An Approach to Single-Event Testing AMD-Xilinx Versal's Most Important New Architectural Feature: Network-on-Chip (NoC)

Sebastián E. García
sebastian@slabs.com.ar
slabs
+54 911 4403 1725
Coronel Suárez, BA, Argentina.

Gary M. Swift
gary.m.swift@ieee.org
SWIFT ENGINEERING & RADIATION SERVICES, LLC
+1 408 679 3785
San José, CA.

Overview

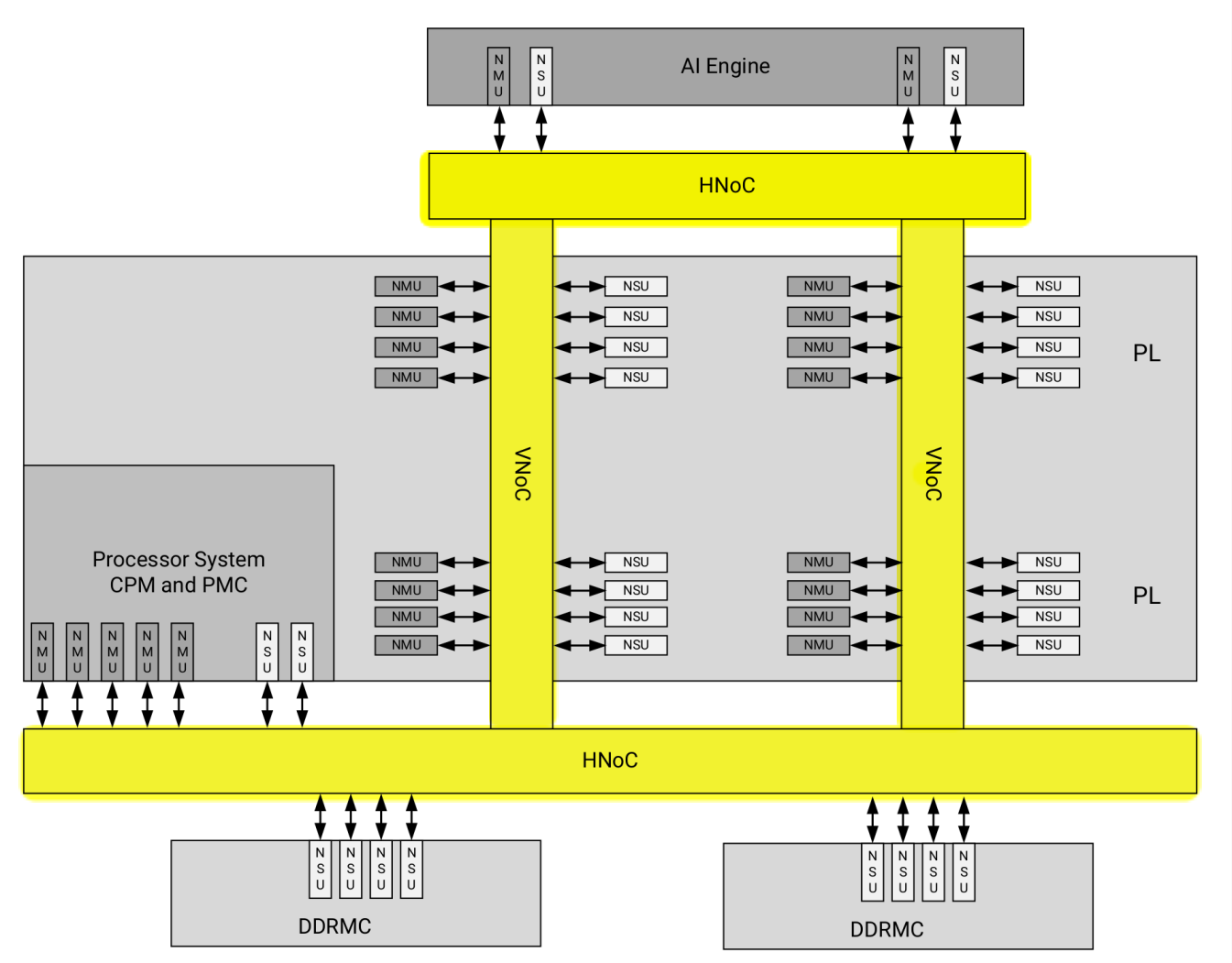
- What's a Network-on-Chip? – *NOT just marketing*.
- Why NoC's? – *Revolution, not merely evolution*.
- Versal VC1902's NoC Overview
  - *Three blocks: two AXI4 endpoints (NMU, NSU) & packet switch (NPS)*.
- How to test NoC's? – *Best and ready:* XRTC Test Platform
  - *Custom DUT Board on Gen-4 apparatus*.
  - *3 stripchart logs: error counters, CRAM upsets, and all power rails*.
- How to test NoC's? – *Custom in-beam DUT design(s)*.
- Conclusion

Network-on-Chip

Concept: from wires to switched packet network.

- Replace signal wires with shared “virtual highways,”
  - Reduces area and power use,
- Communication via “info convoys,”
- Priority rules for sharing the highway,

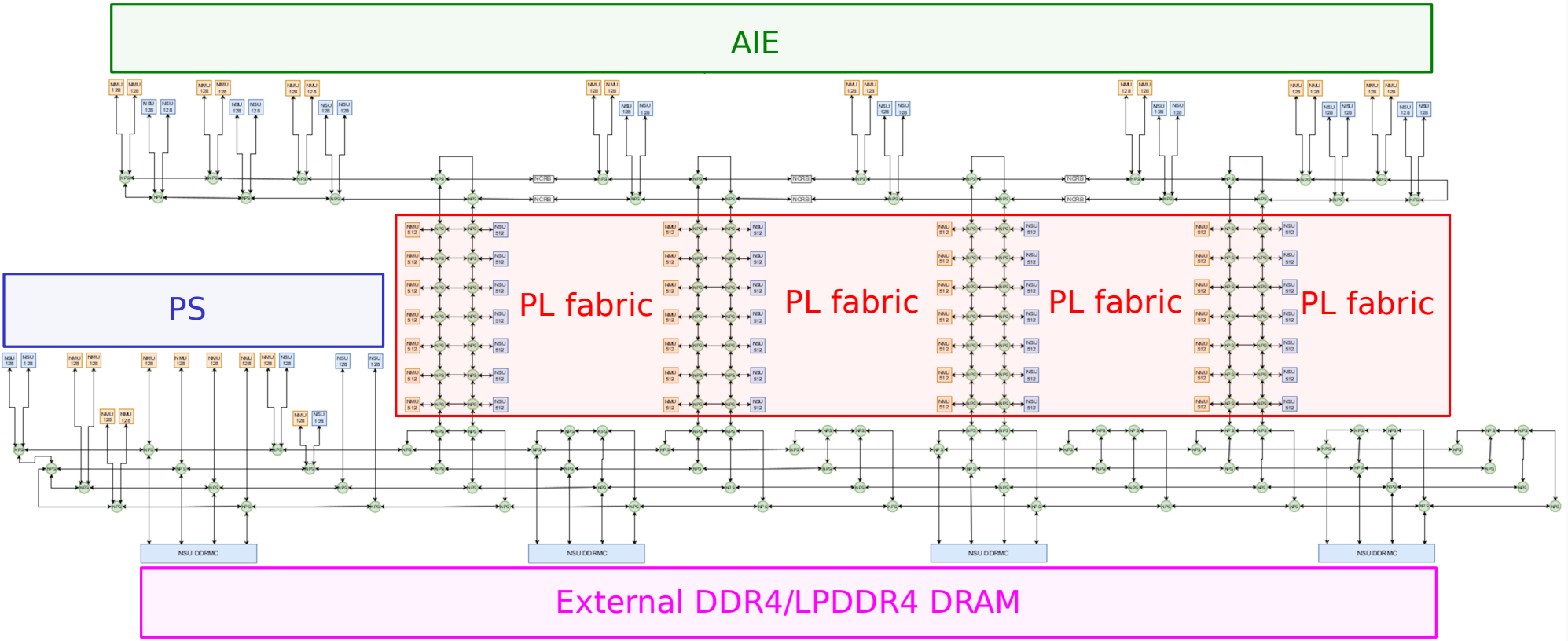
NoC Implementation in the Versal Architecture



(Figure taken from [1])

- A NoC solves the cross-chip hi-speed communication problem for a large heterogeneous SoC plus programmable fabric (ACAP).

NoC Implementation in the VC1902



(Diagram based on figure taken from [2])

NMU: NoC Master Unit (28 on PL, 10 on PS, 16 on AIEs; **54 total**);
NSU: NoC Slave Unit (28 on PL, 6 on PS, 16 on AIEs; 16 in DDRMCs; **66 total**);
NPS: NoC Packet Switch (24 on North-HNoC, 66 on South-HNoC, 56 on VNoCs; **146 total**);
NCRB: NoC Clock Re-convergent Buffer (**6 total**);
RPTR: NoC Repeater (repeater block used in devices with large distance between NPSs);
DDRMC: DDR Memory Controller (tightly integrated in the NoC);
AIE: Artificial Intelligence Engine.

Versal's NoC Characteristics and Performance

- A packet-switching network infrastructure enables efficient data movement (*vs.* wiring scaling challenges) in modern ACAP devices like Versal.
- *Ad-hoc* network topology, optimized for ACAP's components and use models. The endpoints implement AXI4 & AXI4-Stream interfaces.
- Atomic packet units are called *flits*.
  - All buffers and datapaths are flit-sized
  - 128-bit data payload (182-bit total physical width in each direction).
- The NoC is full duplex & has its own clock domain; max. data throughput is $\sim$ 128 Gb/s at 1 GHz (1 clock cycle to move between adjacent buffers).

Versal's NoC Characteristics and Performance

- *Wormhole* routing: Packets flow through the network like carriages of a train; it's possible for the header flit to arrive at the destination endpoint before the last flit has left the source endpoint.
- QoS scheme: 8 virtual channels per physical link, balancing latency and bandwidth requirements of the communication tasks.
- In Vivado, the *NoC Compiler* gathers conns. and configs. from the user's design, and outputs a piece of bitstream with conns. and parameters' values for the NoC and DDRMCs. This is the "NPI" part in the whole PDI image file.
- Data integrity: Flits are encoded with SECDED ECC, checked at endpoints. A parity bit is included in the destination ID; this is checked at switches. Errors are signaled via processor interrupts.

NoC Configuration with an Independent Mini NoC (NPI)

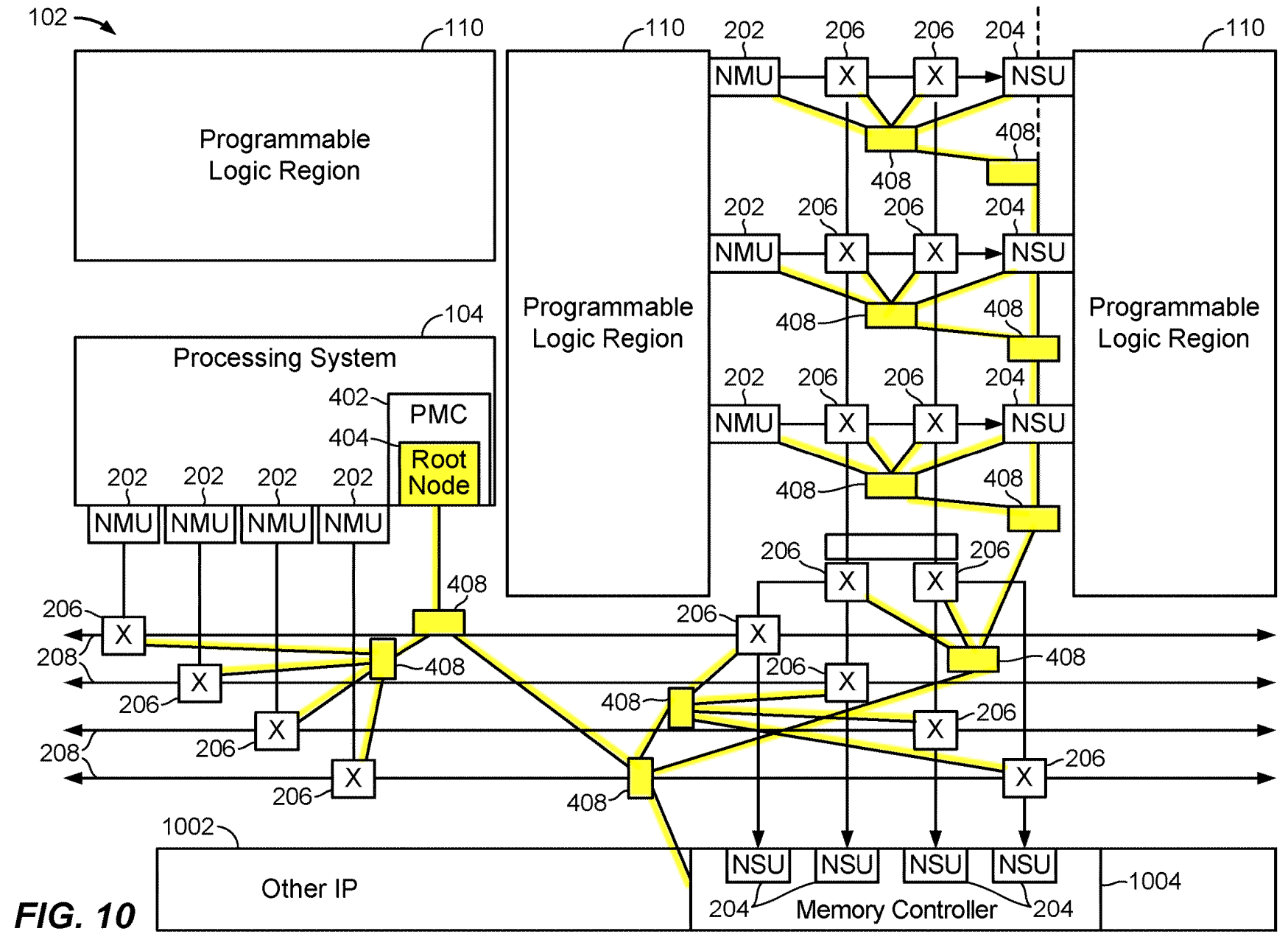


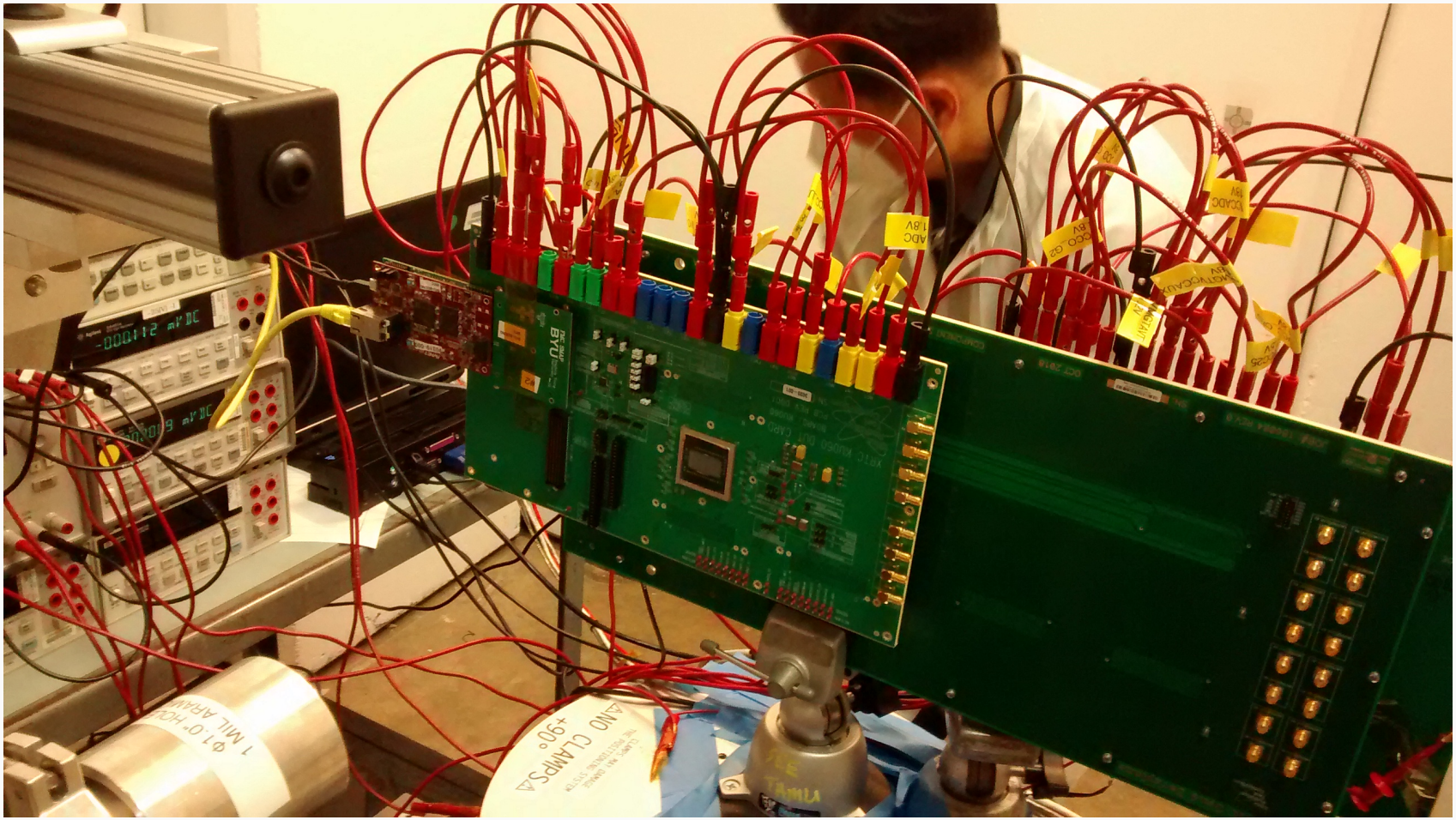
FIG. 10

(Figure taken from [3])

Estimated bit inventories

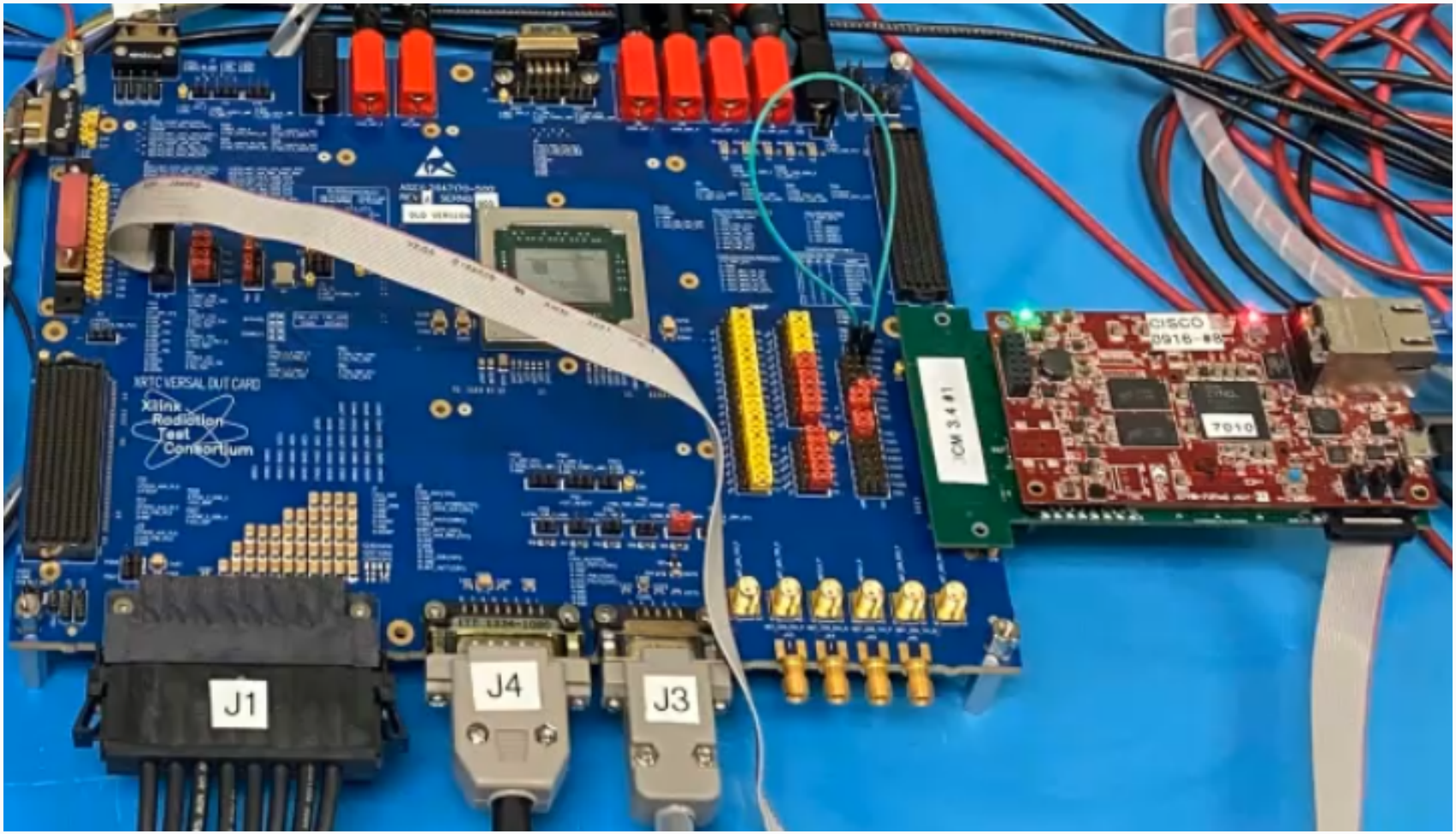
1. In *visible* NPI (NoC/DDRMCs) 32-b status/control registers: **1,261,568** bits [4].
2. In *hidden* memory cells (estimated):
   - NoC routing: NPS' (switches) routing tables configuration
     - **5,376** bits per NPS in use (max. bound: 784,896 = 146×5,376).
   - NoC buffers
     - Virtual channel buffers per NPS in HNoCs: **40,768** (= 4×7×8×182).
     - Virtual channel buffers per NPS in VNoCs: **29,120** (= 4×5×8×182).
     - Registered outputs per NPS: **728** (= 4×182).
     - Pipeline registers: **182** bits per unit in use.
   - NMU & NSU modules: TBD

Testing the NoC – XRTC's Gen-4 Test Platform



- Shown is UltraScale KU060 DUT on Gen-4 tester in beam (2021). Gen-0 tested Virtex in 1999.
- Very flexible, fine-grained visibility, high-speed test apparatus.
- Modular passive backplane based: DUT board, FuncMon (vector exerciser/logger) board on back.

XRTC's Custom VC1902 DUT Board



- Now ready for beam. Shown is Versal DUT card in benchtop check out using ConfigMon daughter module.

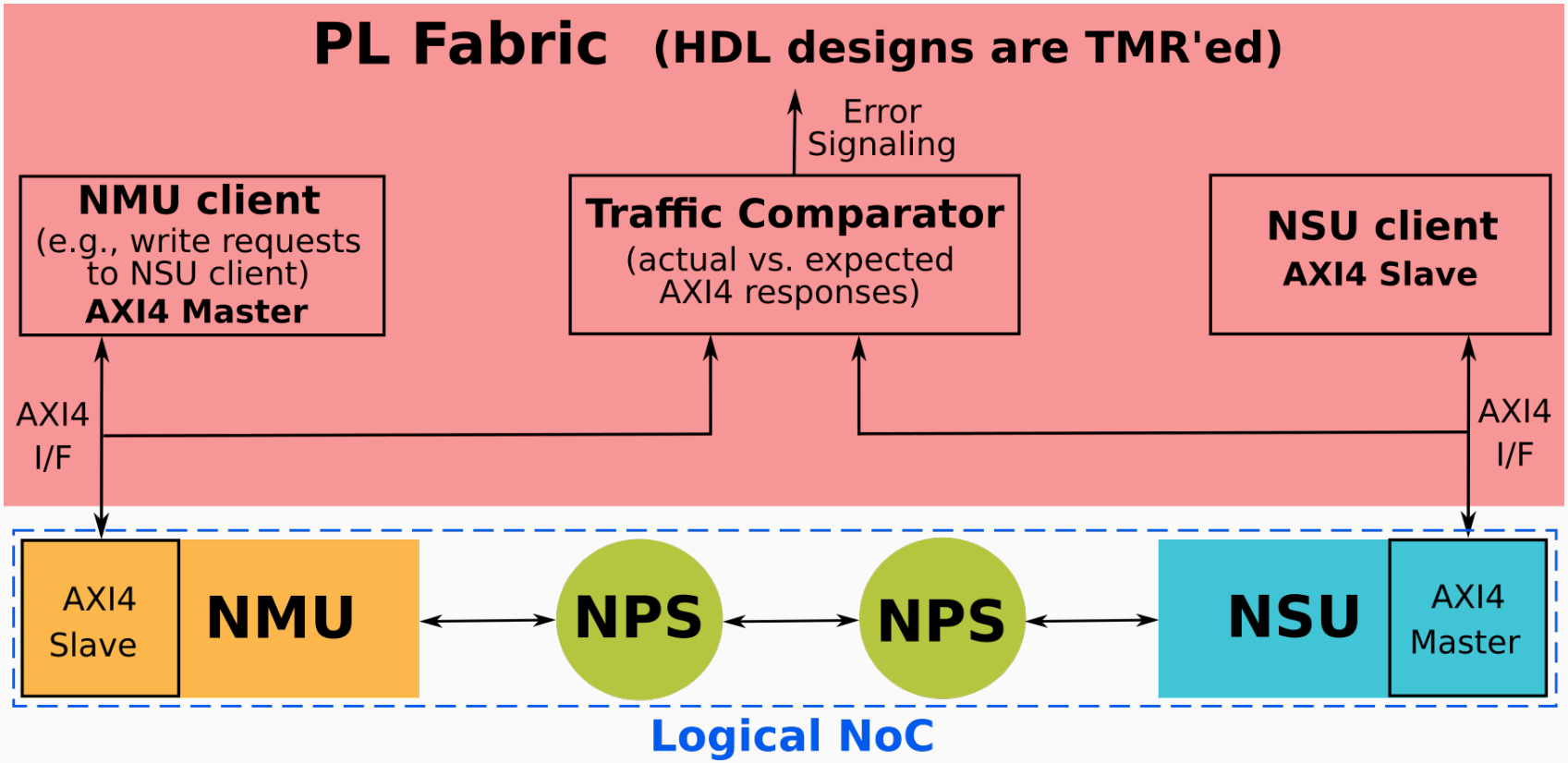
Questions related to beam response signatures

1. Number of involved NPI configuration bits? Number of critical ones?
2. How to scrub the NPI regs after errors caught by an NPI-scan? (XilSEM)
3. How to put the NoC into a “quiescent” state, in order to scrub the NPI registers’ bits? NoC_RESET bit [5]? NoC power-down?
4. Possible ways to rewrite on NPI registers? (besides a master reset of the whole device by POR_B assertion)
5. How many different ways can the beam produce a lockup, including deadlocks?

Beam Test Design Considerations

- Test goals:
  - Separate cross-sections for master and slave endpoints, and packet switches.
  - Understand the individual sources of different upset signatures.
- Test constraints:
  - High beam efficiency ⇒ Max. target area ⇒ Max. # of NoC elements.
- Test approach:
  - Start as-simple-as-possible.
    - Direct traffic (1-input port, 1-output port) through switches.
    - Shortest route: One master, 2 switches, 1 slave.
  - Capture error counts and signatures.
    - Upset buffer bits ⇒ Bad packet ⇒ PMC interrupt.
    - Latency increases.
    - Traffic lockups.
    - Others.
  - Investigate mitigation and recovery mechanisms.

Proposed in-beam test scheme



- Deterministic latency is considered in the comparison.

Conclusion

NoC complexity will require:

- Capable test apparatus.
  - XRTC's Gen-4 test infrastructure was designed to handle as much complexity as possible.
- Complex HDL test designs (even for fundamental NoC components)
- Multiple tests and multiple test trips
  - When testing goes as expected, next test will add complexity.
  - When testing doesn't go well, then troubleshoot and re-test.

Thank you!

References

[1] AMD, *Versal ACAP Programmable NoC and Integrated Memory Controller*, PG313 (v1.0), 2022.12.14.

[2] I. E. Lang, *Worst-Case Latency Analysis for the Versal Network-on-Chip*, MS Thesis, University of Waterloo, Canada, 2021.

[3] I. A. Swarbrick and D. P. Schultz, *Peripheral Interconnect for Configurable Slave Endpoint Circuits*, US Patent 10621129, 2020.04.14.

[4] AMD, *NoC and Integrated Memory Controller NPI Register Reference*, AM019 (v1.0), 2021.09.09.

[5] ——, *Versal ACAP Technical Reference Manual*, AM011 (v1.5), 2022.12.16.

[6] ——, *Versal AI Core Series Data Sheet: DC and AC Switching Characteristics*, DS957 (v1.4), 2022.05.03.

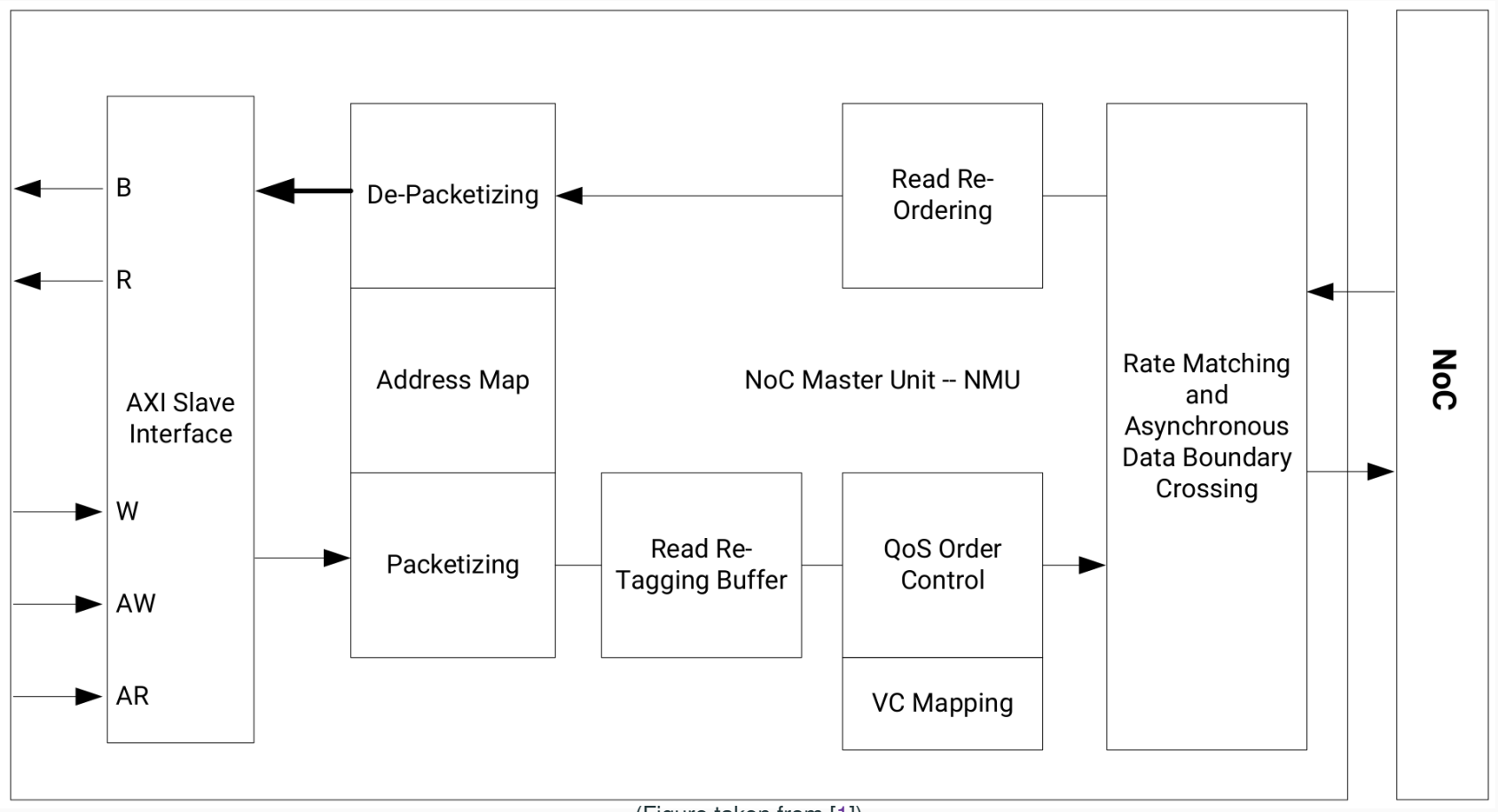
[7] I. A. Swarbrick, D. Gaitonde, S. Ahmad, B. Gaide, and Y. Arbel, *Network-on-Chip Programmable Platform in Versal ACAP Architecture*, FPGA Conference, 2019.

[8] AMD, *Xilinx Standalone Library Documentation – BSP and Libraries Document Collection*, UG643 (v2022.2), 2022.10.19.

[9] D. P. Schultz, I. A. Swarbrick, and D. Nagendra, *Circuit for and Method of Configuring and Partially Reconfiguring Function Blocks of an Integrated Circuit Device*, US Patent 10680615, 2020.06.09.

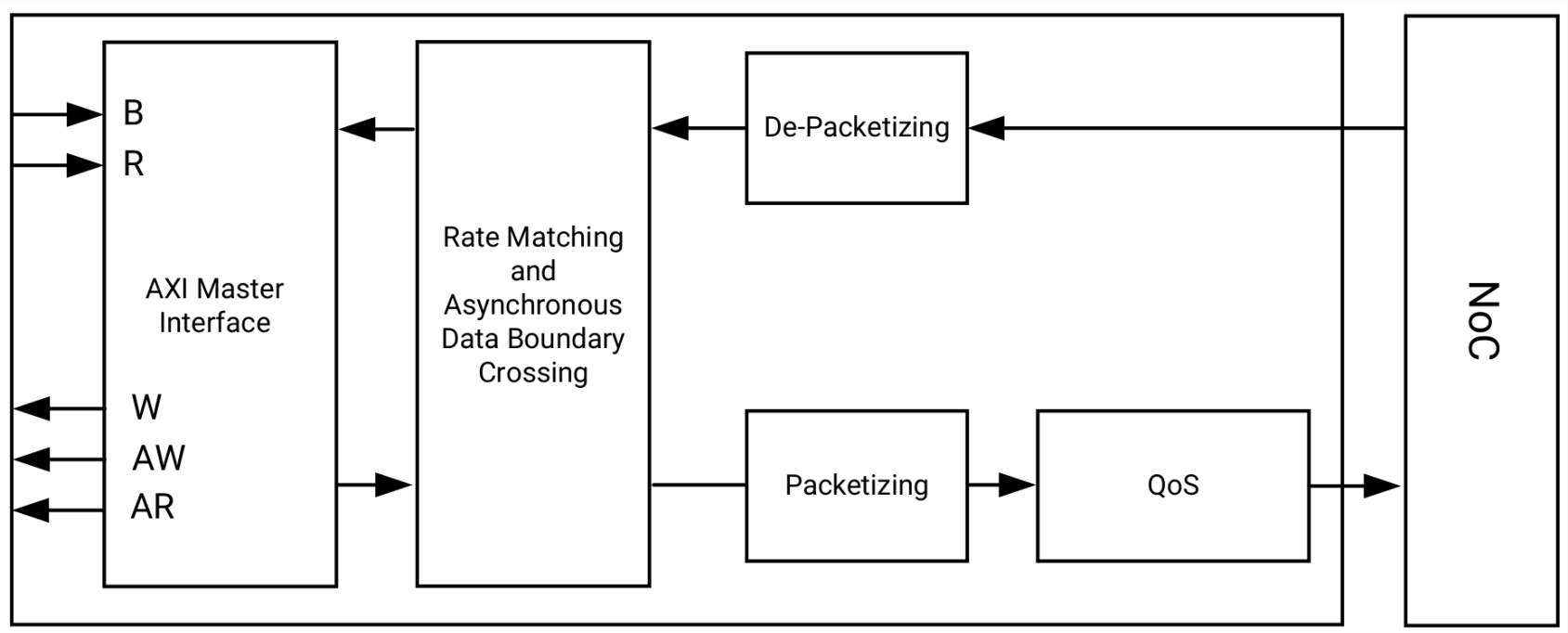
Backup Slides

NoC Master Unit (NMU)



(Figure taken from [1])

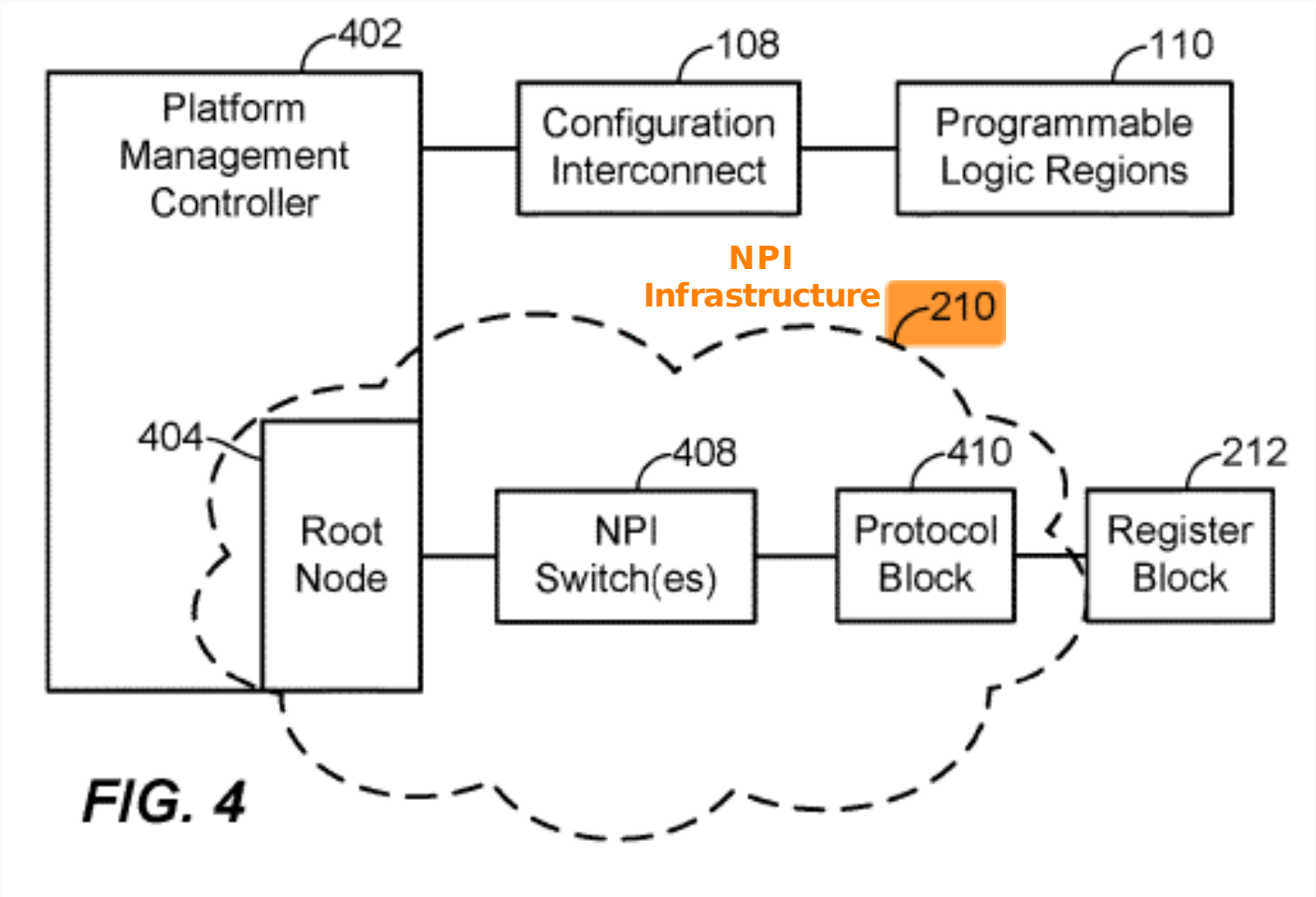
NoC Slave Unit (NSU)



(Figure taken from [1])

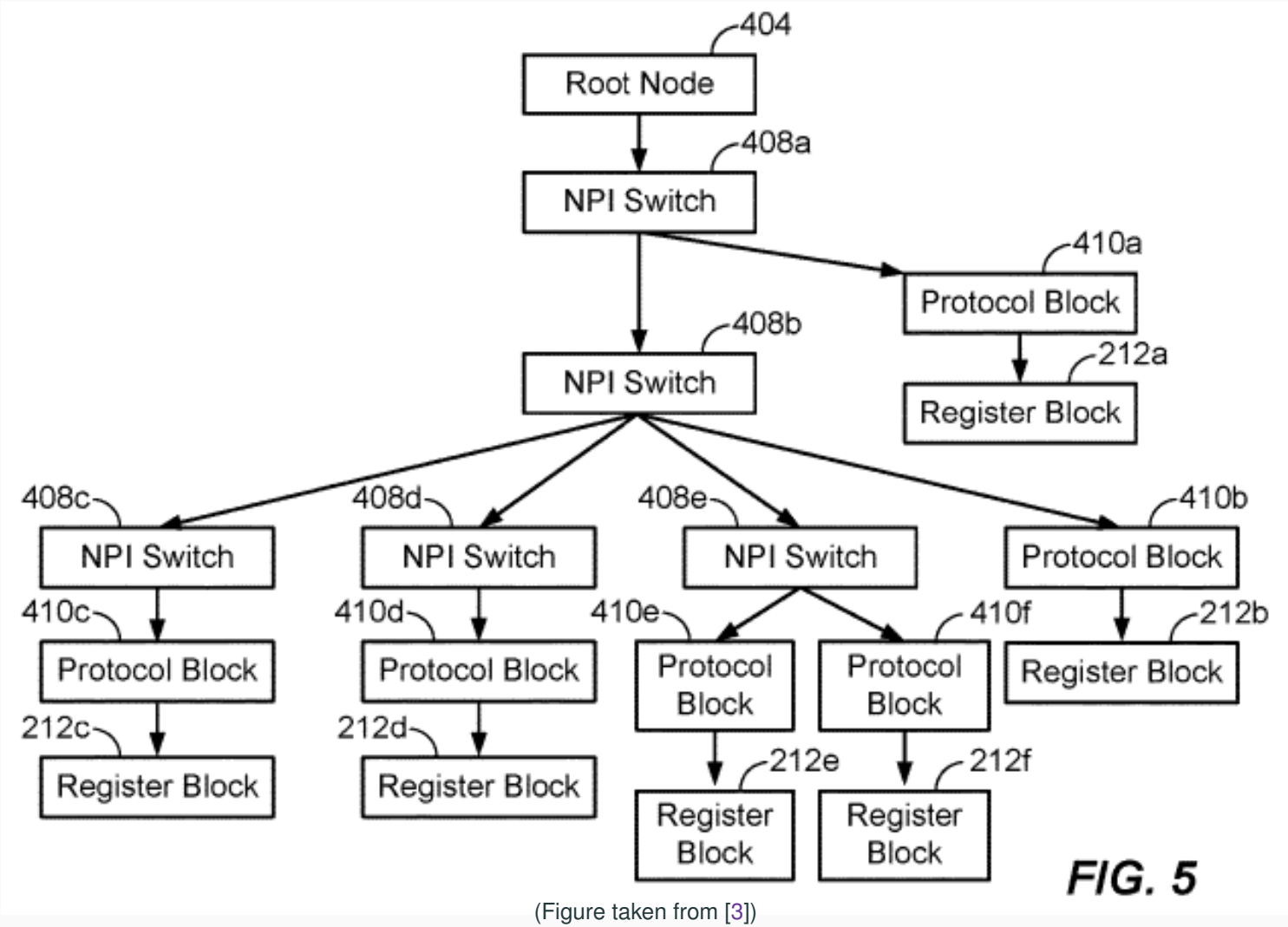
NPI interface/interconnect: The big picture

NPI: *NoC Programming Interface* (a.k.a. *NoC Peripheral Interconnect*). This is an *auxiliary and independent* NoC (32-bit RD/WR [5], 300 MHz [6]) to access config./status registers for the main NoC, DDRMCs, MGTs, etc. (block type details on [5], Chapter 21). It's memory-mapped on PMC's processing unit.



(Figure taken from [3])

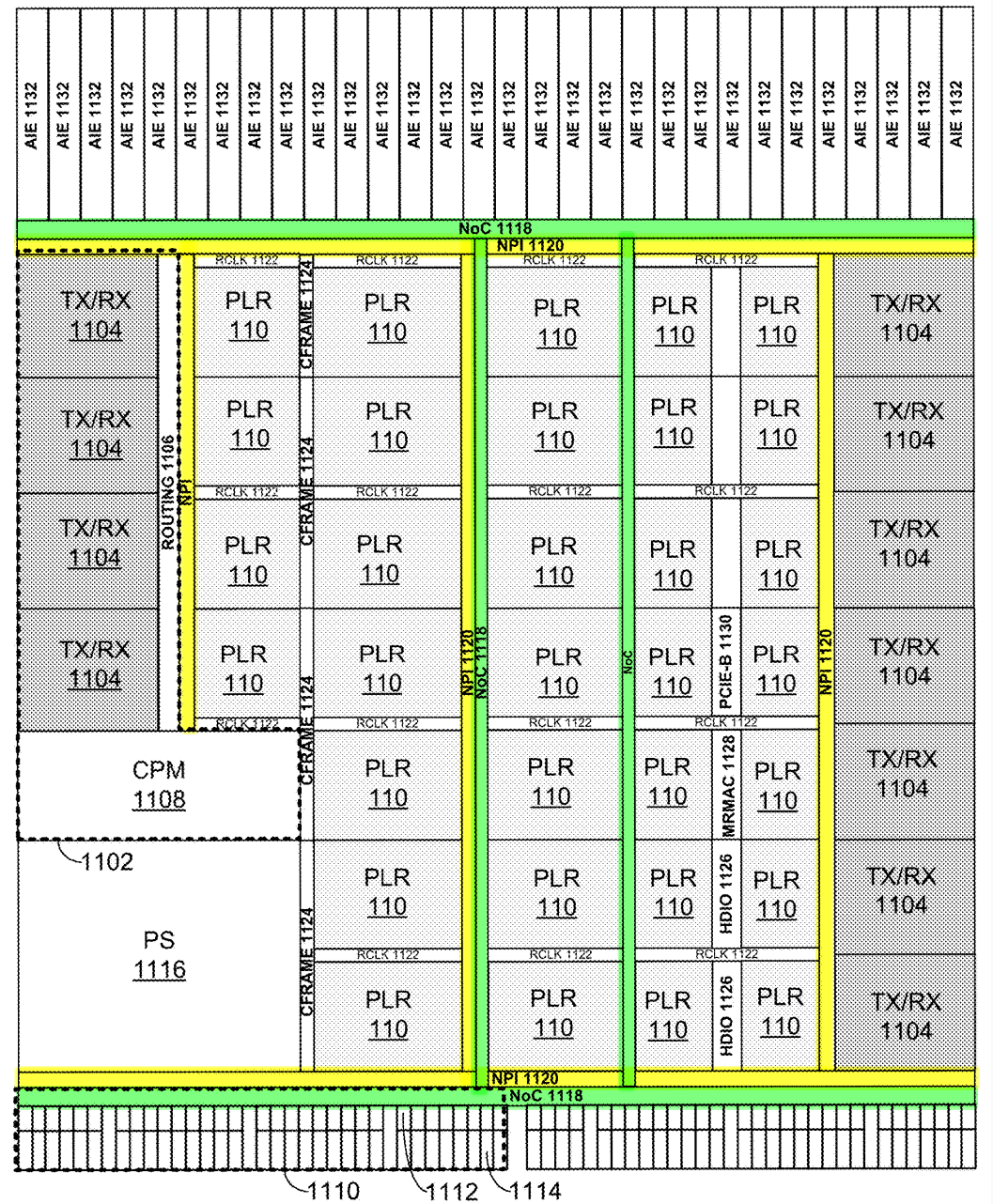
NPI interface tree (independent NoC for accessing main NoC's registers)



(Figure taken from [3])

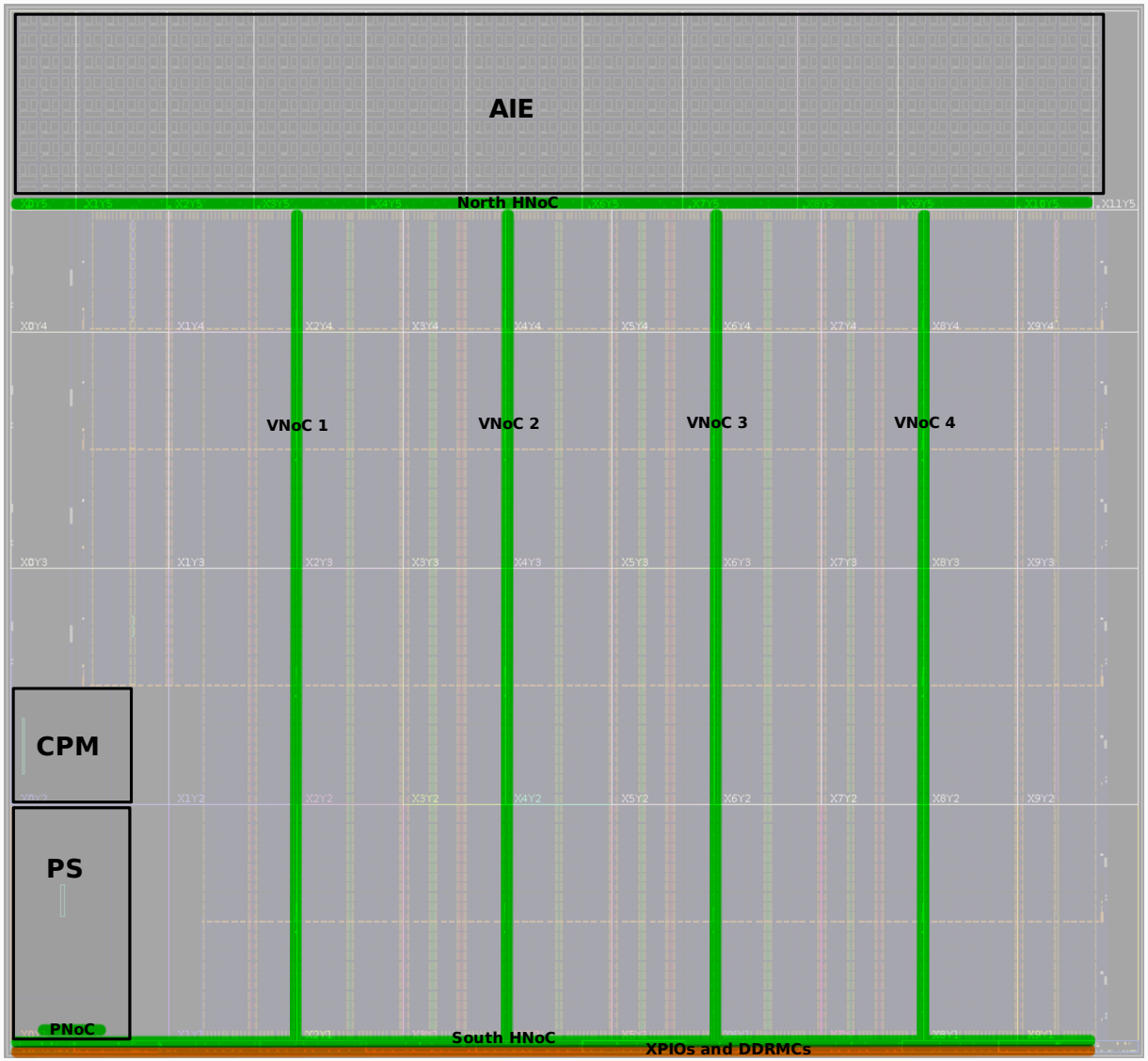
– Paper [7] mentions a "tree-structured peripheral bus" implemented to program the NoC.
– UG [8] says: "This memory (NPI registers) is physically distributed throughout the device".

NPI's place on device's floorplan (as per patent)



(Figure taken from [9])

Features visible in Vivado's device view (vc1902)



– Four VNoC (full) columns clearly identified, each immediately adjacent to an unlabeled column **(NPI?)**.
– Two HNoC rows **(NPI infrastructure there, too?)**.
– "PNoC" block (tightly coupled to the PS) contains an "NPI_NIR" box. This is the **NPI root node** [1].

Number of *visible* NPI (NoC/DDRMCs) registers' bits

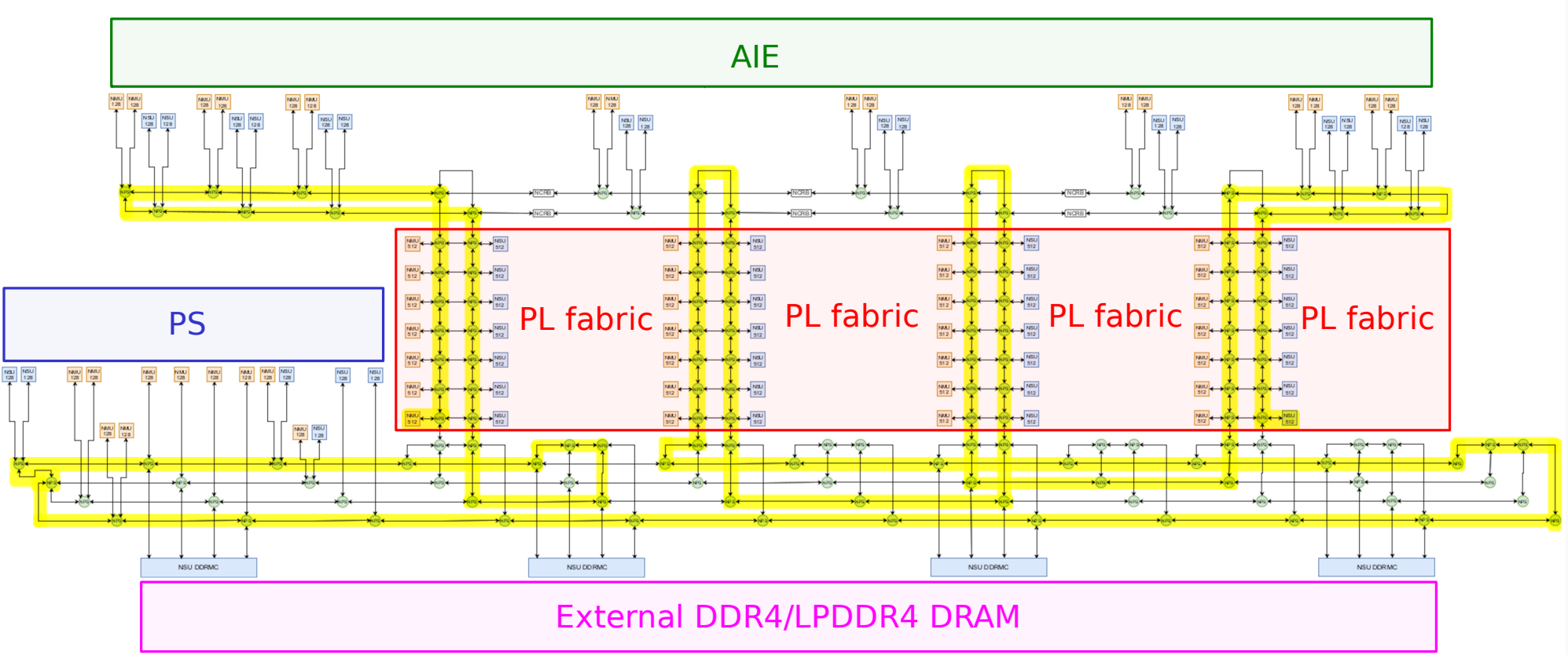
- CMT_MMCM module type: 2688 bits (12 modules; 7 registers/module)
- CMT_XPLL module type: 5376 bits (24 modules; 7 registers/module)
- DDRMC_DDR4_XRAM module type: 399616 bits (4 modules; 3122 registers/module)
- DDRMC_LPDDR4_XRAM module type: 390144 bits (4 modules; 3048 registers/module)
- DDRMC_MAIN module type: 9216 bits (4 modules; 72 registers/module)
- DDRMC_NOC module type: 31744 bits (4 modules; 248 registers/module)
- DDRMC_UB module type: 1408 bits (4 modules; 11 registers/module)
- NOC_NCRB module type: 5760 bits (6 modules; 30 registers/module)
- NOC_NMU module type: 115776 bits (54 modules; 67 registers/module)
- NOC_NPS module type: 210240 bits (146 modules; 45 registers/module)
- NOC_NSU module type: 89600 bits (50 modules; 56 registers/module)

TOTAL NUMBER OF BITS: 1,261,568

NOTE:

1. Register width is 32-bit in any case.
2. Above numbers were taken from [4].

Logical NoC with large number of NPSs (highlighted)



- Each switch using only 1 input and 1 output port; no switch repeated.
- To achieve this, we'd need to force the following settings:
  - Specify the particular NPS instances to be used;
  - On each NPS, specify the routing table for the input port in use.