IBM Network Processor, Development Environment and LHCb Software

LHCb Readout Unit Internal Review

July 24th 2001

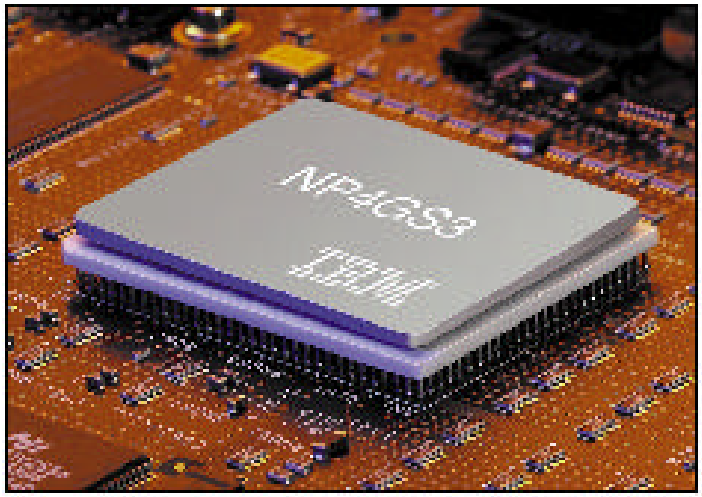
Niko Neufeld, CERN

Outline

- IBM NP4GS3 – Architecture
- A Readout Unit based on the NP4GS3
- Dataflow in the NP based RU
- Sub-event building algorithms
- Software Development Environment
- Performance of Sub-event building
- Calibration of simulation results with measurements on Reference Kit Hardware

IBM NP4GS3

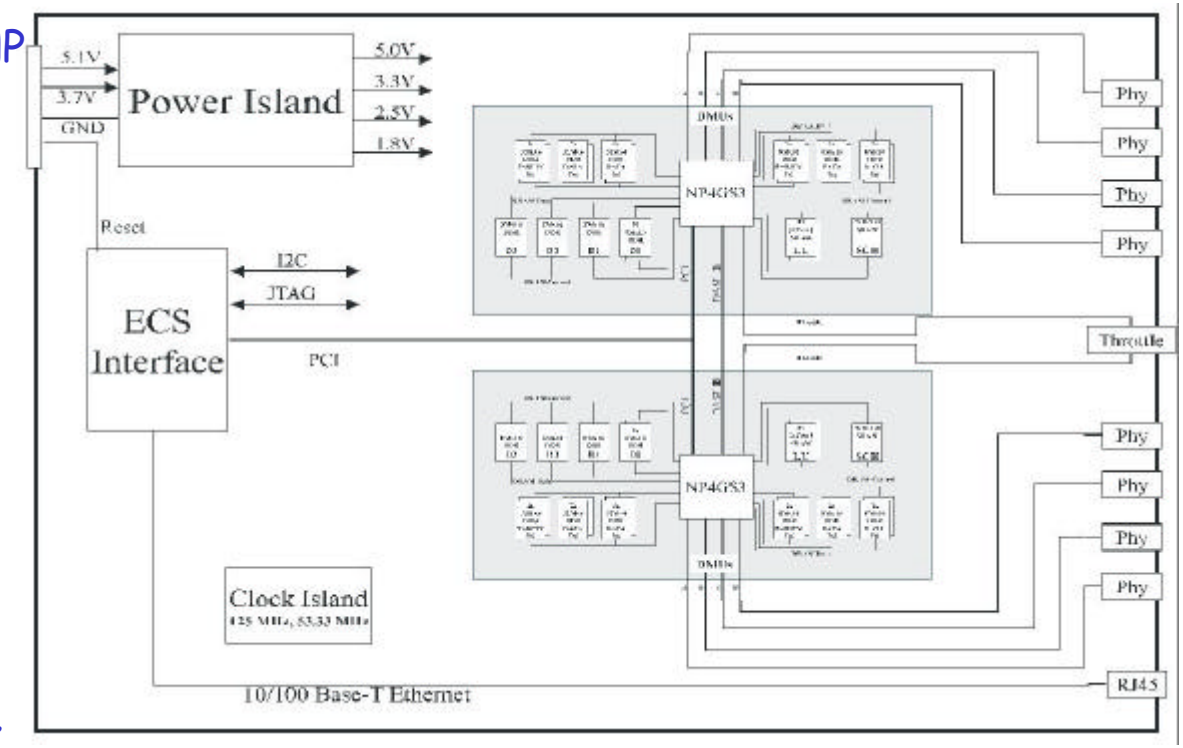
- 4 full duplex Gigabit Ethernet MACs
- 16 Processors / 2 Hardware threads @ 133 MHz ➔ 2128 MIPS to handle up to 4.1×10^6 packets/s
- 128 kB on-chip input buffer, up to 128 MB DDR RAM output buffer
- 2 x Switch Interface ("DASL") @ 4 Gb/s
- Embedded PPC 405
- In production since beginning of this year (R1.1)



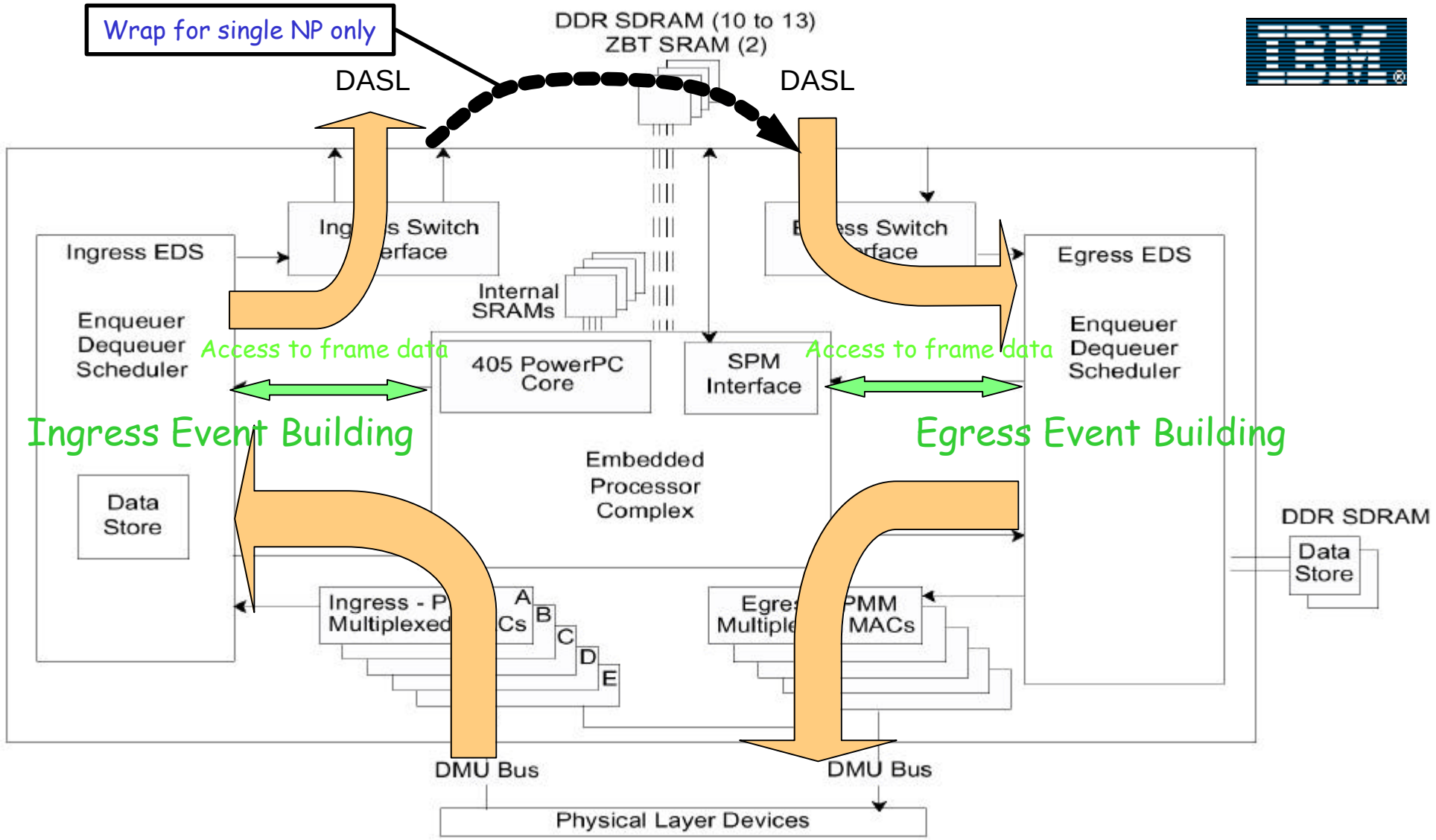
Readout Unit based on NP4GS3

- 1 or 2 Mezzanine Cards containing each
  - 1 Network Processor
  - All memory needed for the NP
  - Connections to the external world
    - PCI-bus
    - DASL (switch bus)
    - Connections to physical network layer
    - JTAG, Power and clock
- PHY-connectors
- L0 Trigger-Throttle output
- Power and Clock generation
- LHCb standard ECS interface (CC-PC) with separate Ethernet connection

Board Block Diagram



Data flow in the NP4GS3



Sub-event merging software

- Sub-event building is the main task of the software running on the NP4GS3 when used in a Readout Unit
- Two locations for frame manipulation (ingress & egress) insinuate two different algorithms with different advantages and possible fields of application
- Ingress event building for high frequency up to 1.5 MHz, small frames (~ 64 bytes)
- Egress event building for large frames (up to 9000 bytes), or fragments spanning multiple frames

Ingress Event Building

- ☺ On chip memory (no wait cycles!)
- ☺ Memory buffers organized via descriptors
- ☺ Code is more "stream-lined", because copying is done on linear memory
- ☹ Only 128 kB of memory
- ☹ Multiple frames more difficult, because they have to be sent in order over the DASL

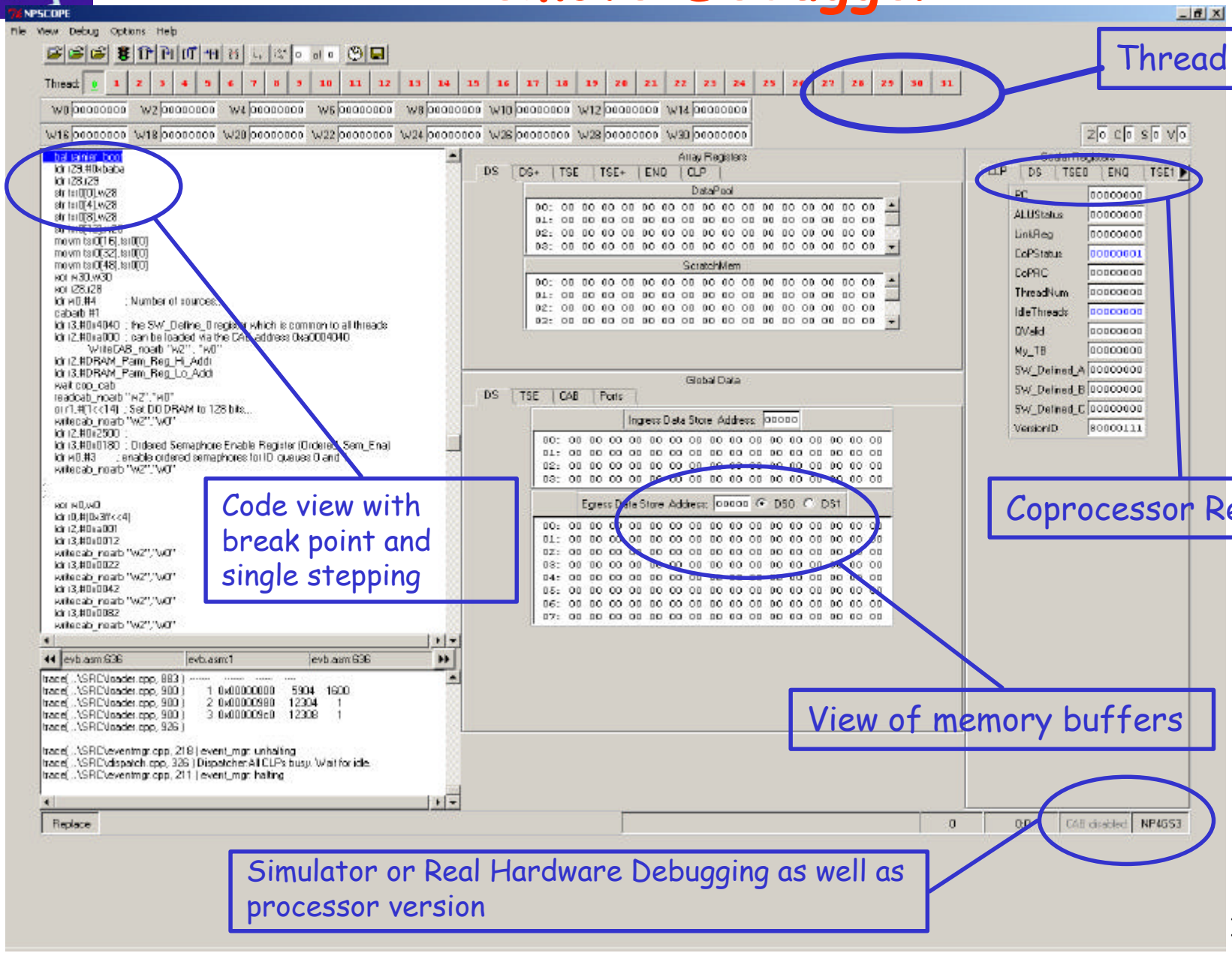
Egress Event Building

- ☺ 64 MB of external buffer space (Access to memory over 128 bit wide bus). Two copies of 64 MB each, for increased throughput.
- ☺ Only contested resource is the memory bus.
- ☺ Multiple frames are handled easily.
- ☺ For larger fragments only part of the data (ideally 1/8) need to be read.
- ☹ Memory is external (wait cycles!)
- ☹ Buffer data structure is awkward (chunks of 2 x 58 bytes)

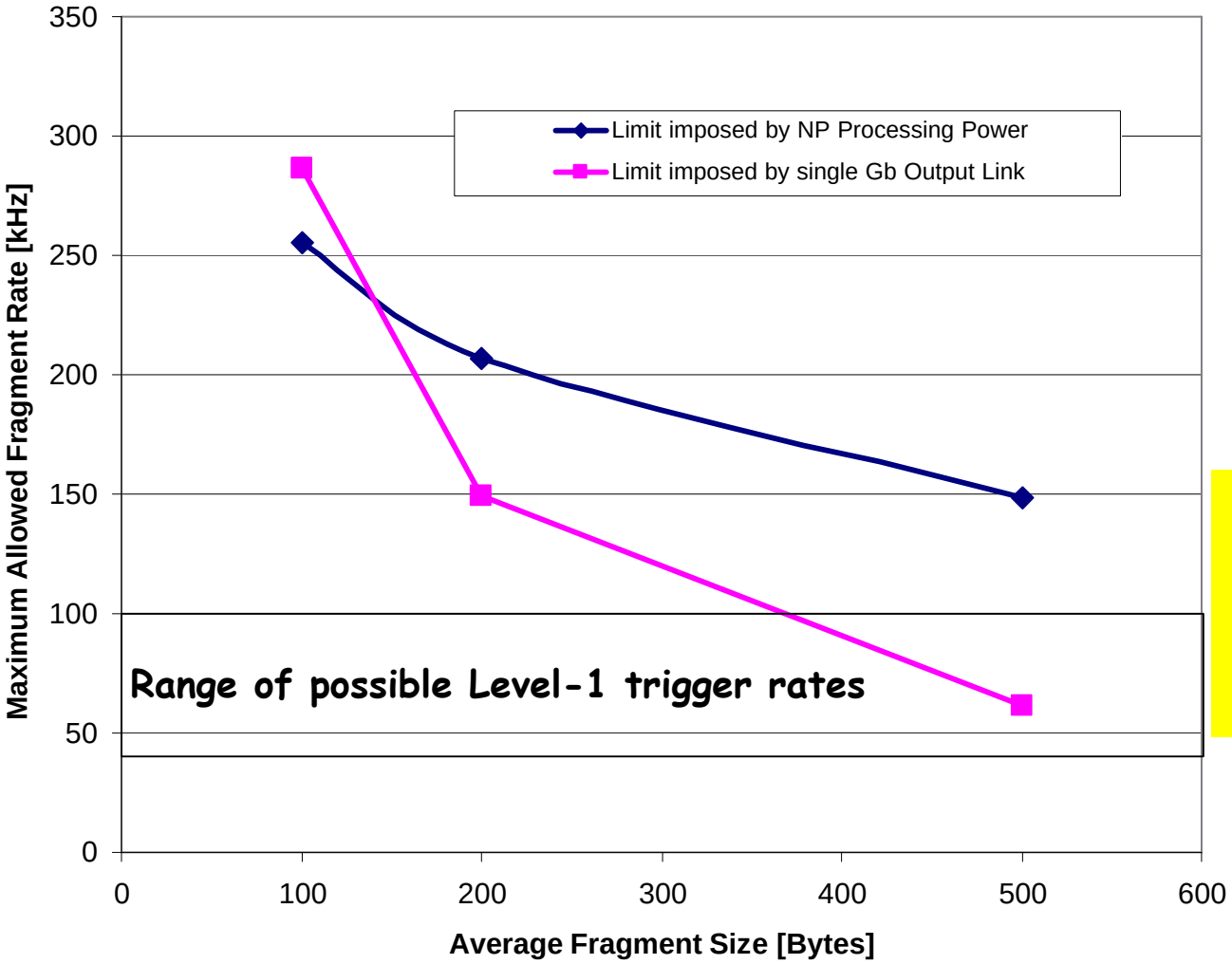
Software Development Environment

- Development software consists of: Assembler, Debugger, Simulator and Profiler (the debugger can either be run on the simulator or connect to real hardware via a Ethernet to JTAG interface ("RISC Watch") attached to the NP4GS3)
- Rich set of documentation
- Many examples, such as complete routing software, are available

User interface of Simulator and Remote Debugger

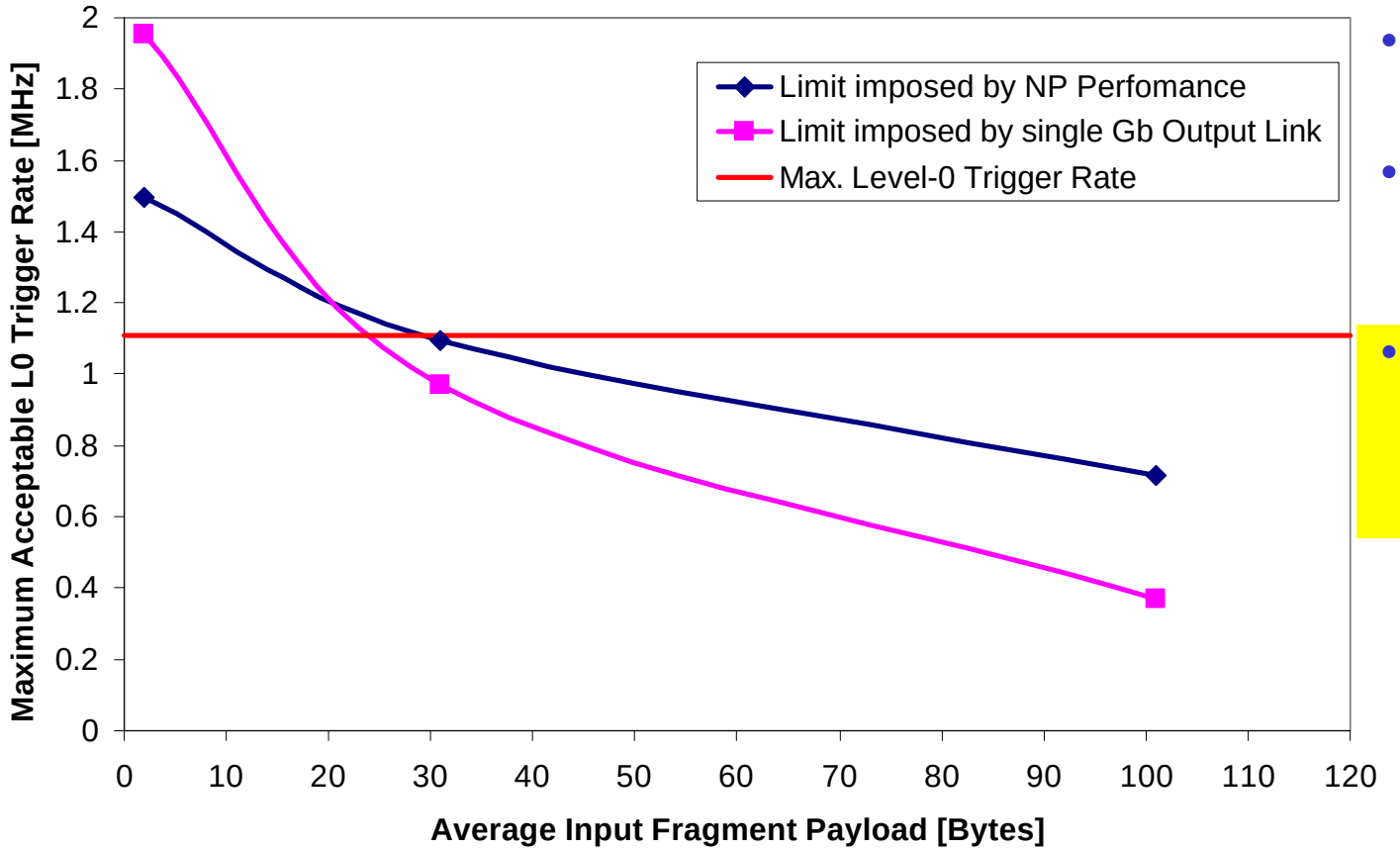


Performance for 4:1 Egress Event-Building



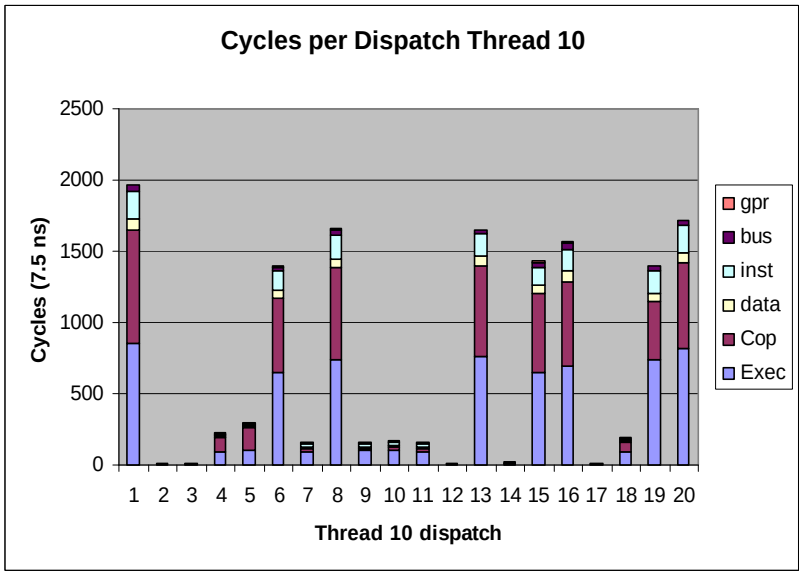
- Only 16 out of 32 threads simulated
  - hence blue line is highly pessimistic
  - Should increase by at least 50% for 32 threads
- At any frame size the performance of the RU is limited only by the output link speed of 125 MB/s

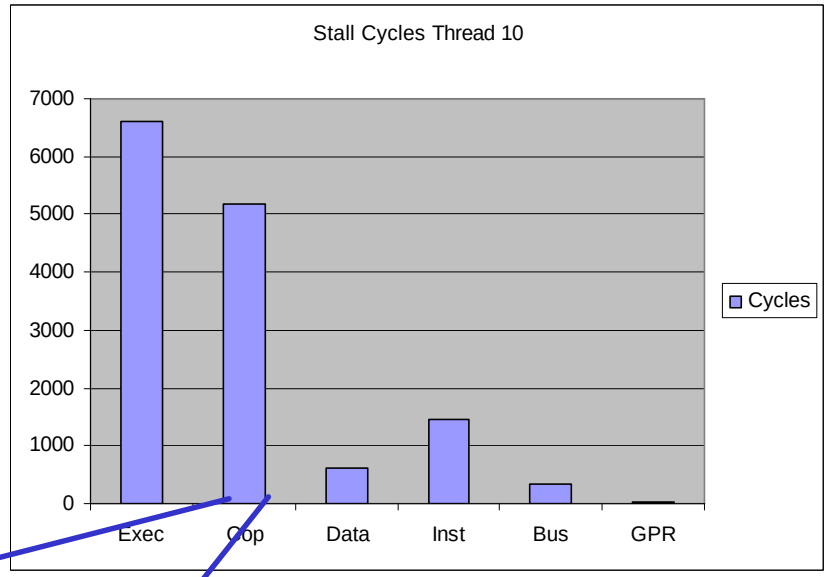
Performance for 3:1 Ingress Event- Building

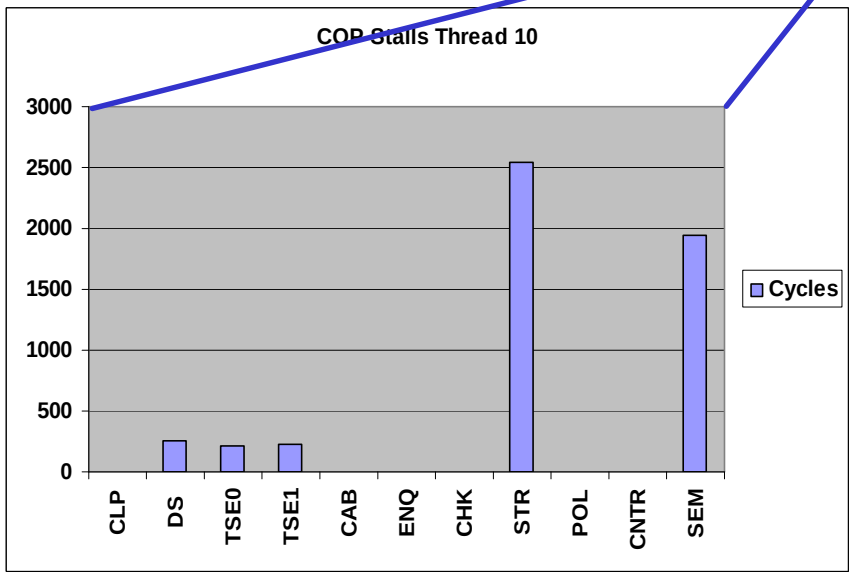


- Only 16 out of 32 threads simulated
- Overall performance limited by single output link speed only
- NP performance sufficient for any fragment size

Detailed Profiler Information



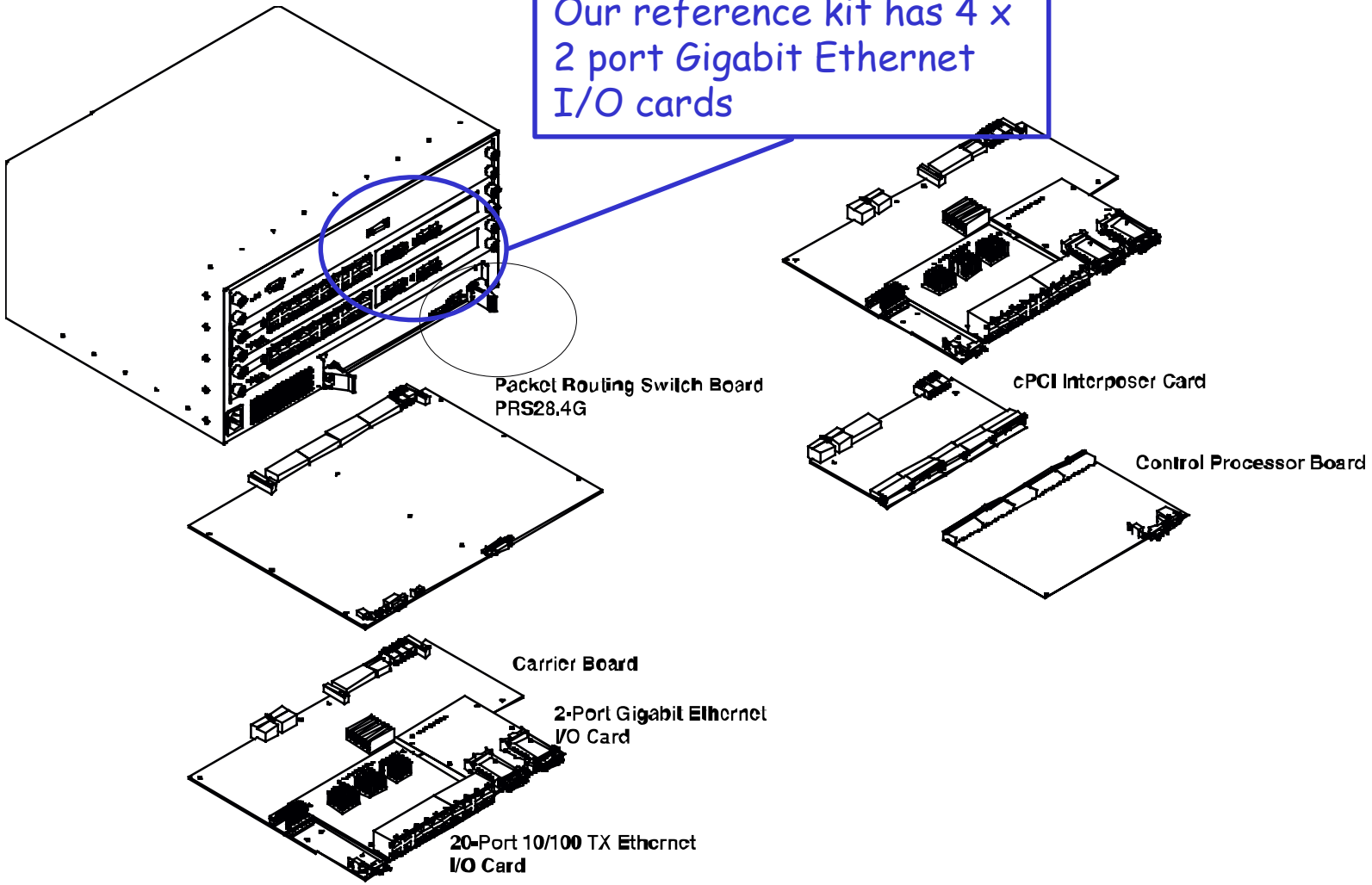




Details of stall ('enforced NOP') cycles
They are assigned to the coprocessor, which controls access to the resource in question (and is causing the stall in that case)

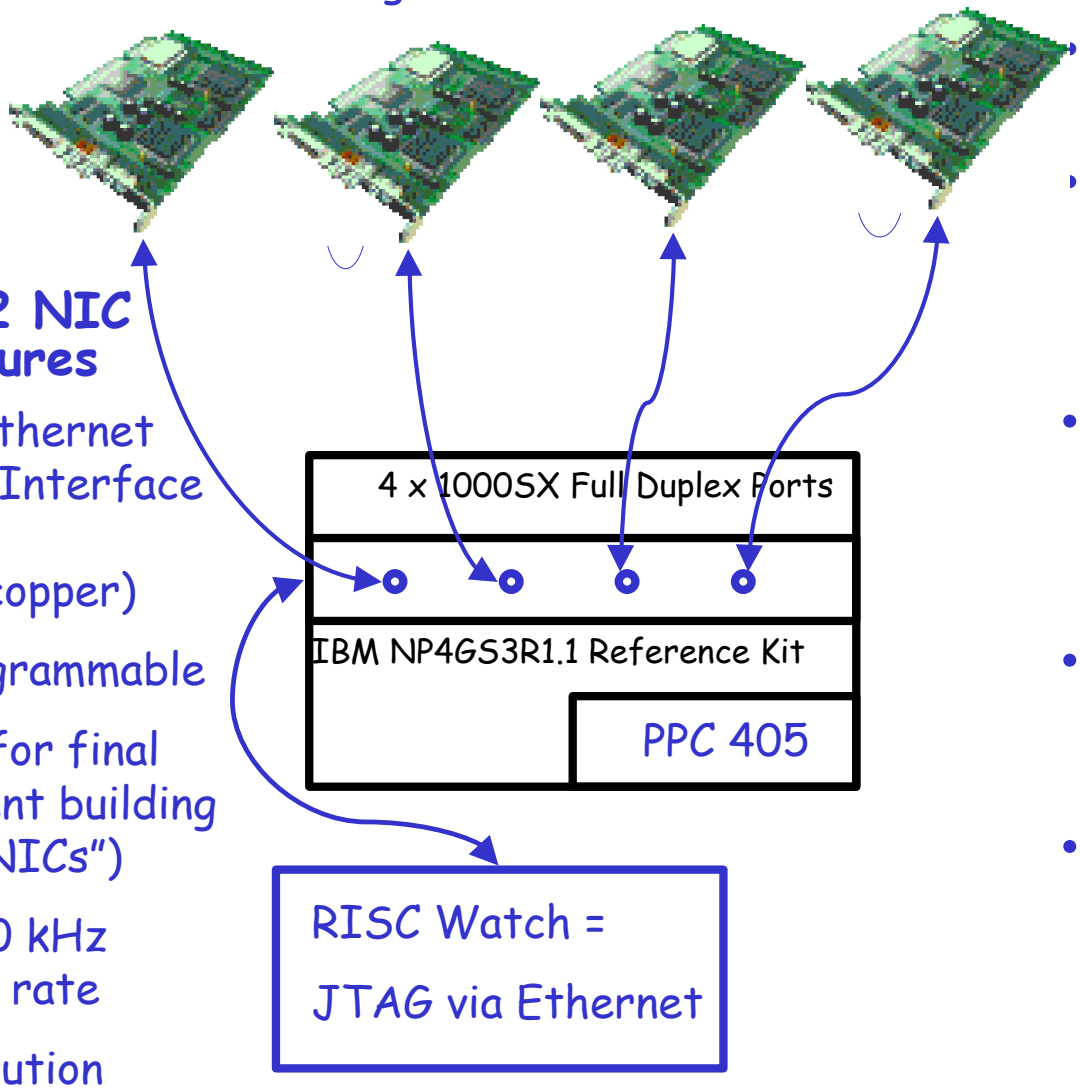
Reference Kit Hardware

Our reference kit has 4 x 2 port Gigabit Ethernet I/O cards



Test Set-up

4 x Tigon2 NIC 1000SX



Tigon2 NIC Features

- Gigabit Ethernet Network Interface Card (optical/copper)
- Fully programmable
- Baseline for final stage event building ("Smart NICs")
- Up to 620 kHz fragment rate
- 1 μs resolution timer

Measurement Procedure

- Download code into NP4GS3 via RISC Watch (JTAG)
- Send special frame to NP4GS3 to trigger synchronization frame being sent to Tigons
- Tigons start sending fragments. They add their internal time-stamp to each frame (1 μs resolution).
- NP4GS3 processes and adds its internal time-stamp (1 ms resolution).
- Tigons receive frames and calculate elapsed time

Calibrating the simulation by measurements

- Event building with a single thread enabled and 4 sources
- Event building with a single source and multiple threads (16) enabled
- With release 1.1 version of the NP4GS3 can unfortunately not run easily multiple sources on multiple threads (no semaphore coprocessor)

Results from Measurements (Ingress Event-Building)

- Round-trip time (Tigon-out Tigon-in: 1.7 ms per fragment (frame of 60 Bytes)
- Simulation says that 600 ns/fragment are used in the NP4GS3. This is not really measurable with our setup.

	Measurement [µs/fragment]	Simulation [µs/fragment]
1 source 1 thread	6.6(4.9)	4.9
4 sources 1 thread	4.5(2.8)	3.2
1 source 16 threads	1.7 (0.0)	0.5

NOTE: Since the system is pipelined, times smaller than the intrinsic overhead of the Tigon, (i.e. 1.7 ms), cannot be measured accurately!

→We can trust the timing results obtained with the simulator.

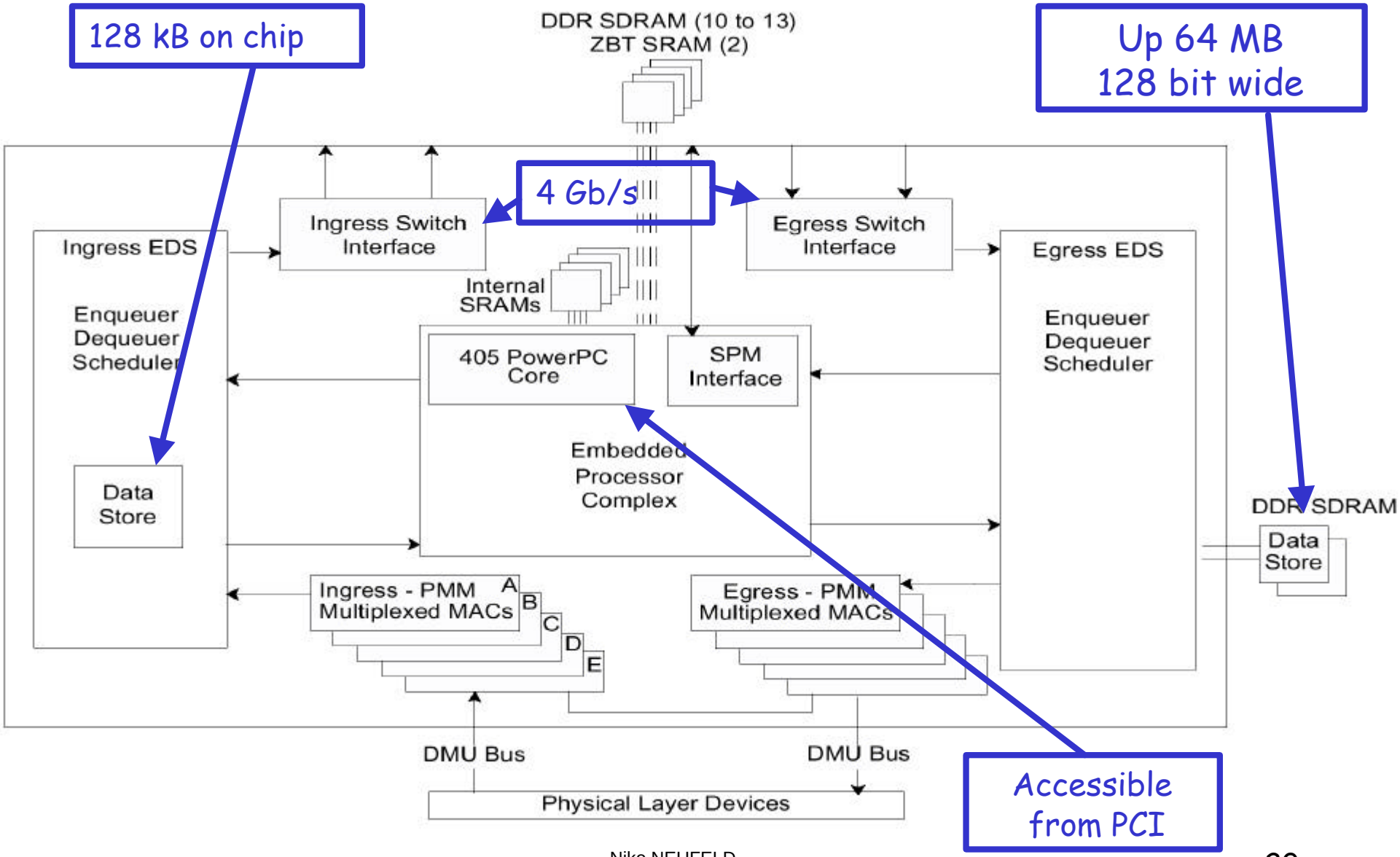
Conclusion

- The software development environment makes the full power of this complex chip available to the software developer
- Two sub-event merging codes for large fragments @ rates of a few 100 kHz and small fragments @ rates of 1 MHz have been developed and benchmarked using simulation.
- The results for the more demanding ("ingress") of the two has been verified using the IBM NP4GS3 reference kit hardware
- The simulation results show that the performance requirements on a readout unit are fully met an NP4GS3-based implementation.

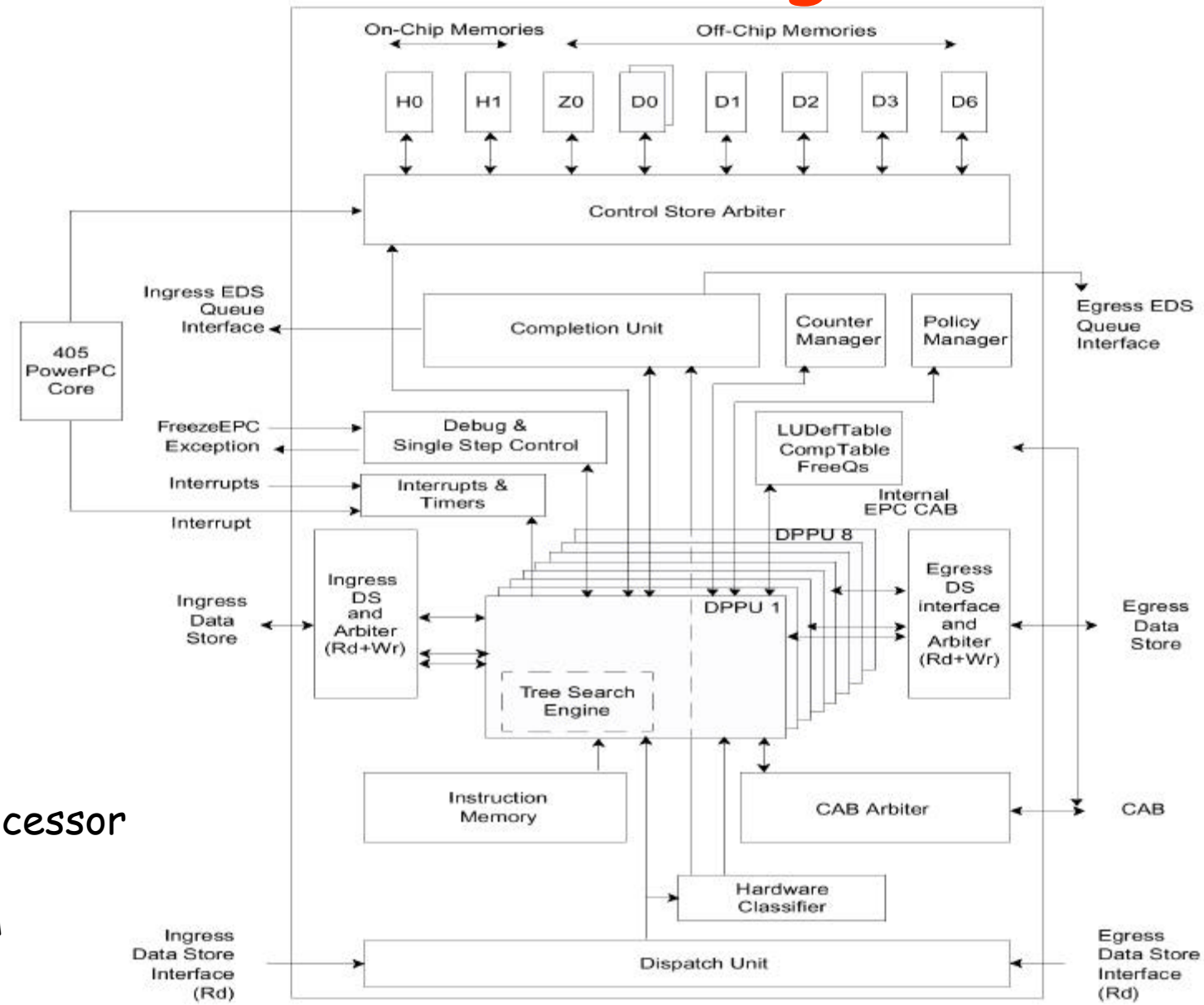
Future Work

- With the upgrade of the NP4GS3 reference kit to version 2.0 of the processor verify (again) code for egress and ingress event building
- With more Tigon NICs and faster fragment generation code, test also e.g. 7 to 1 multiplexing.
- Develop and test layer 2 switching application (much simpler than event building due to static routing tables)
- Develop code for the communication between Linux operated embedded PPC 405 and the CC-PC

NP4GS3 Architecture

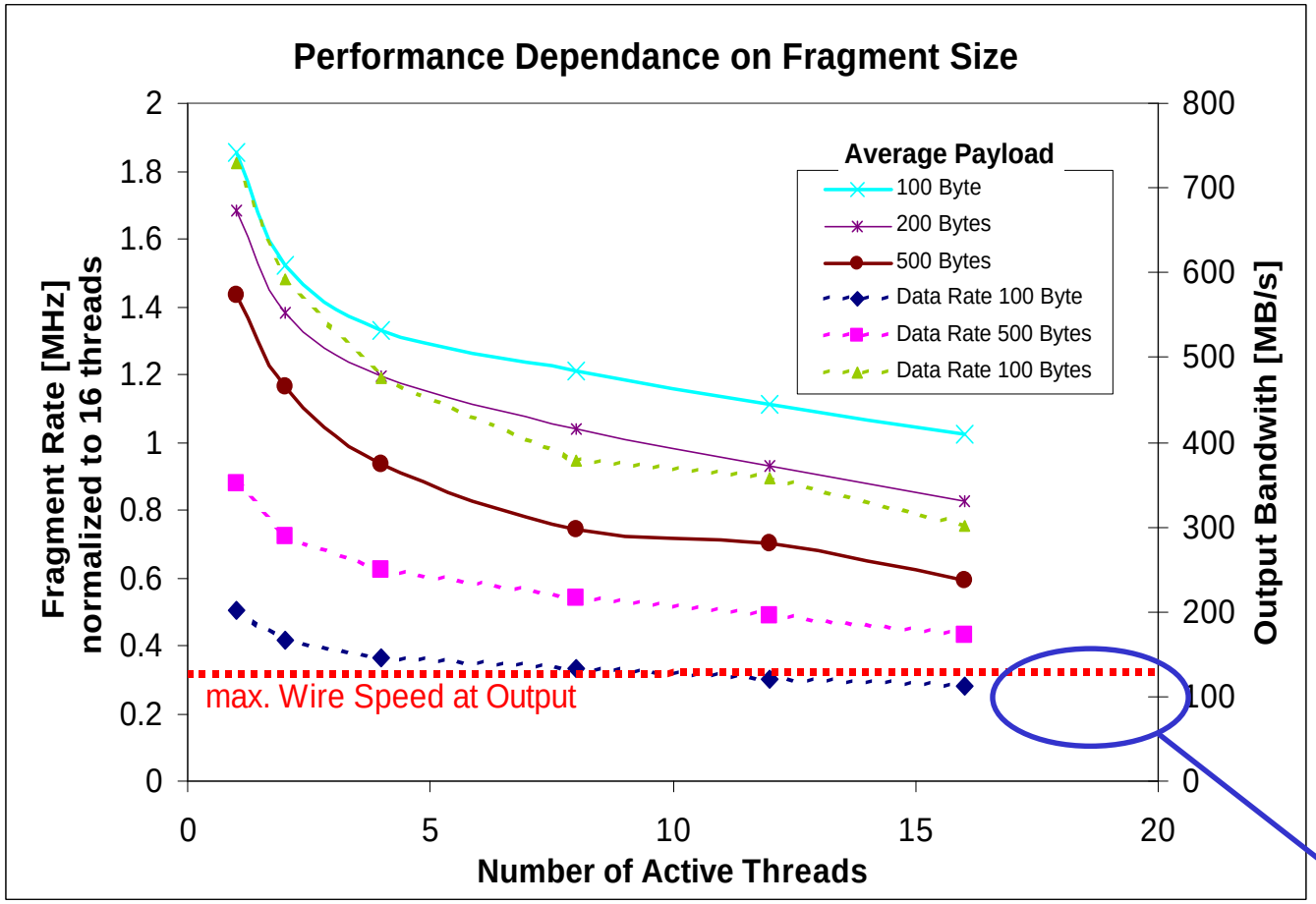


EPC Block-Diagram



IBM NP4GS3 Embedded Processor Complex (EPC) Block-Diagram

Performance for 4:1 Egress Event-Building



Throughput as a function of the number of active threads and the payload.

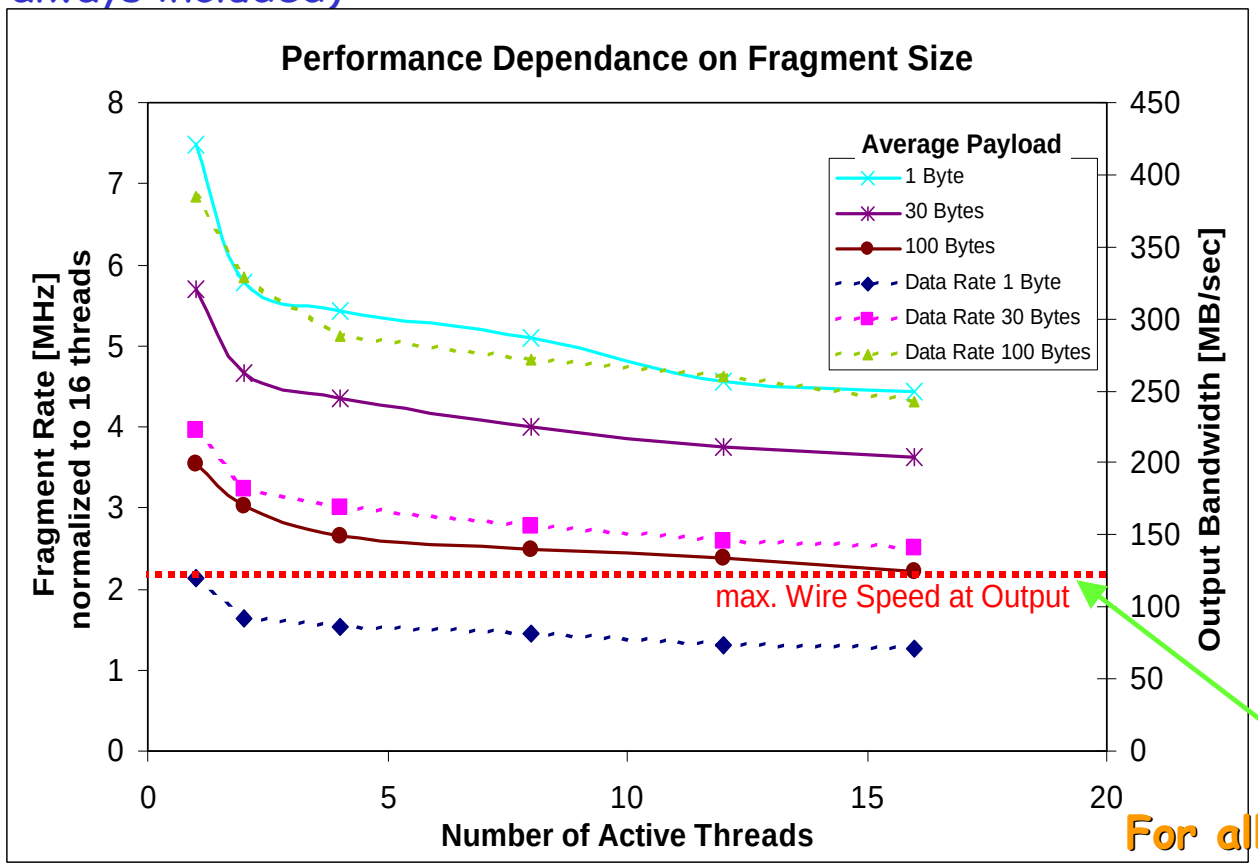
The throughput is normalised to the one achieved with 16 threads for comparability

Contention for resources such as locks and memory prevents linear scaling

Limit of Gigabit Ethernet 125 Mbyte/s

Performance for 4:1 Ingress Event- Building

Fragment rate and output bandwidth as a function of active threads for various amounts of payload. (A 36 byte transport header is always included)



64 bytes = 1.9 MHz

For all practical working points beyond wire-speed