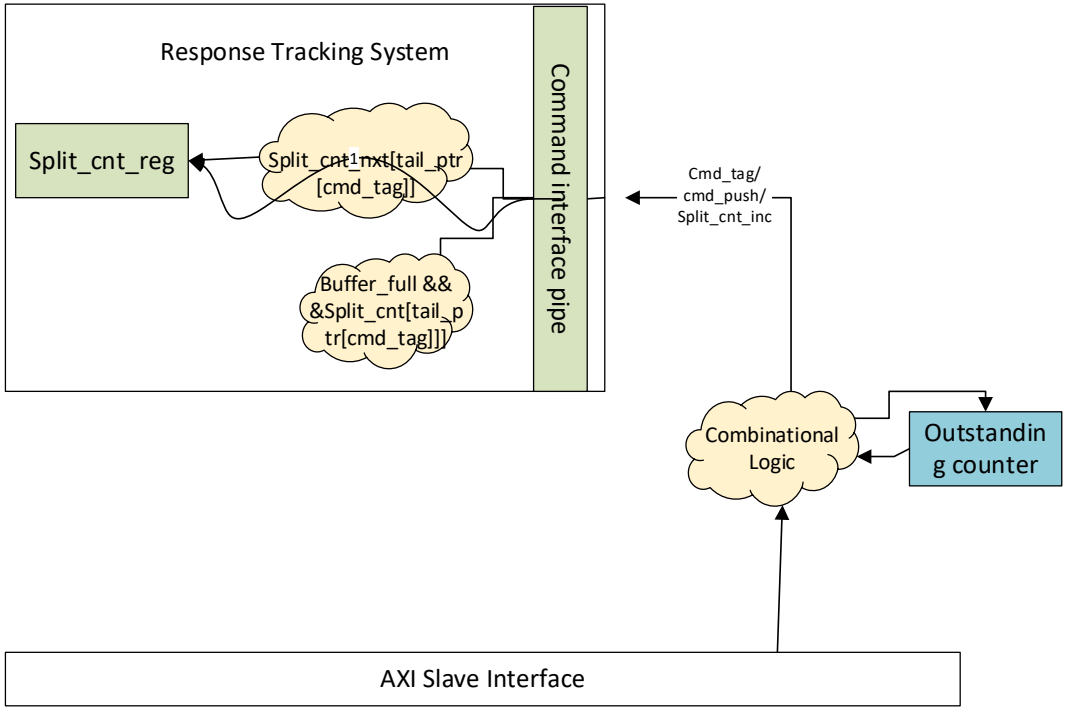
Entry push into RTS

- Pushing the entry after buffer full indication check from RTS leads to huge combinational path
- Pipelining command push will lead to improper functionalities



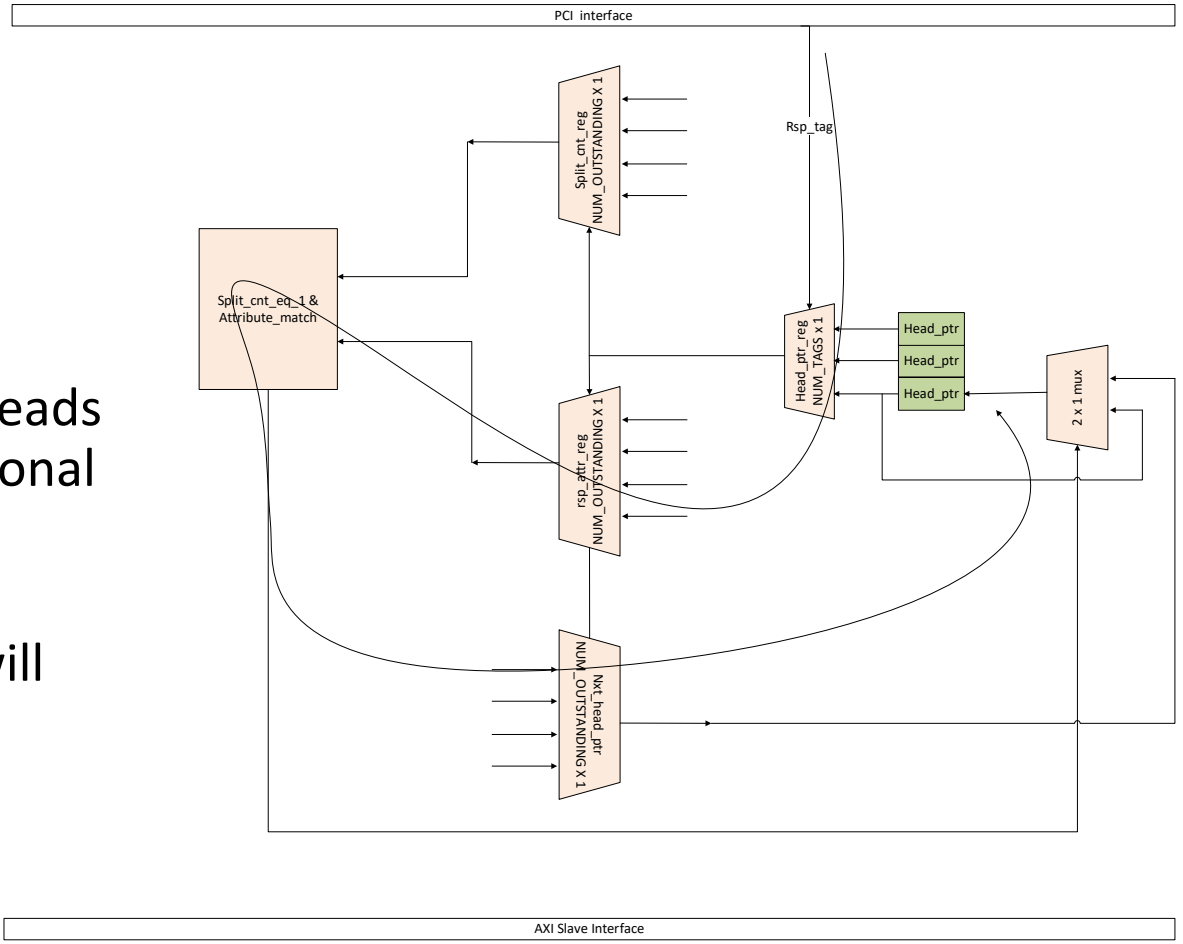
Micro-arch solution For Entry push

- Instead of using buffer full information from the RTS, check with one outstanding transaction counter, before pushing the transaction
- It will not create functional implications, though command input interface is pipelined.



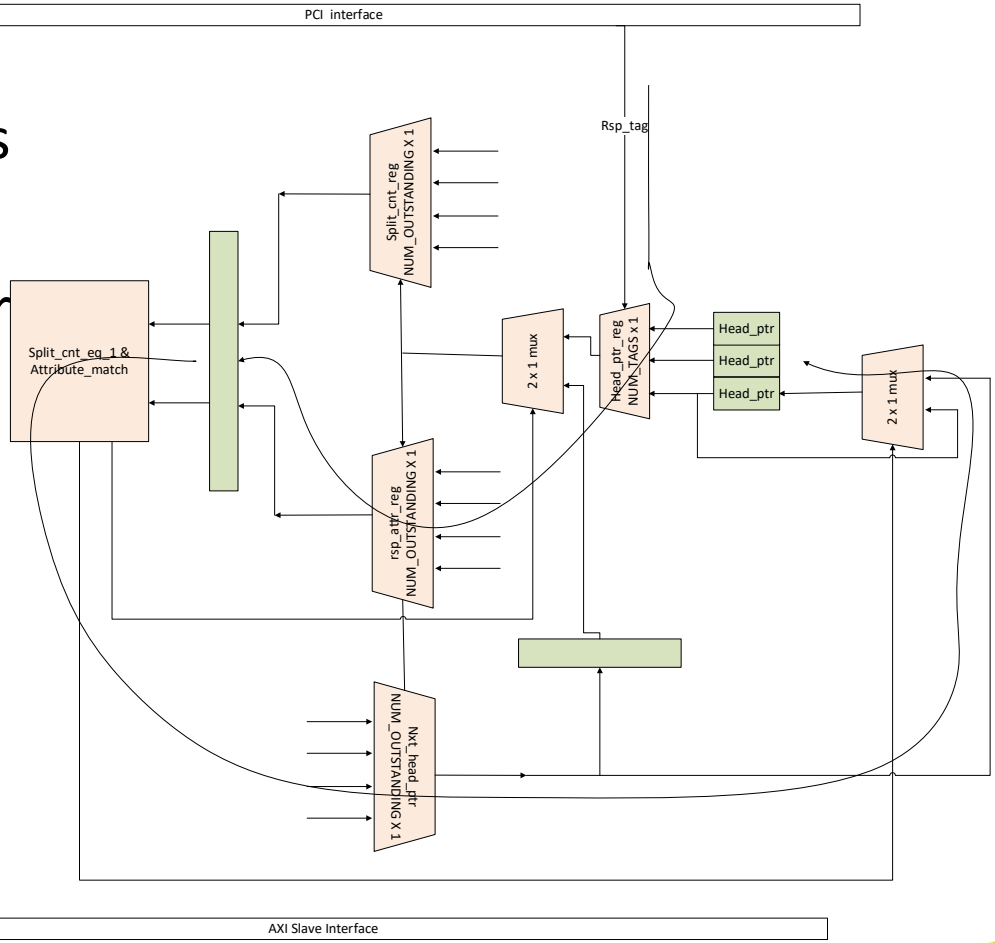
Complexity during Response pop from RTS

- Popping out entry from the RTS after checking the split count and other attributes checks leads to huge combinational path.
- Pipelining on the rsp_pop directly will lead to improper functionalities.



Micro-arch solution for Response pop from RTS

- Pipe line the all attributes separately.
- It has to pop the entry for the previous response, and to get the attributes for the current response of the same tag.



Summary

Response Tracking system with the micro-architectural solutions provided in this paper can be deployed in High Frequency SoC/NoC Designs

References

- [1] W. J. Dally and B. Towles, “Bufferd flow control,” in Principles and Practices of Interconnection Networks. San Francisco, CA, USA: Morgan Kaufmann, 2003.
- [2] H. Zhang, K. Wang, Y. Dai, and L. Liu, “A multi-VC dynamically shared buffer with prefetch for network on chip,” in Proc. IEEE 7th Int. Conf. Netw., Archit., Storage, Xiamen, China, Jun. 2012, pp. 320–327.
- [3] M. Lai, Z. Wang, L. Gao, H. Lu, and K. Dai, “A dynamically-allocated virtual channel architecture with congestion awareness for on-chip routers,” in Proc. 45th ACM/IEEE DAC, Anaheim, CA, USA, Jun. 2008, pp. 630–633

Questions ?

Thank You