Software Driven Readout Unit

B. Jost & N. Neufeld

Readout Unit (RU)

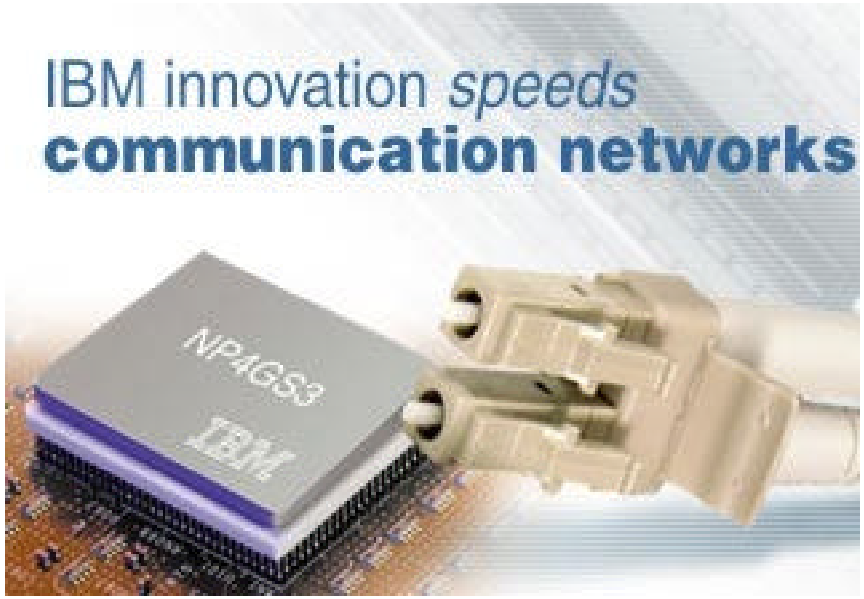
- Receives data from one or more front end links and assembles the fragments belonging to each event
- Complete fragments are then sent out to the readout Network

Requirements for RU

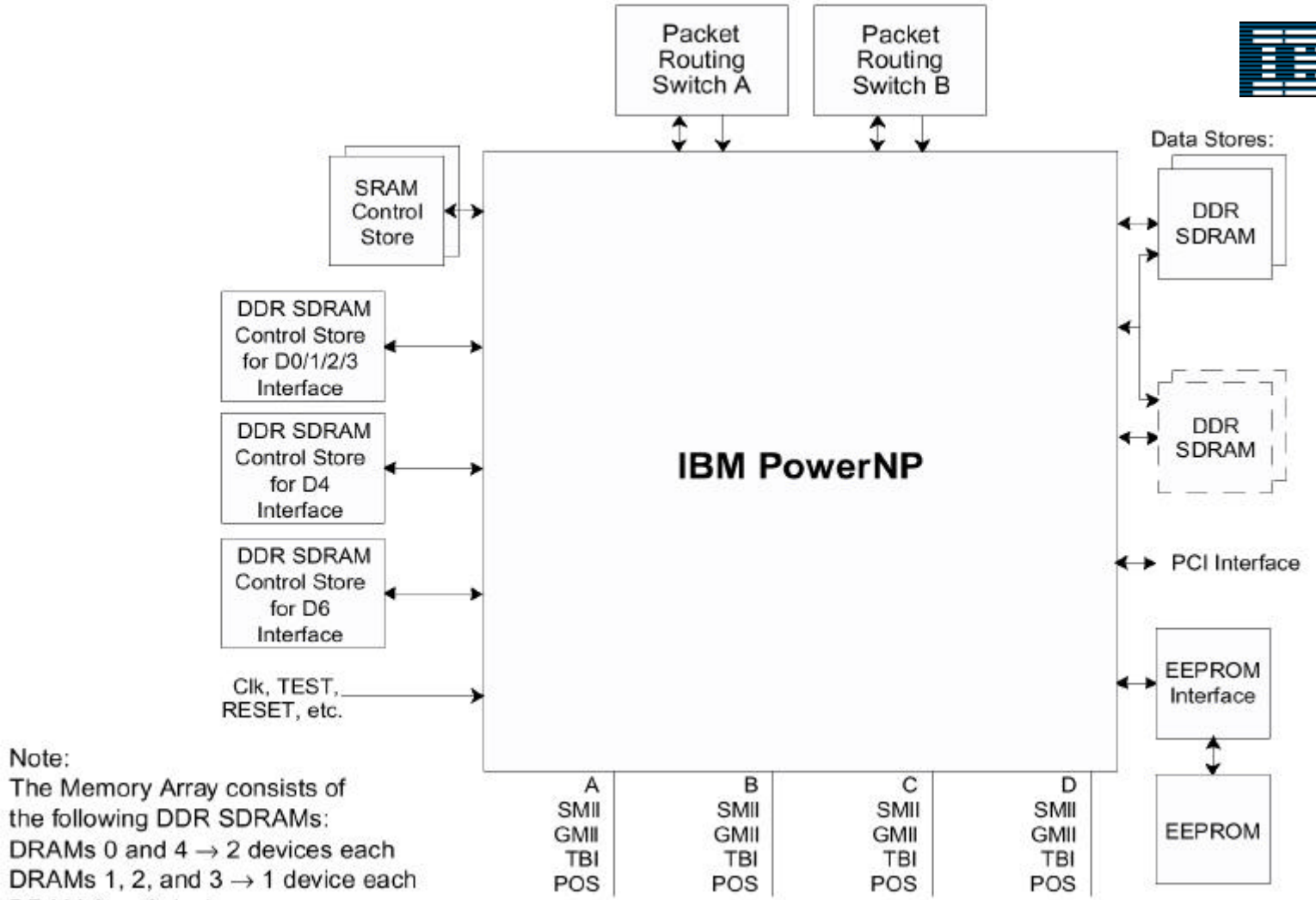
- Handle up to 4 input links at ~1 Gb/s each
- Write out data to the readout network, at a maximum rate of ~ 1 Gb/s (e.g.: Gigabit-Ethernet, Myrinet)
- Provide some buffering for short network congestions
- Eventually raise a throttle signal when unable to evacuate data

A proposal for a RU based on the IBM NP4GS3

- The IBM NP4GS3 is a network processor designed for wire-speed switching/routing and frame manipulation
- It can handle over 4.5 Million packets per second on 4 1-Gigabit full duplex links
- It is fully software programmable



NP4GS3-based System Architecture

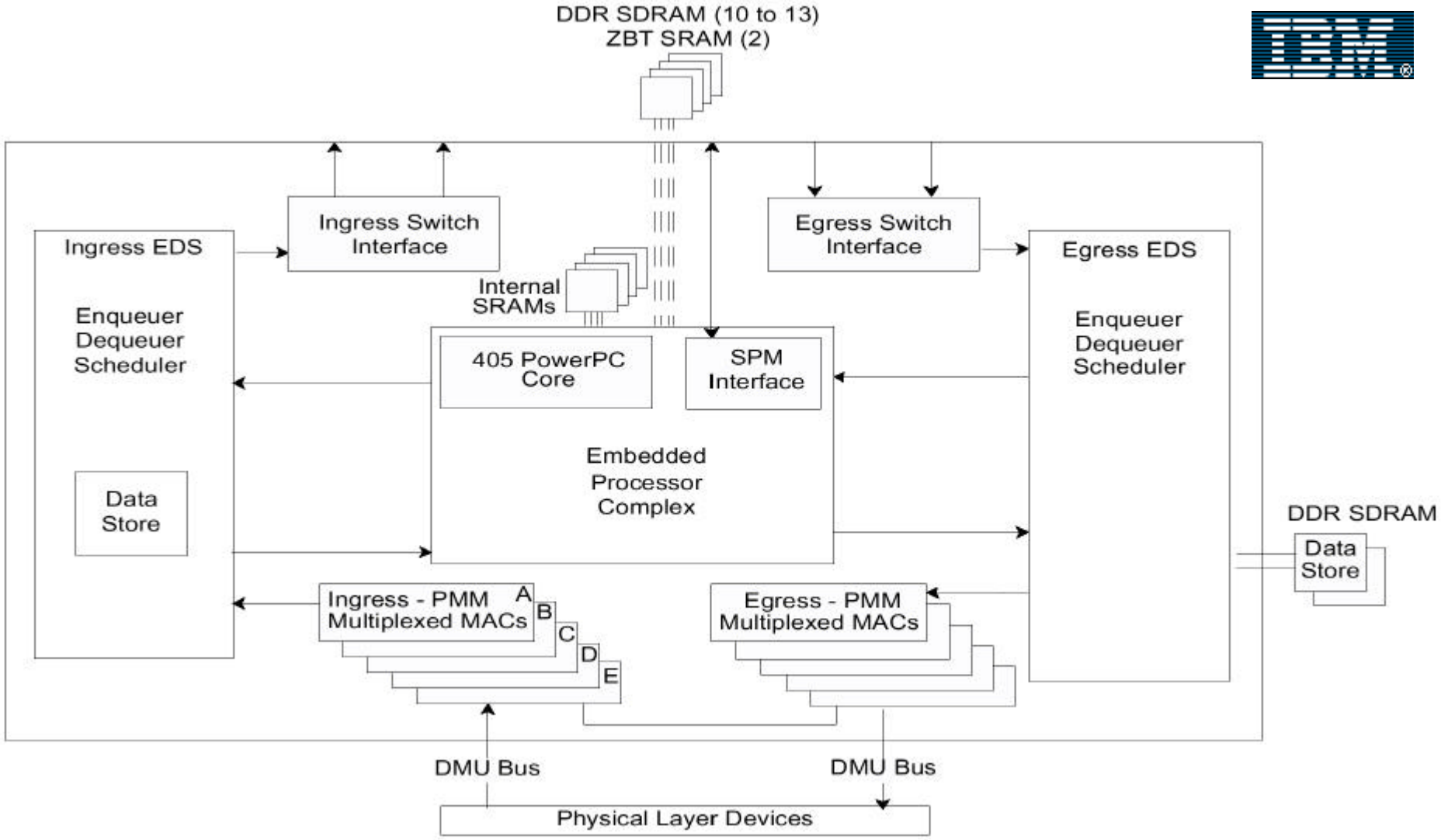


Note:
The Memory Array consists of
the following DDR SDRAMs:
DRAMs 0 and 4 → 2 devices each
DRAMs 1, 2, and 3 → 1 device each
DRAM 6 → 6 devices

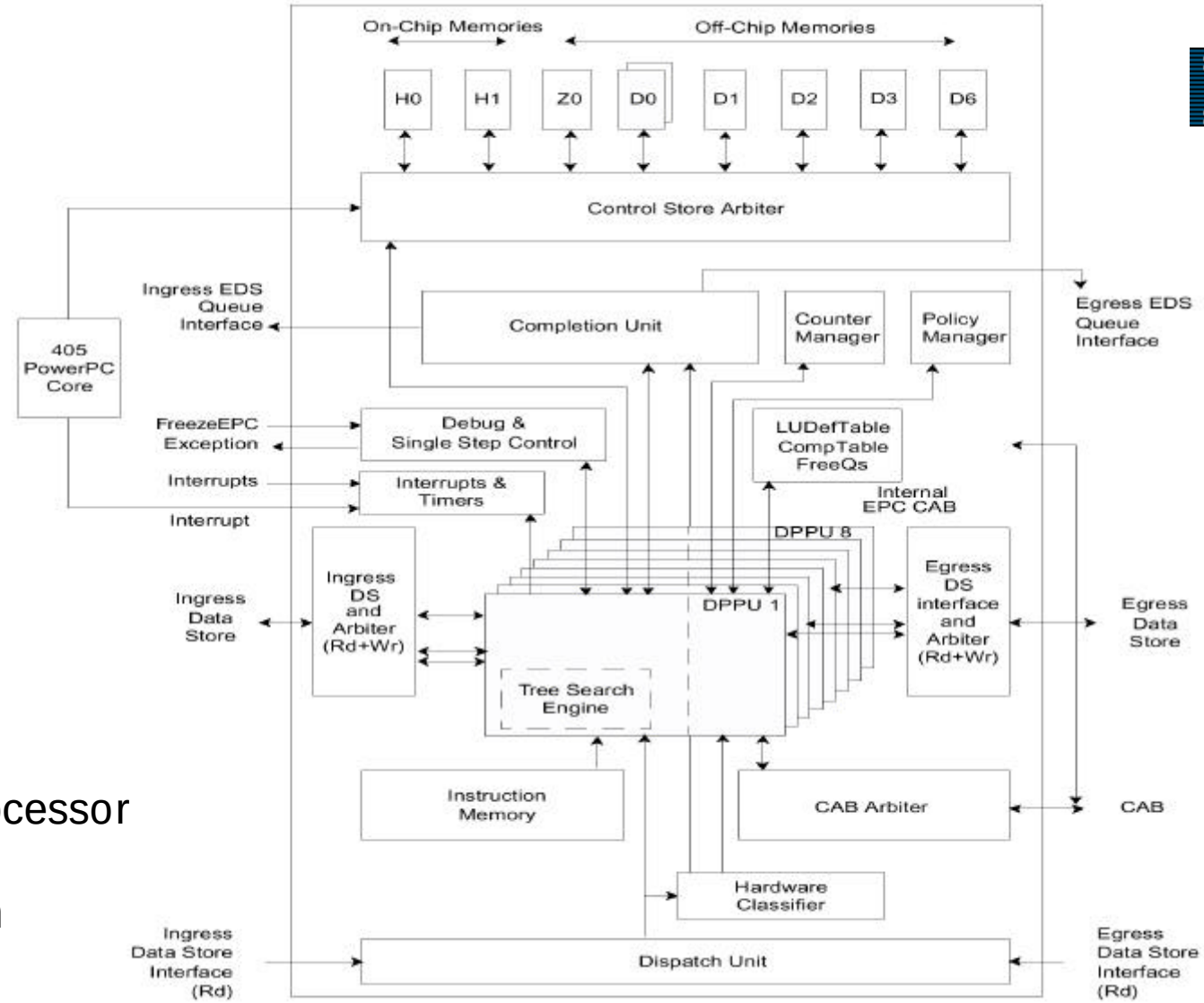
IBM NP4GS3 features

- 16 Special Purpose Processors @133 MHz each with 2 Hardware threads for Datahandling
- Always 2 processors share a set of coprocessors for all important tasks (data moving, table look-up, check-summing, etc.)
- High-speed interface to a switch (DASL)
- 128 kByte of storage on the input (ingress) side
- Up to 64 MByte on the output (egress) side
- Embedded PowerPC for remote management, monitoring, debugging and PCI access

NP4GS3 internal Architecture

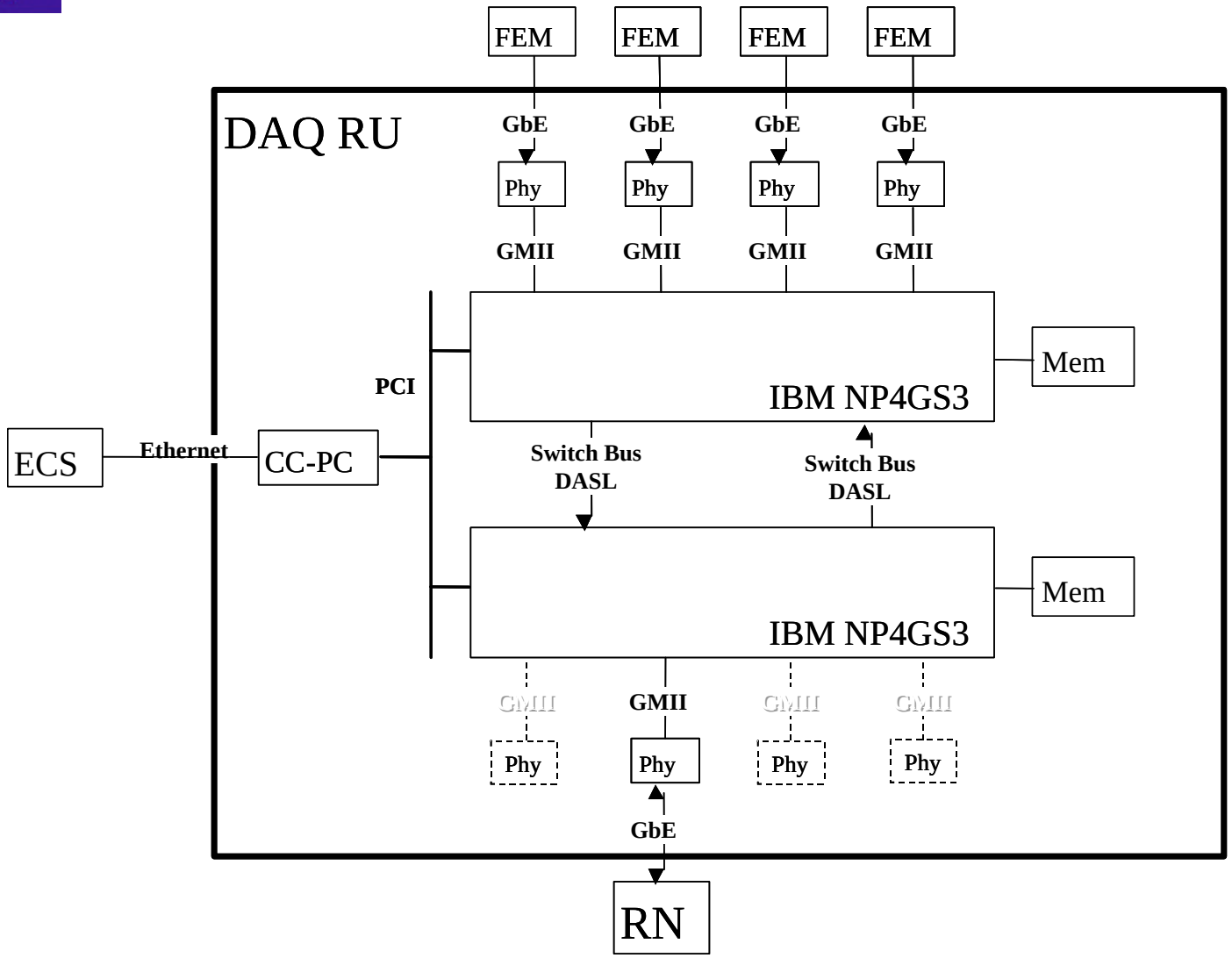


EPC Block-Diagram

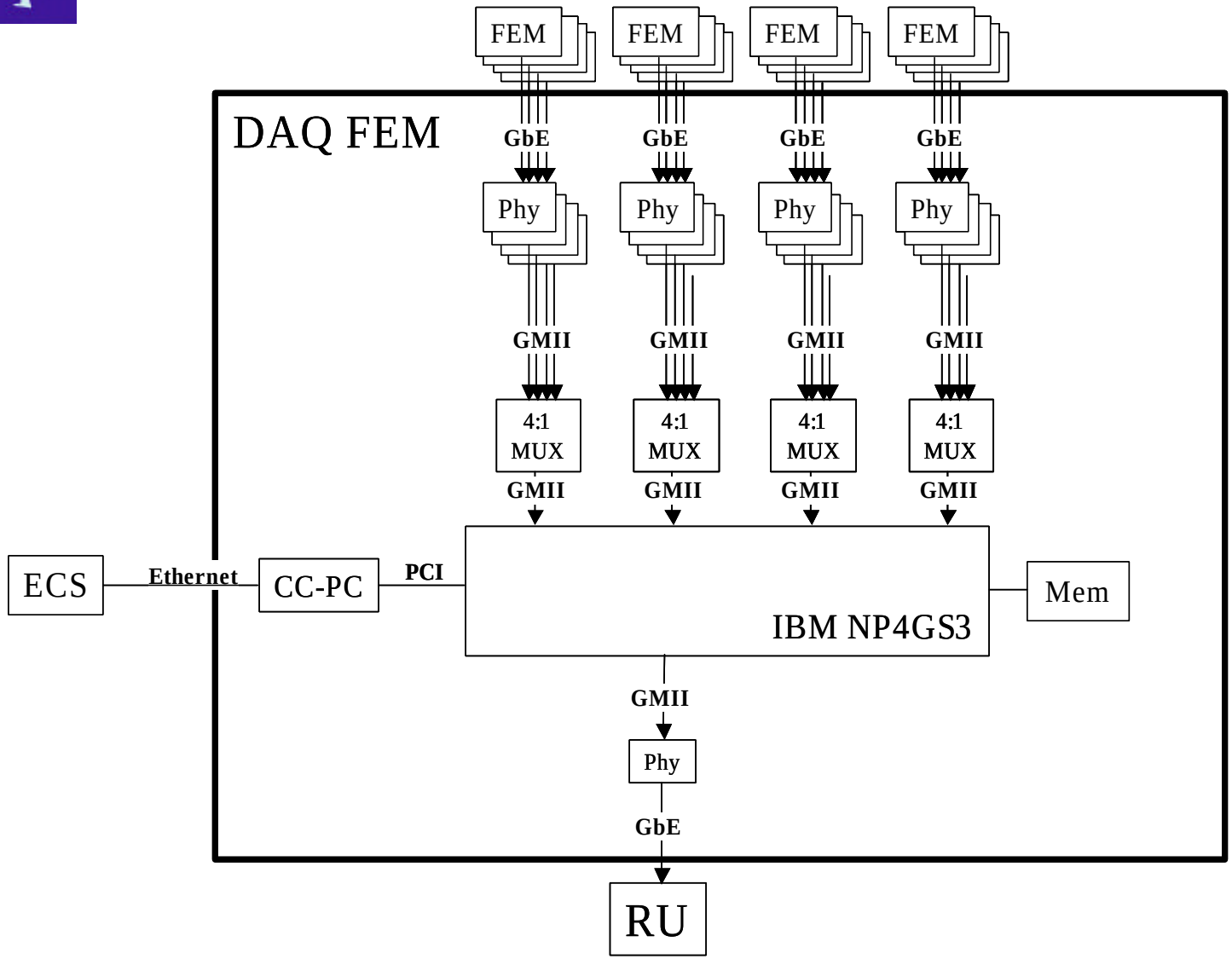


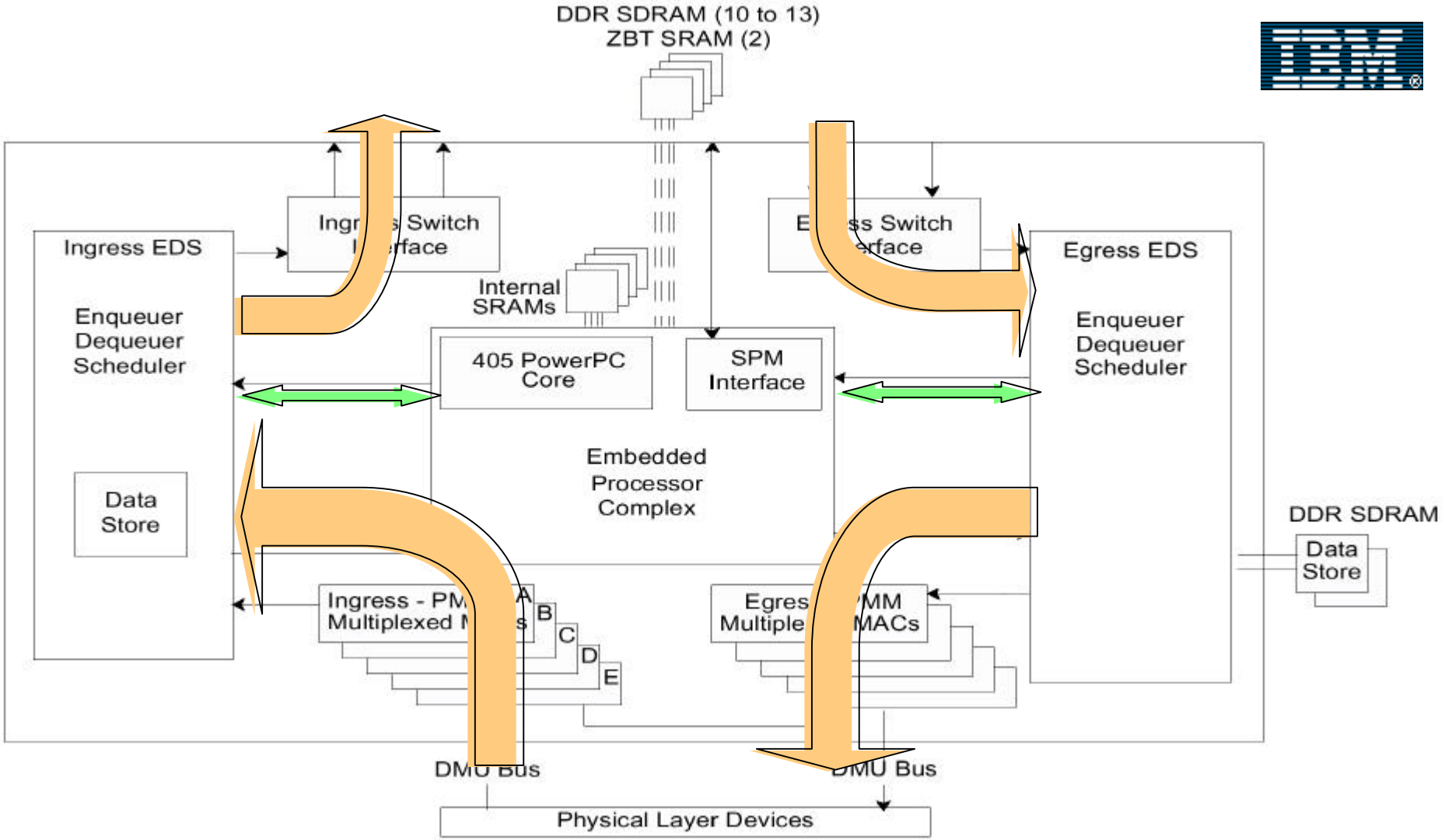
IBM NP4GS3 Embedded Processor Complex (EPC) Block-Diagram

RU Implementation using NPs



FEM Implementation using NPs



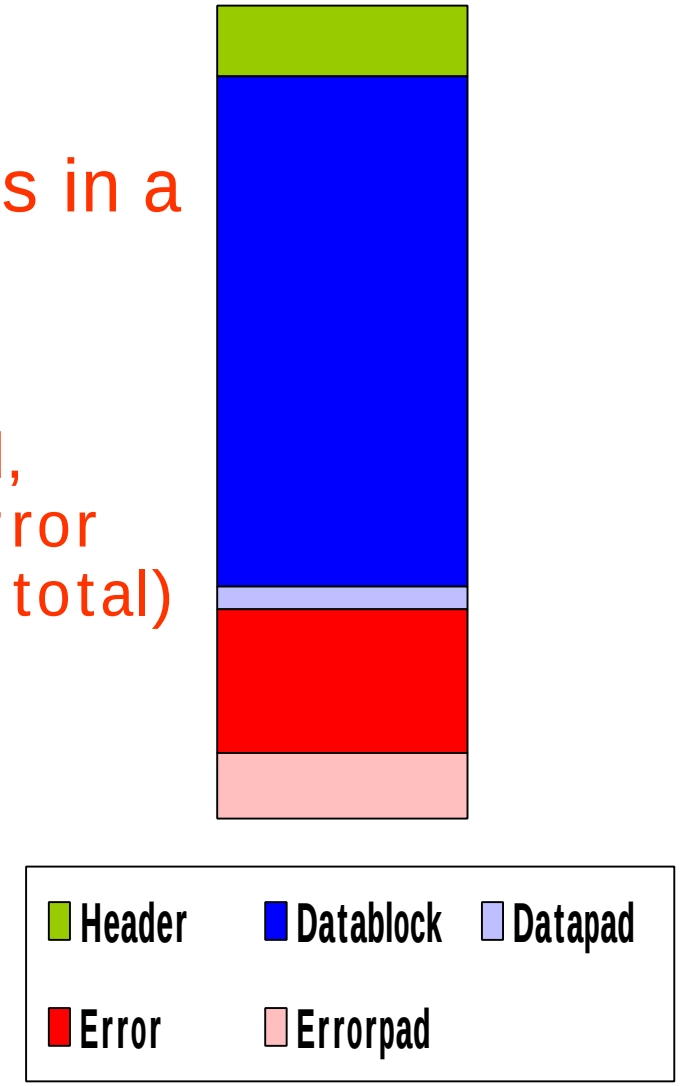

DDR SDRAM (10 to 13)
ZBT SRAM (2)
Ingress EDS
Enqueuer
Dequeuer
Scheduler
Data
Store
Internal
SRAMs
405 PowerPC
Core
SPM
Interface
Embedded
Processor
Complex
Egress EDS
Enqueuer
Dequeuer
Scheduler
DDR SDRAM
Data
Store
A
B
C
D
E
DMU Bus
DMU Bus
Physical Layer Devices

Software Environment

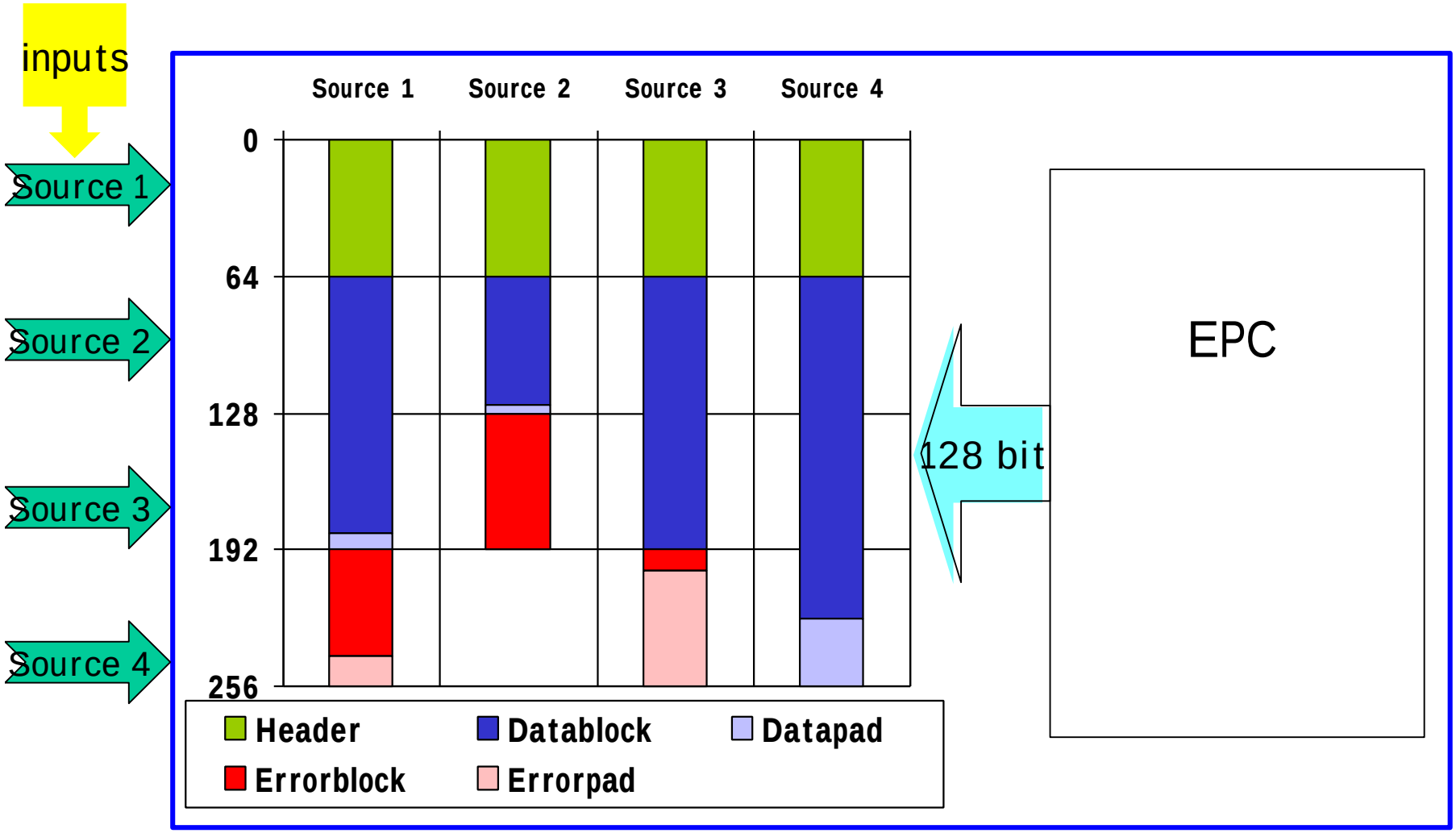
- IBM provides a rich software environment for the processor:
  - Assembler, Emulator and Debugger and Network traffic simulator
  - See Demo later…
- There exists also a hardware evaluation kit
  - The kit comprises NP, Gigabit Ethernet Ports, switch chip carrier, connector to Remote Debugger etc.
  - The Reference Evaluation Kit is a fully functional RU prototype (in principle)

Dataformat

- For technical reasons it is advantageous to have the blocks in a data fragment 64-byte aligned
- Data blocks consist of:
  - a header: event number, source id, offset of data block, offset of error block, and some others (64 Bytes total)
  - a data block (possibly empty)
  - a data padding to 64 bytes
  - an error block (possibly empty)
  - an error padding to 64 bytes



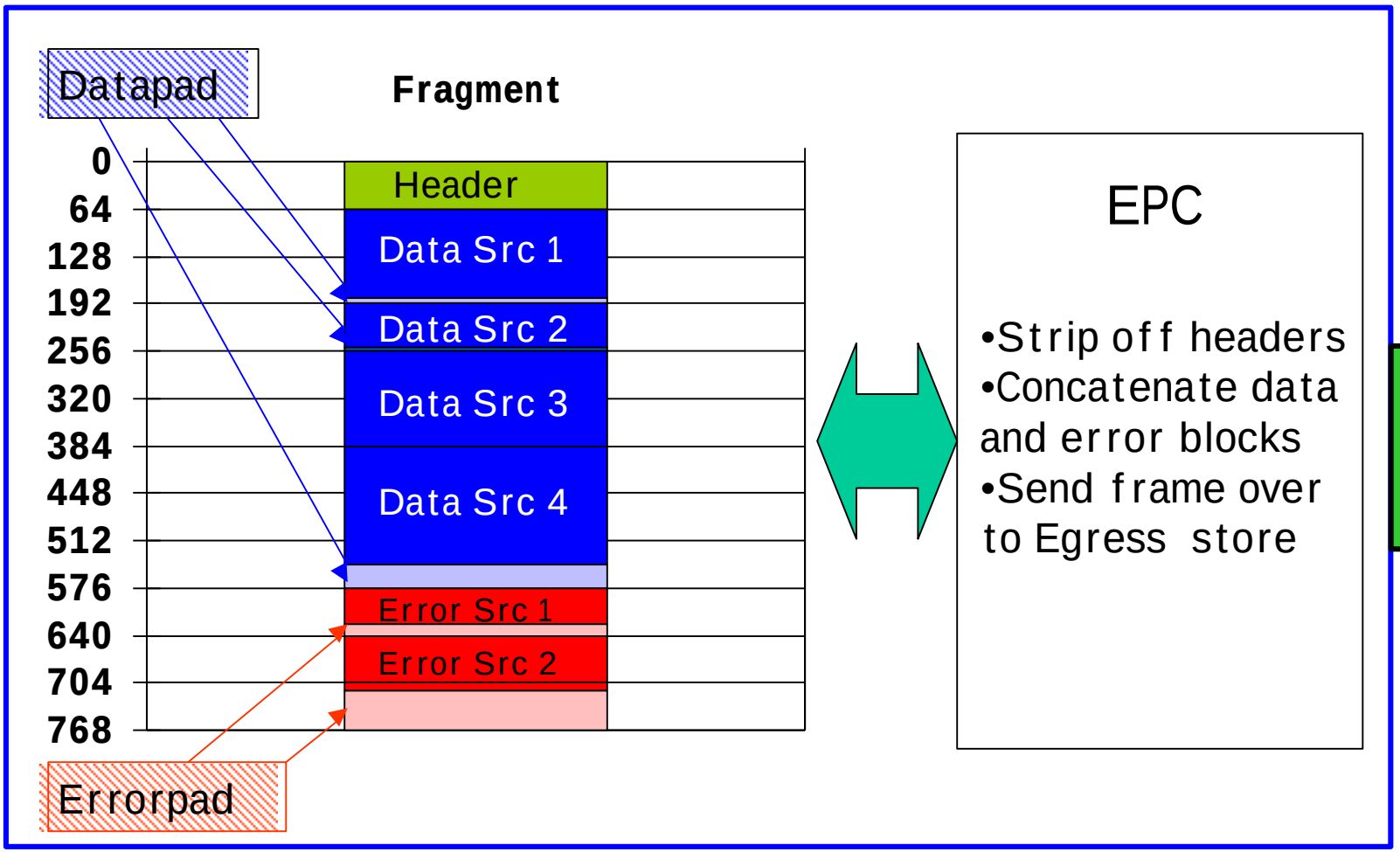
Data on the ingress side



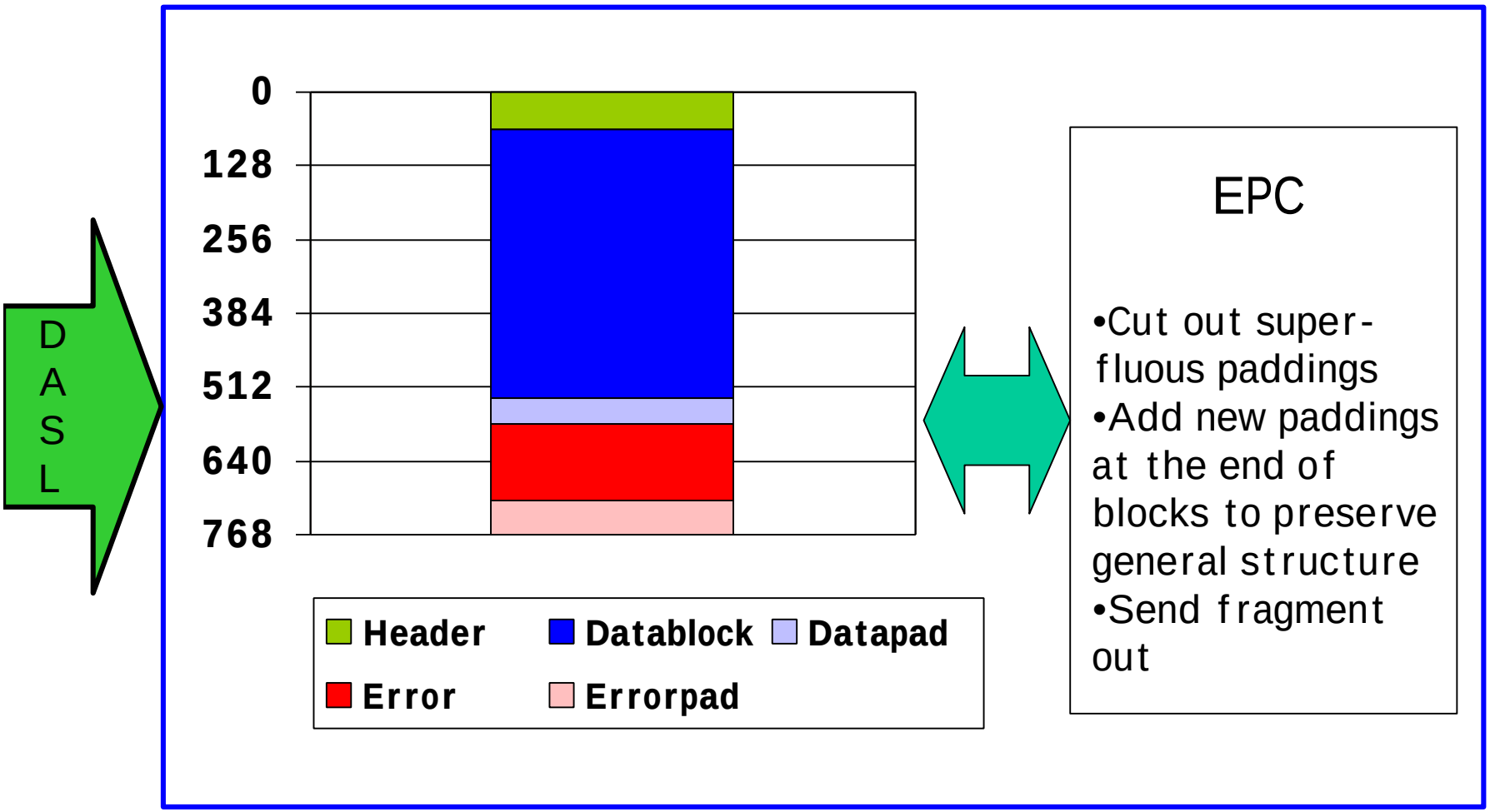
(Sub)-Eventbuilding

- The original headers of the individual fragments are stripped of, a new header of the (sub-)event is created
- Data blocks are concatenated in the order of their source
- Error blocks are concatenated in the order of their source and appended to the data block
- The paddings have to removed from the data and error blocks
- The concatenated block have again to be padded to 64 Bytes, to preserve the original structure

Concatenation in Ingress



Re-assembly on the Egress Side



Performance of Event Building Code

- The first running version of the code takes 4.5 μs / fragment (for a *single* thread out of ***30*** general purpose threads!)
- Single thread execution time can easily be cut by a factor 30%
- All main important features of the processor understood

Advantages of Software RU

- Simple board with very few components: NP4GS3s, 5xPhy, (PC-)Memory-Chips
- Ready-to-buy evaluation board to implement any desired prototype
- “Full” flexibility
  - e.g. destination assignment as function of key word in header
  - traffic shaping
  - etc.

Next Steps

- Evaluate different software versions for Multiplexer, RU
- Improve Running Speed of code
- Acquire a Reference Platform from IBM
- Study RU functionality (is a 'discrete' prototype)