StepNP™: A Driver for Multi-Processor SoC Tools

Pierre Paulin

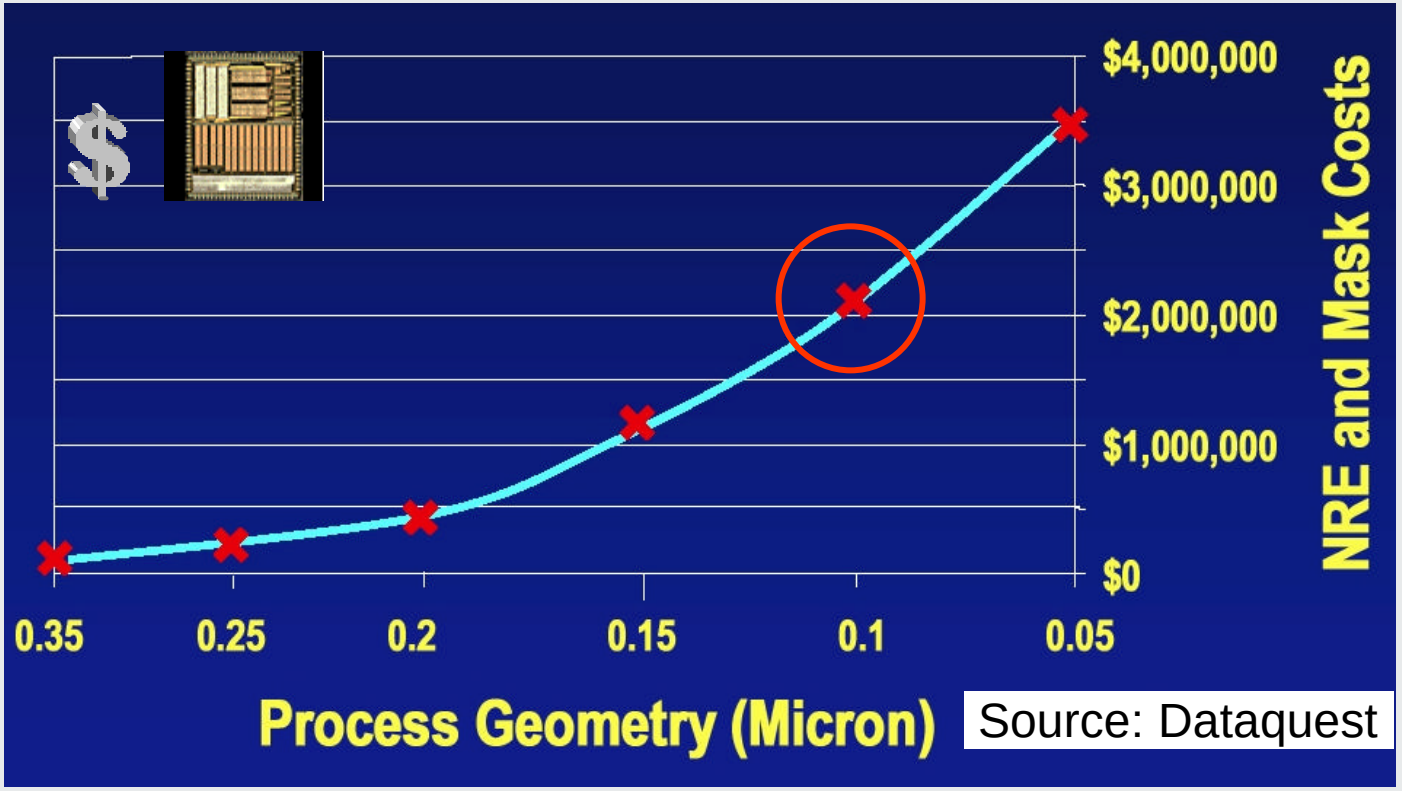
Director, SoC Platform Automation

Central R&D, STMicroelectronics

Outline

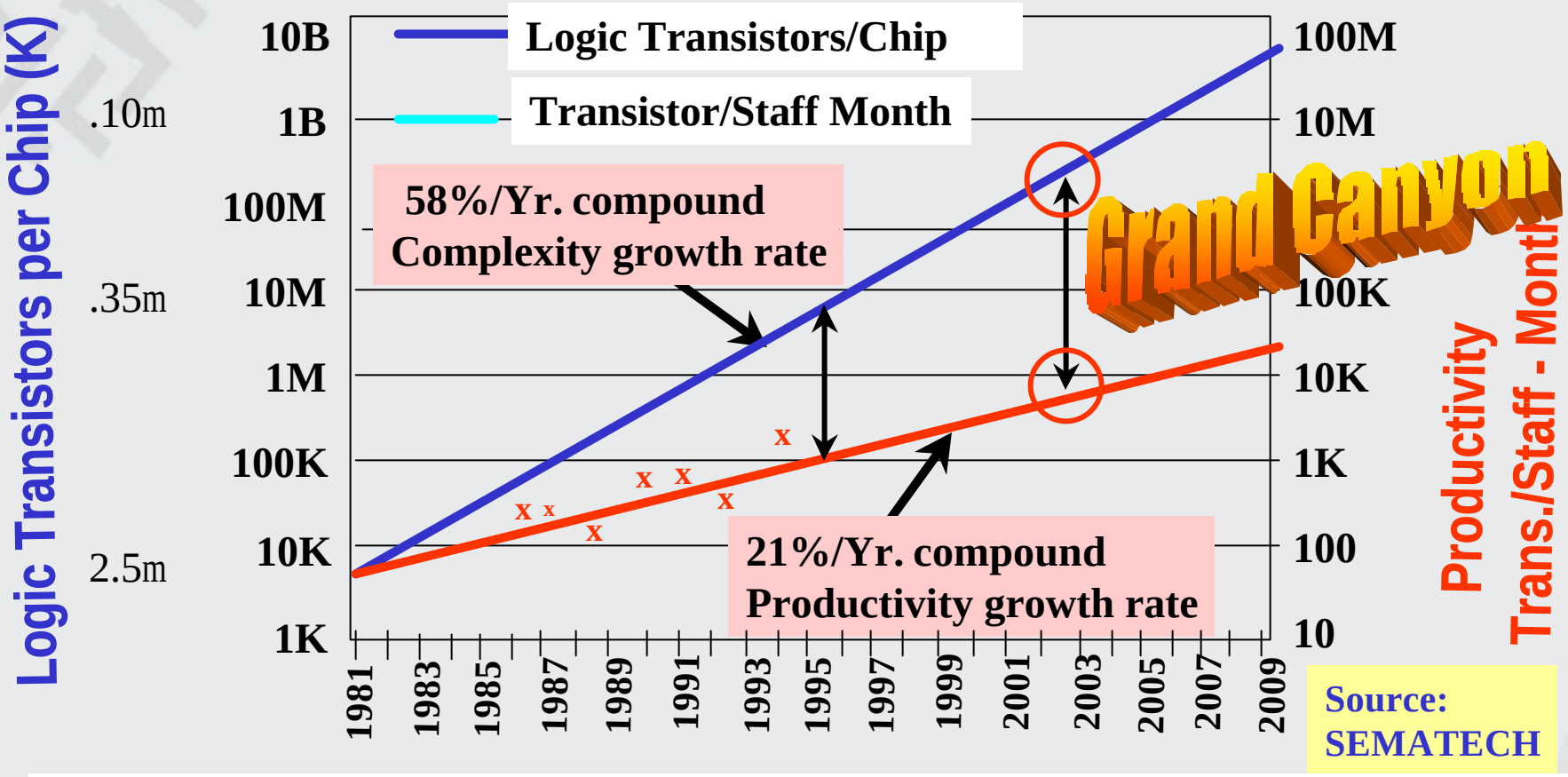
- Key SoC trends
- Heterogeneous multi-processor platforms
- ST MP-SoC R&D
  - StepNP™ architecture platform
  - MultiFlex MP-SoC tools & methods
  - IPv4 application results
- Research challenges

SoC Economic Trends: Mask NRE



- For $5 ASP with 20% profit margin:
  Need to sell over 2M parts to break even

The Productivity Gap



- 100M logic gates in 90nm = Logic of 1000 ARM7's
- Current 0.13u SoC's: 10M$ ~100M$ design cost

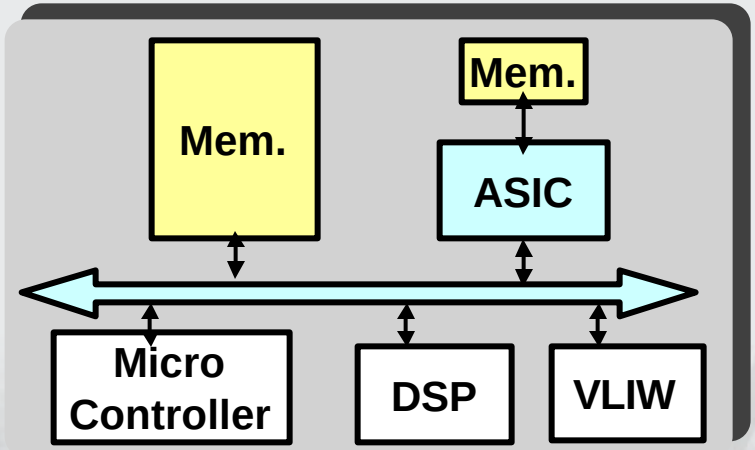
Key Trends

- ASIC/ASSP ratio: 80/20 in 2000, 50/50 now
  - In-house ASIC design is down
  - Replaced by off-the-shelf, programmable ASSP
- Number embedded processors in SoC rising:

Recordable DVD	5
Set-top box	7
Wireless base station	8
HDTV platform	8
Latest mobile handsets	10
Image processor	128
In-house NPU	>150

Current Practice

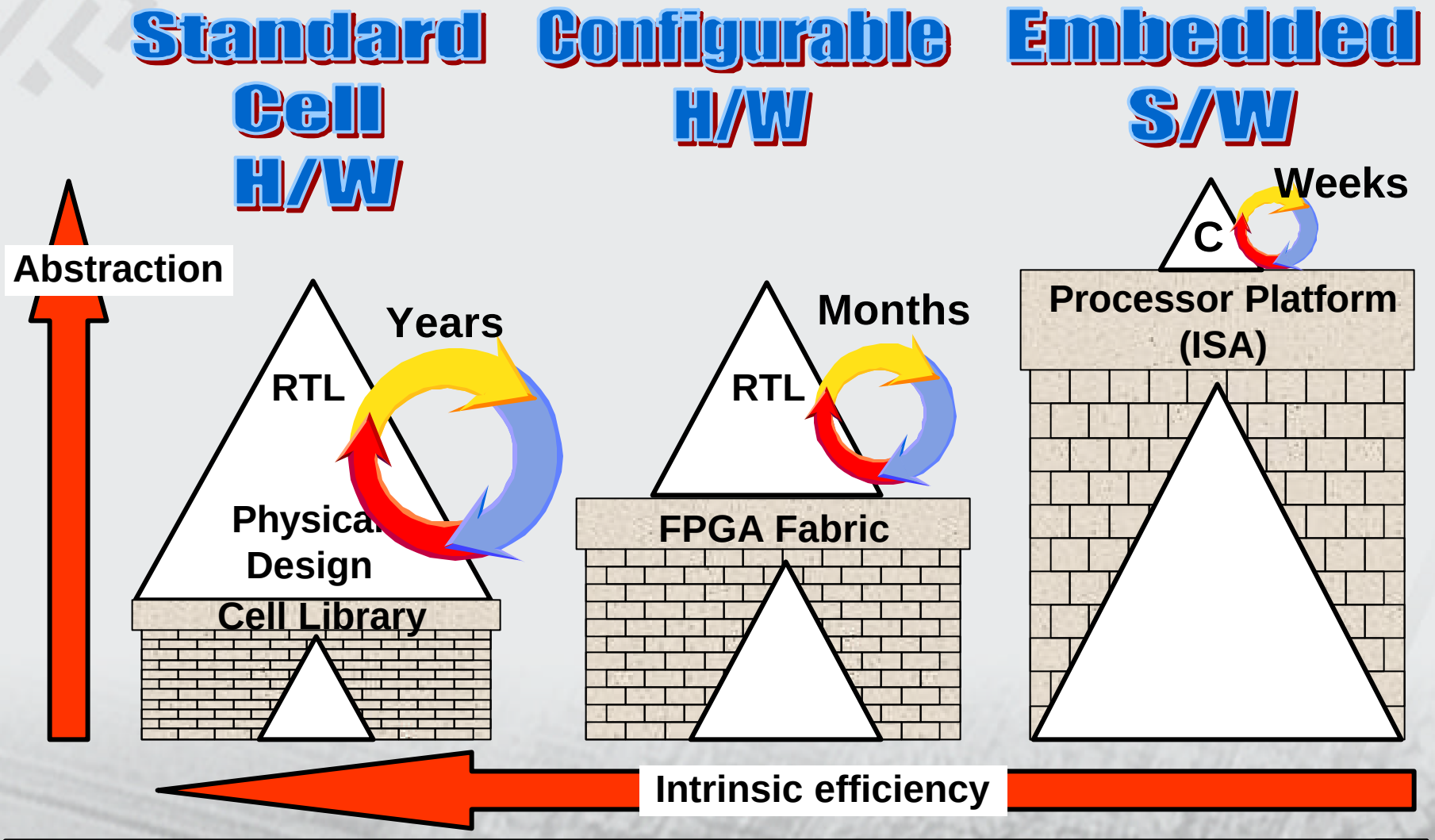
- Heterogeneous multi-processor SoCs are already current practice
- Problem is that each system is an ad-hoc solution: reaching complexity barrier
  - Little flexibility
  - No effective programming model
  - Lots of low-level programming
    - Poor SW productivity
    - Code not portable



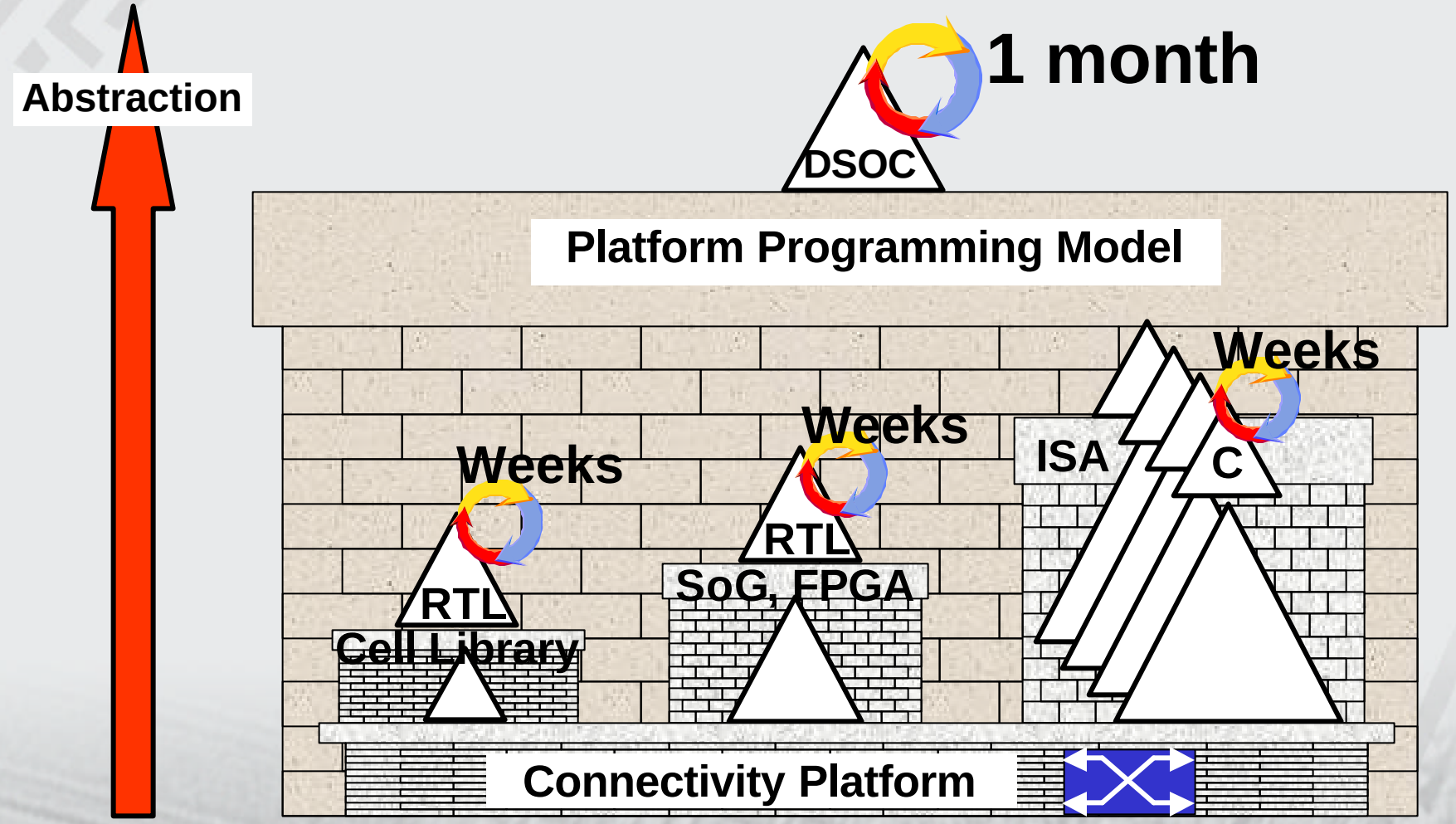
Next-generation SoC Platforms Key Objectives

- Flexibility: amortize NRE over more products
  - ‘Softer’ systems: eFPGA, eSoG, eProcessors, combined with H/W IP’s (standard, fast/low-power)
- Fast platform implementation
  - Use of synthesizable, off-the-shelf IP components
  - Scalable SoC interconnect
  - Trend towards standardized platforms
- Fast time-to-market for platform user
- Need clean programming model
  - Shield architecture complexity

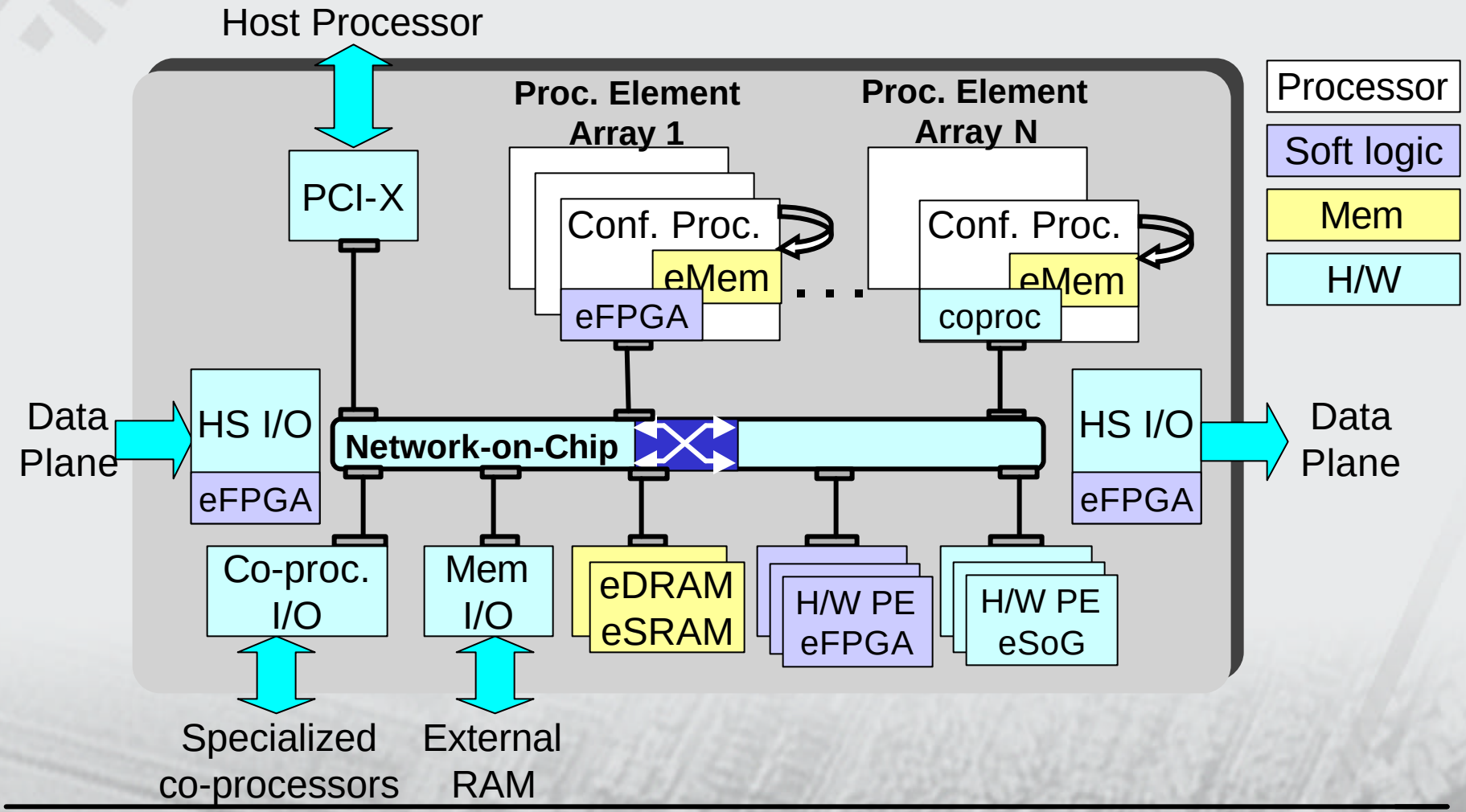
Configurability Trade-offs



ASAP: Advanced System Architecture Platform



Multi-processor Platform Example for Networking Applications

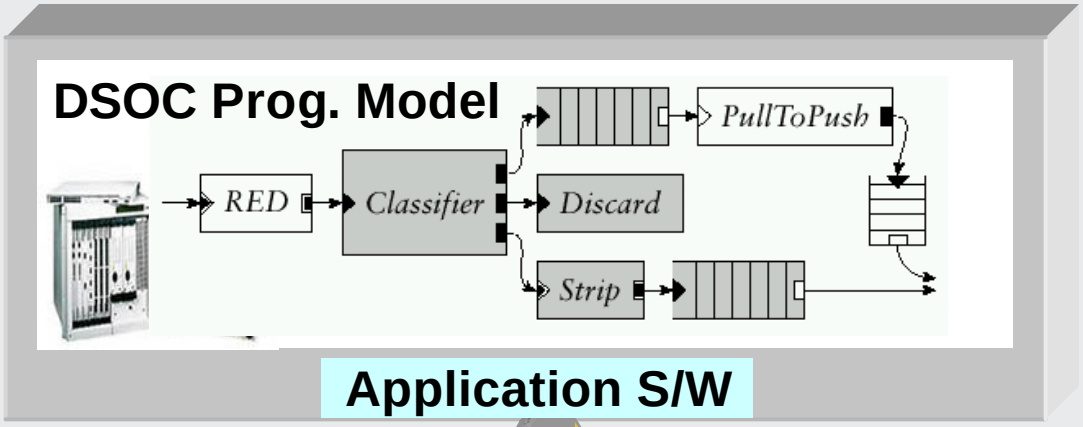


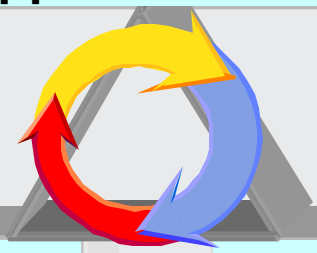
Outline

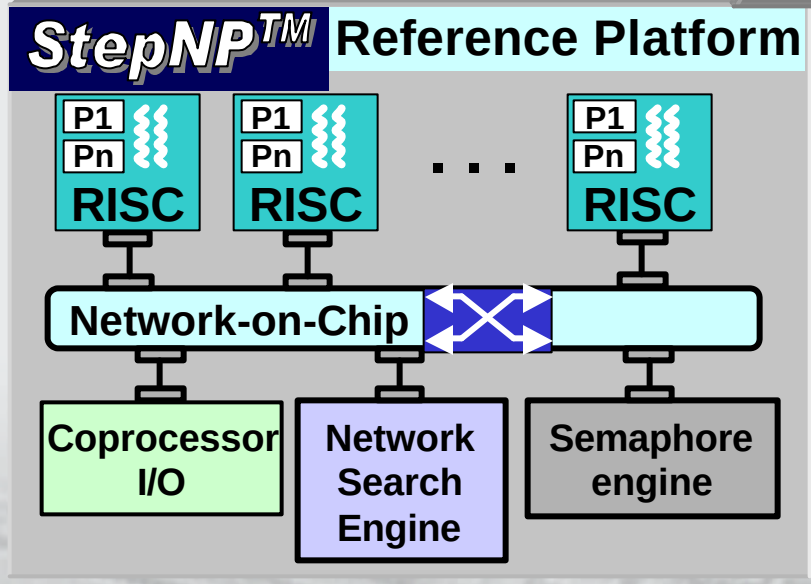
- Key SoC trends
- Heterogeneous multi-processor platforms
- ***ST MP-SoC R&D***
- Research challenges

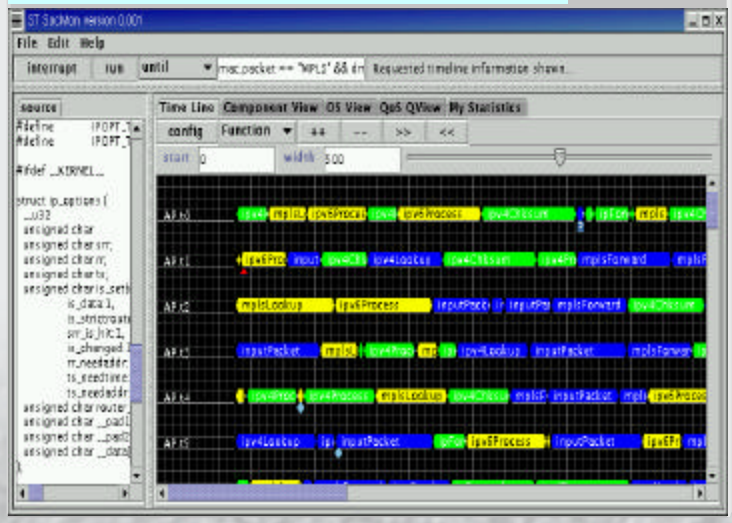
ST MP-SoC R&D

- *StepNP™* Architecture Platform
  - Multi-threaded processors
  - Interconnect
- MultiFlex Tools and Methodologies
  - MP-SoC analysis and debug tools
  - MP-SoC compilation
- Applications
  - IPv4 packet forwarding at 10Gb/s
  - Traffic management at 10Gb/s


DSOC Prog. Model
PullToPush
RED
Classifier
Discard
Strip
Application S/W

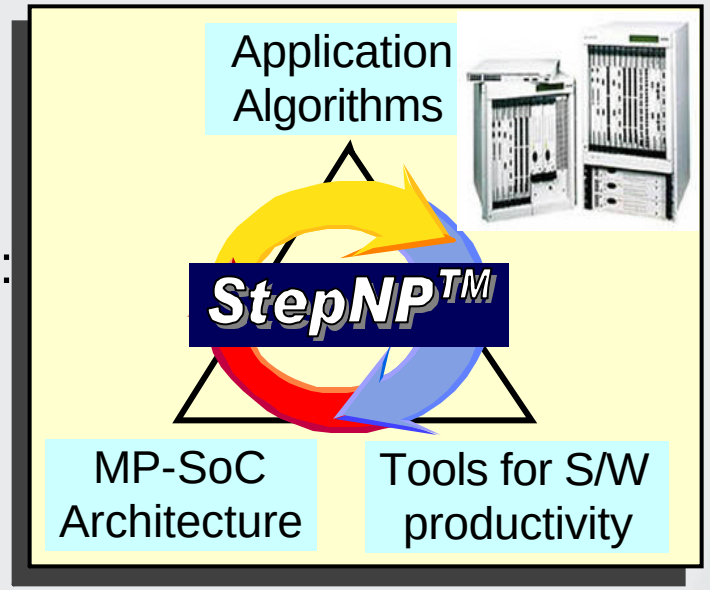



StepNP™
Reference Platform
P1
Pn
RISC
P1
Pn
RISC
. . .
P1
Pn
RISC
Network-on-Chip
Coprocessor I/O
Network Search Engine
Semaphore engine

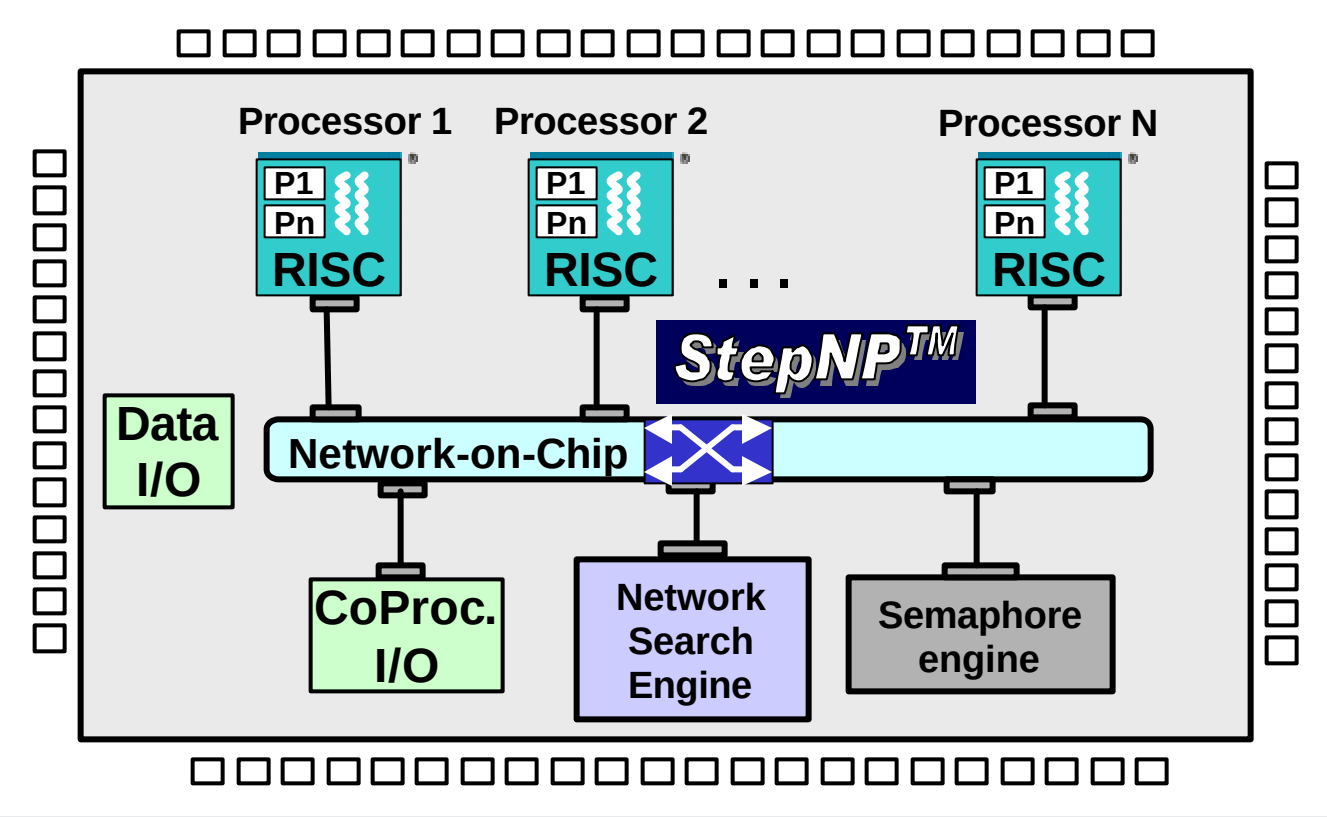

MultiFlex SoC Tools

StepNP™ Reference Platform

- System-level Telecom Exploratory Platform for Network Processing
  - Embodies key NPU characteristics: Multi-processors, H/W multi-thread Complex interconnect and memory hierarchies
- For ST SoC tool teams
  - Challenging internal driver
- For ST customers
  - Reference platform for networking IP (e.g. ST search engine)
  - Framework for customer packet processing SoC system design and eS/W tools development



StepNP™ Architecture



ARM POWERED ✓

DLX ✓

PowerPC® ✓

tensilica xtensa ✓

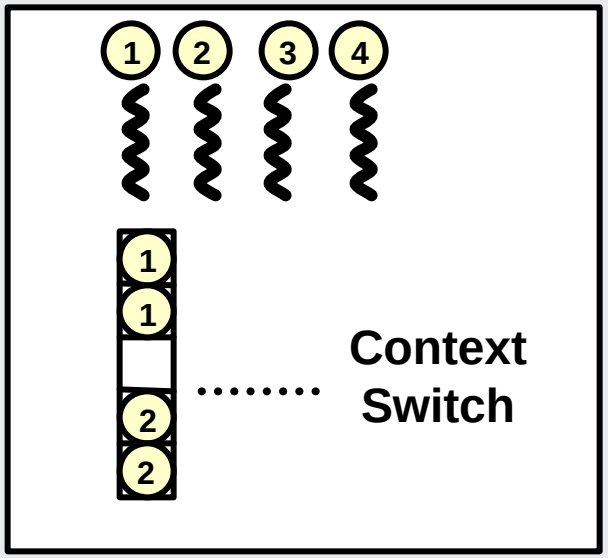
XILINX® MicroBlaze® ✓

MIPS TECHNOLOGIES INC

- Programmable multi-threaded network processor platform
- Popular processor models with configurable extensions: H/W multithreading and pipeline depth

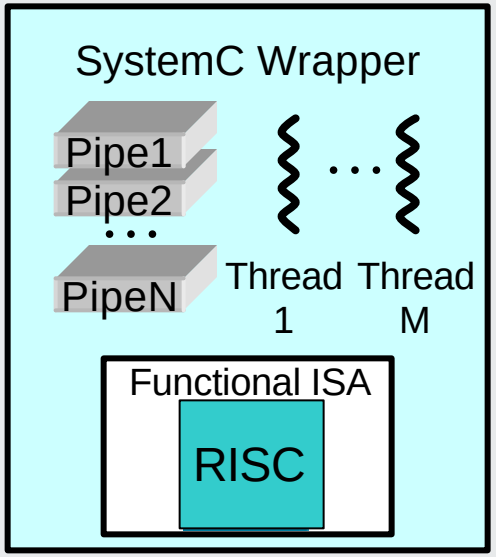
H/W Multi-threaded Processor

- Block interleaving technique
- Instructions of a thread executed successively until an event occurs that causes latency
  - Memory read/write
  - Table lookup
- Low overhead context switch

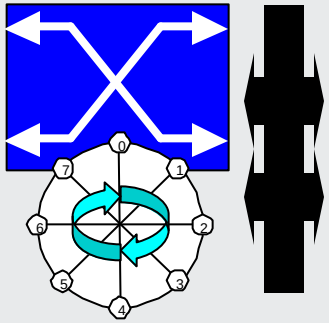


StepNP™ Reference Processors

- H/W multi-threaded processor
  - Hide latency via zero overhead thread context switch
  - Configurable number of hardware threads per processor
  - Configurable pipeline depth
- Each thread executes functional instruction set
- SystemC cycle-accurate model
  - Encapsulation of functional ISS's

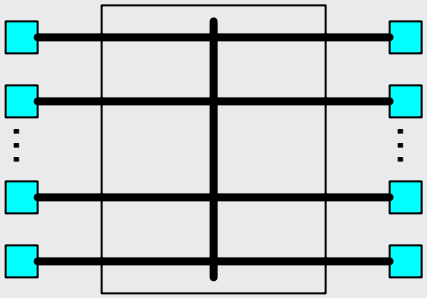


StepNP™ Interconnect Models



- General interconnect framework
  - NoC architecture exploration
- Multi-level modelling
  - Transaction-level to cycle-accurate models
- Various interconnect architectures
  - Bus
  - Ring
  - Tree
  - Crossbar

Network-On-Chip (NOC) Overview

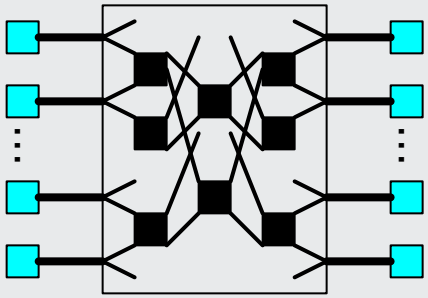


BUS-like

Low latency

Blocking (large contention)

Not easy to scale, need hierarchy



Tree-like

Medium latency

Blocking (blind routing)

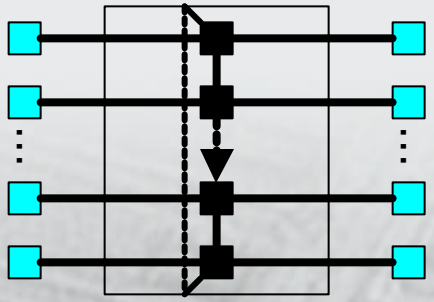
Medium scalability

Ring-like

Large latency

Can be non-blocking

Scalable

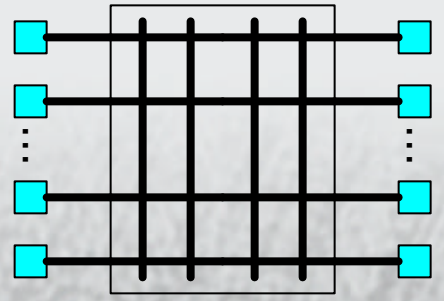


Crossbar-like

Low latency

Non-blocking

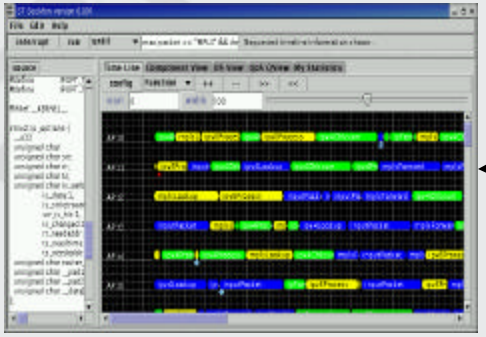
Costly, poor scalability



ST MP-SoC R&D

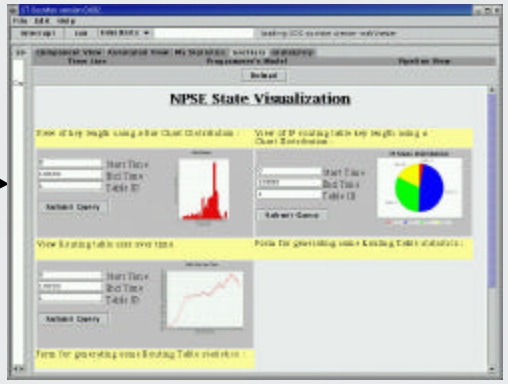
- *StepNP™* Architecture Platform
  - Multi-threaded processors
  - Interconnect
- ***MultiFlex Tools and Methodologies***
  - MP-SoC analysis and debug tools
  - MP-SoC compilation
- Applications
  - IPv4 packet forwarding at 10Gb/s
  - Traffic management at 10Gb/s

MultiFlex MP-SoC Tools



SoC Tools

Std. Views | Model Control | User-defd. Views

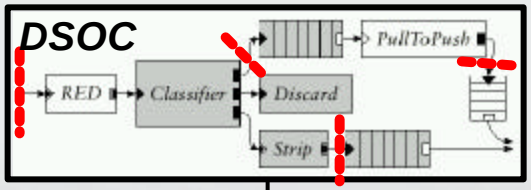


Packets In

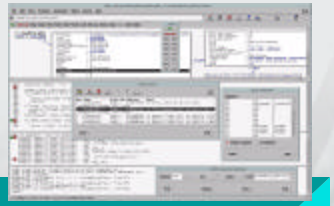
Packets Out

SIDL Access API

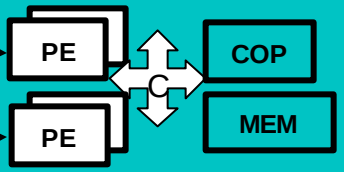
MP performance analysis



NPU Models:
Functional / Cycle-based

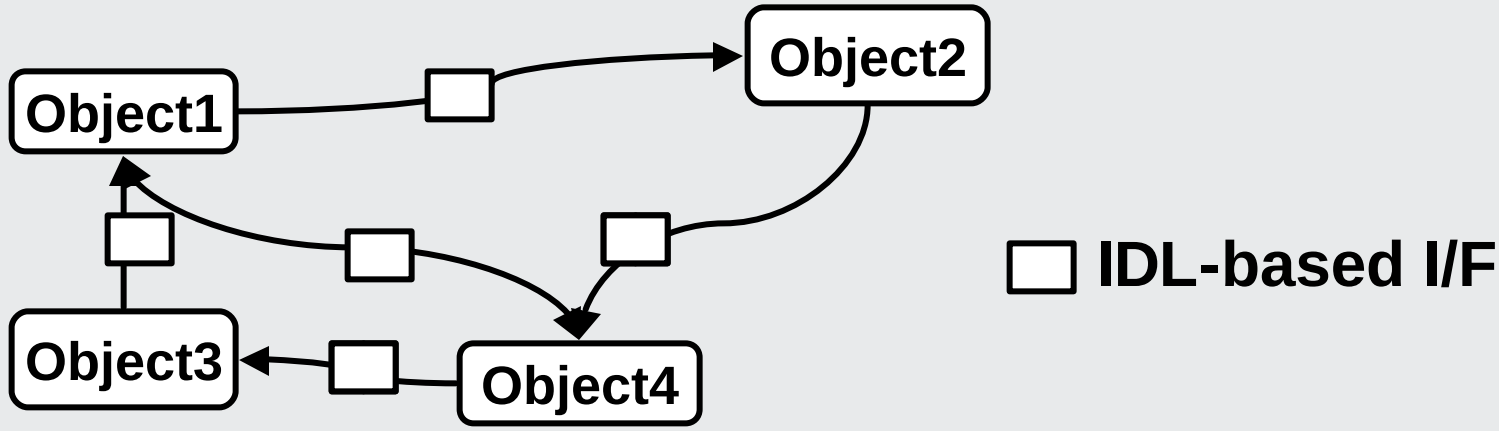


Multi-Processor Compilation



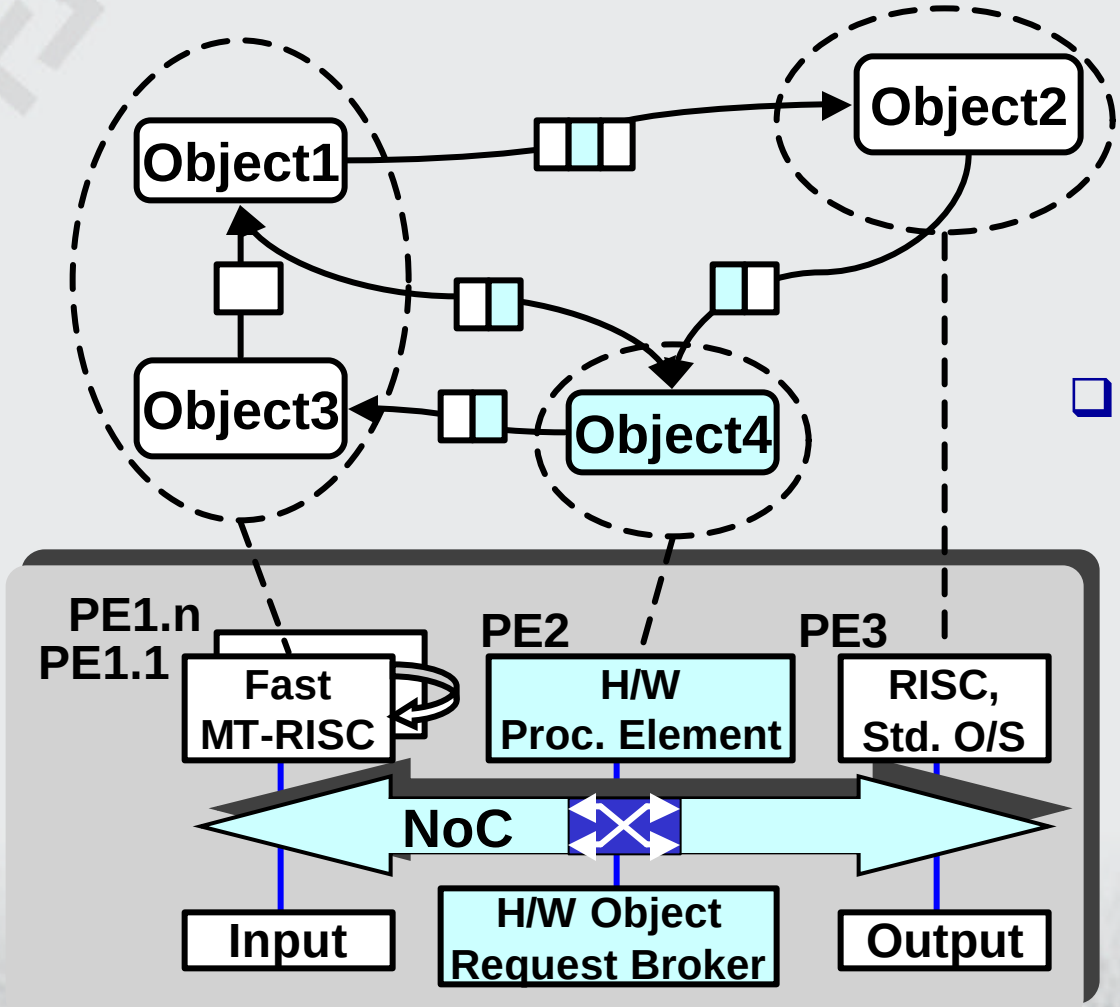
MP debugger

Parallel Programming Model
Distr. System Object Component



- Distributed Object Modeling: e.g. CORBA, DCOM
- Objects represent application functionality
  - Granularity of simple function to complete sub-system
- Inter-object communication via standard interface
  - Use of lightweight IDL (Interface Description Language)
- No assumptions on subsequent assignment of processing engines (H/W or S/W), interconnect, O/S

DSOC to Platform Mapping

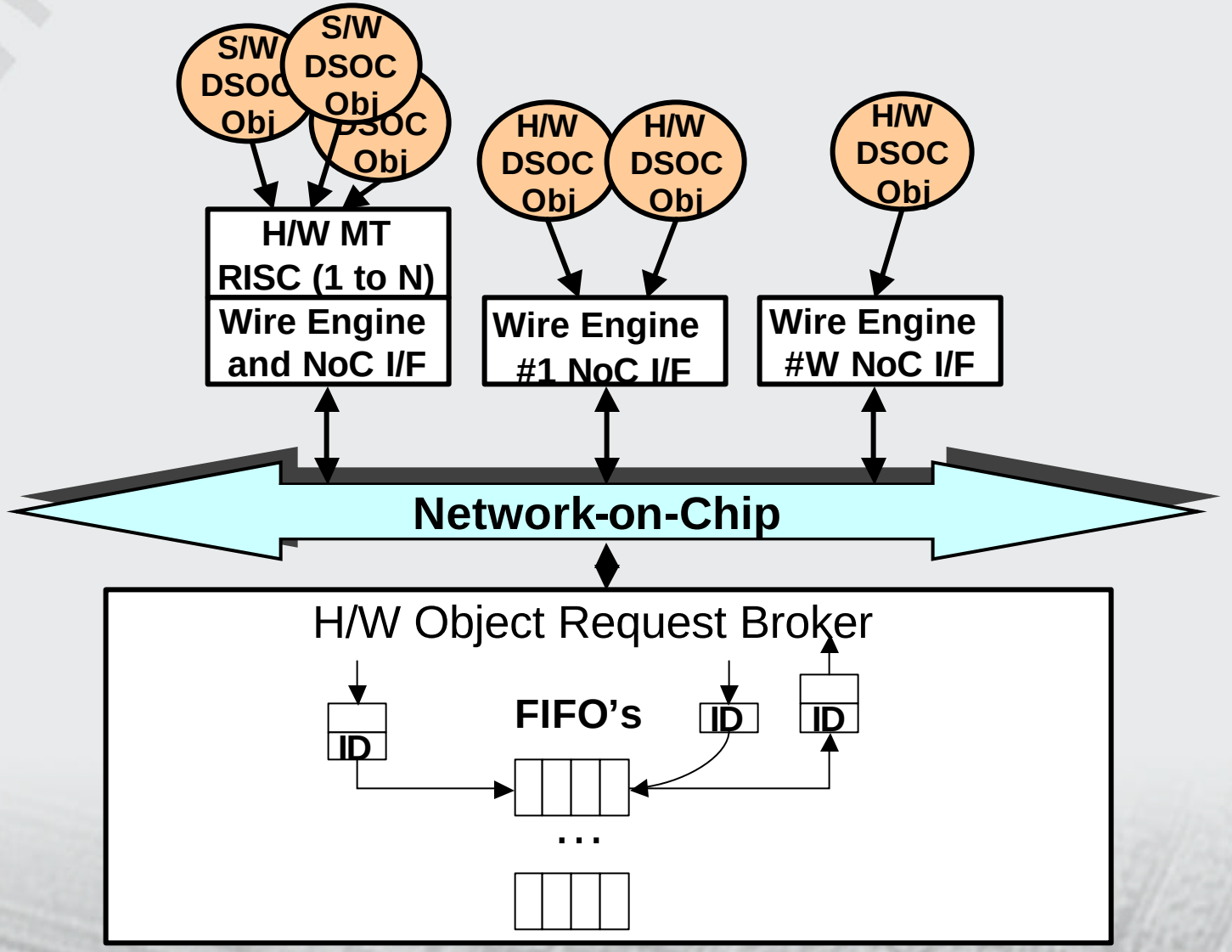


S/W-S/W com

S/W-H/W com

- Synthesis from IDL: Auto generation of
  - Drivers between different O/S
  - Drivers between S/W PE's and network-on-chip
  - Glue logic between H/W PE's and network-on-chip

DSOC to Platform Mapping



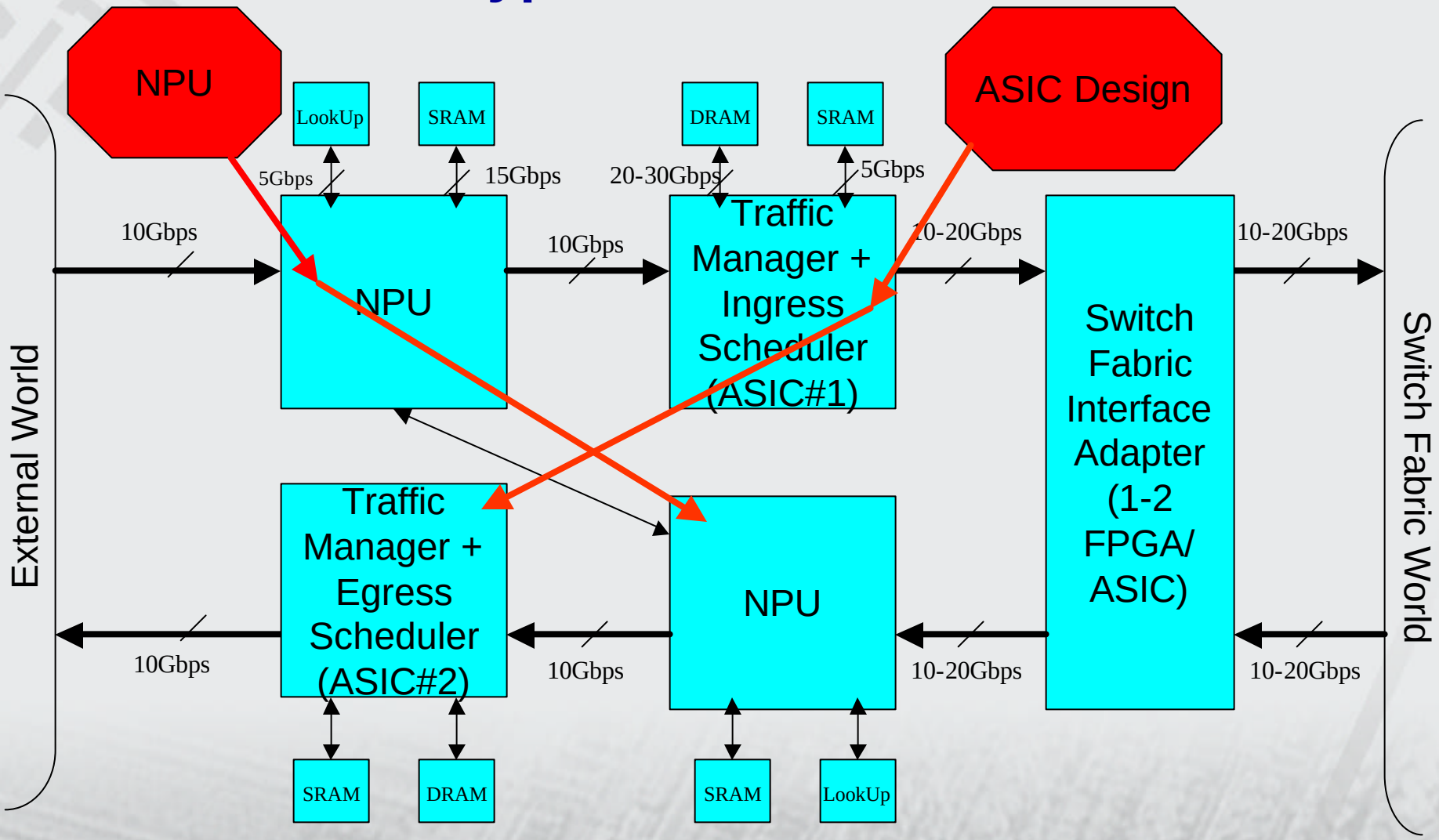
Implementation Details

- Goal: very high performance
  - To support 10 Gbits traffic management
- Current Implementation
  - Object communication all in C++
  - O(10) RISC instructions per remote object invocation, plus 2 or 3 per argument.
  - Sustained ARM to ARM object method invocations at 10 MHz.

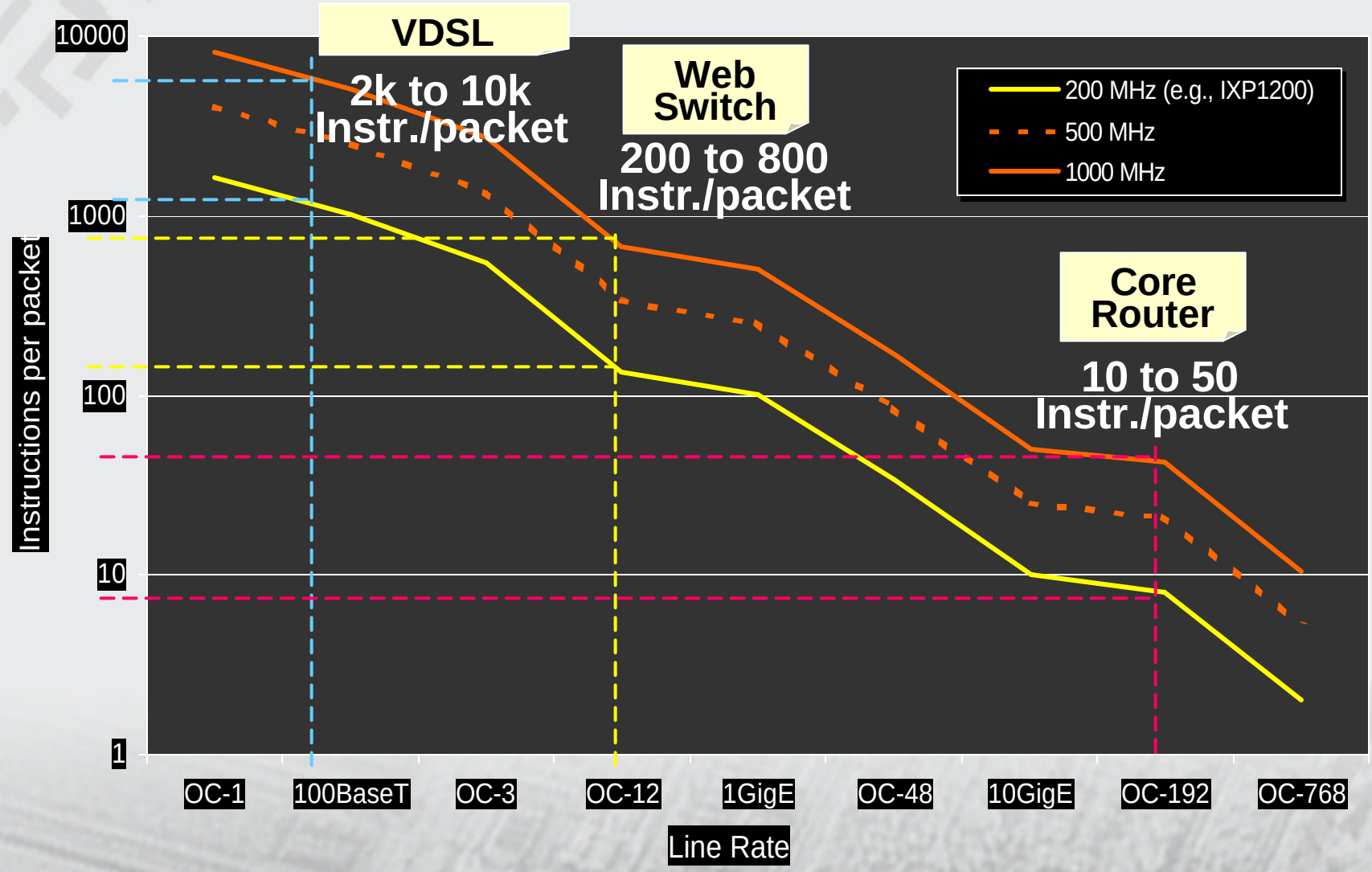
ST MP-SoC R&D

- *StepNP™* Architecture Platform
- MultiFlex Tools and Methodologies
- ***Applications***
  - IPv4 packet forwarding at 10Gb/s
  - Traffic management at 10Gb/s

Typical Architecture



NPU Instructions / Packet



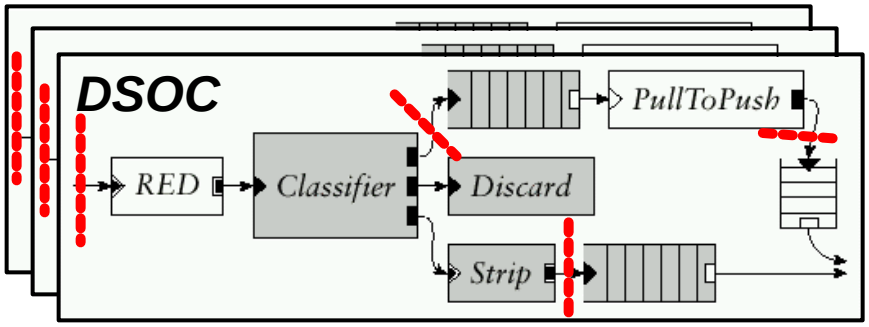
Applications

- IPv4 packet forwarding application
  - DSOC model
    - Click/C++ internal object function
- Traffic management
  - DSOC model
    - C++ for internal object function
  - Mappable to SystemC or RISC processor

10Gb/s IPv4 Demonstrator

- Automatic mapping of high-level IPv4 model onto parameterizable NPU platform model
  - Mapping to processor, threads, H/W coprocessors
  - Configurable model parameters
    - Number of processors, threads per processor
    - Interconnect and memory latencies
- Fast simulation (TLM + ISS): 7KHz (32 PE's)
- Control, visualization and analysis tools
  - connects to model automatically
  - extracts relevant information via SIDL API
  - Interactive waveform viewing, analysis, annotation
  - Performance analysis

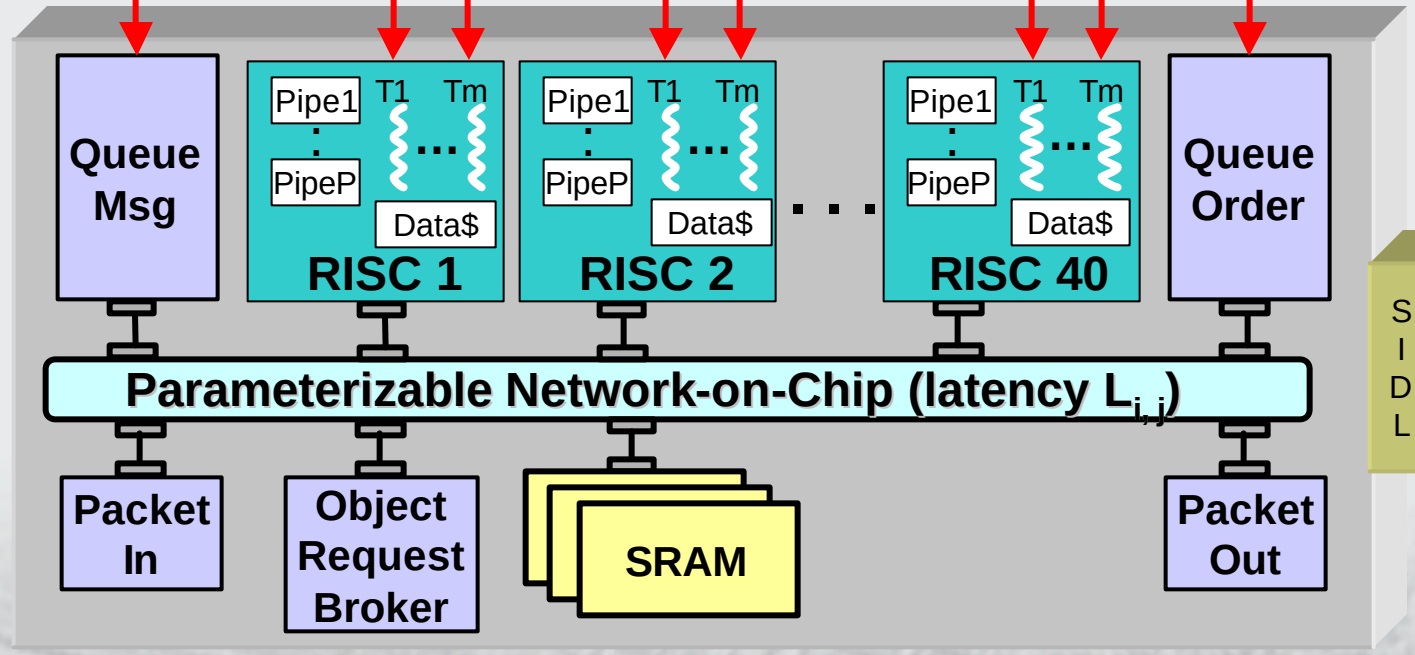
10 Gb/s IPv4 packet forwarding



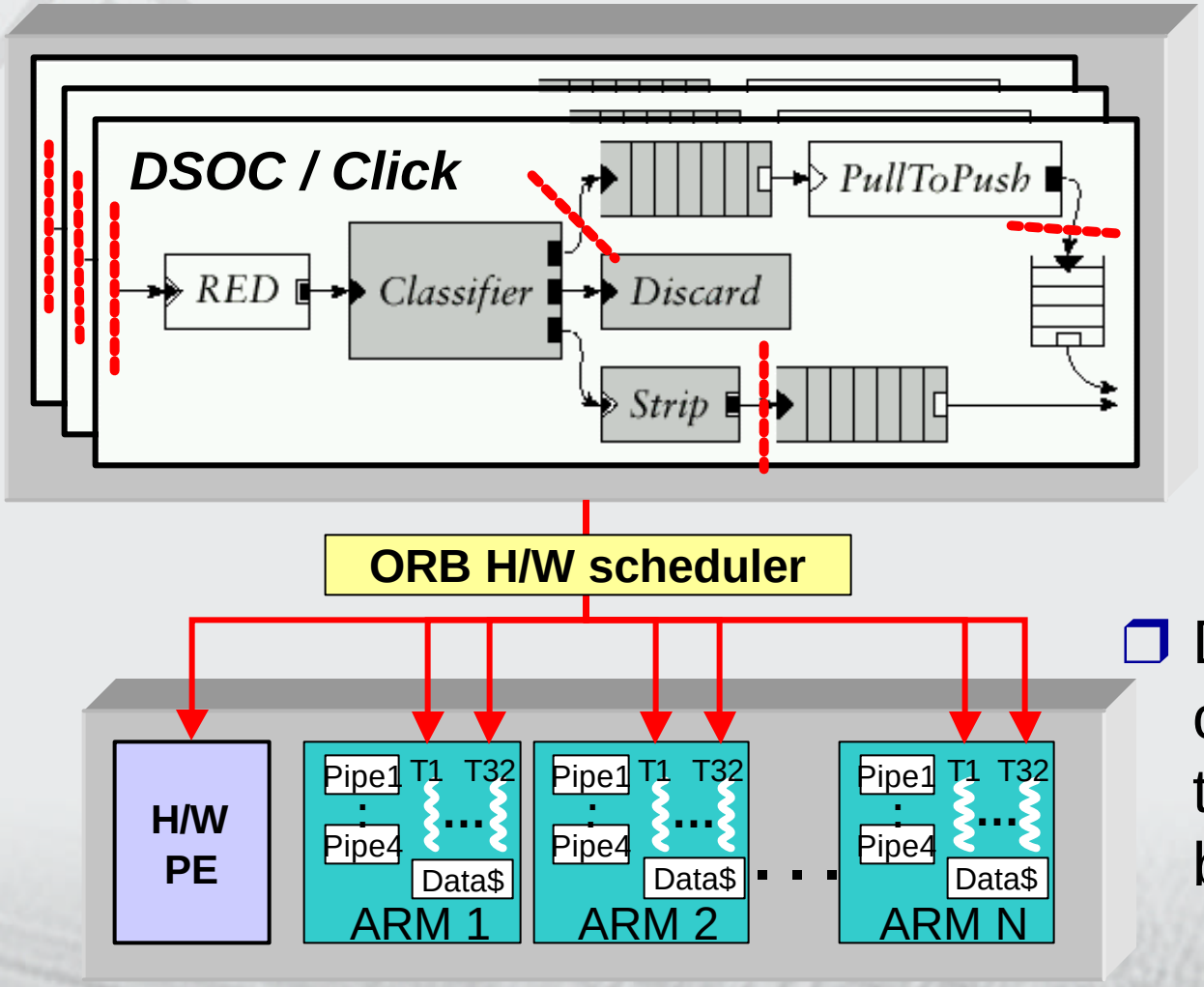
processors N = 1-40
clock = 500 MHz
pipe stages = 4
threads Tm = 1-32
D$ sets = 256
D$ size = 4 KB
Latency $L_{i,j}$ = 0-320 ns
SRAM banks = 4
SRAM acc = 10 ns

Automatic mapping on MP/MT architecture

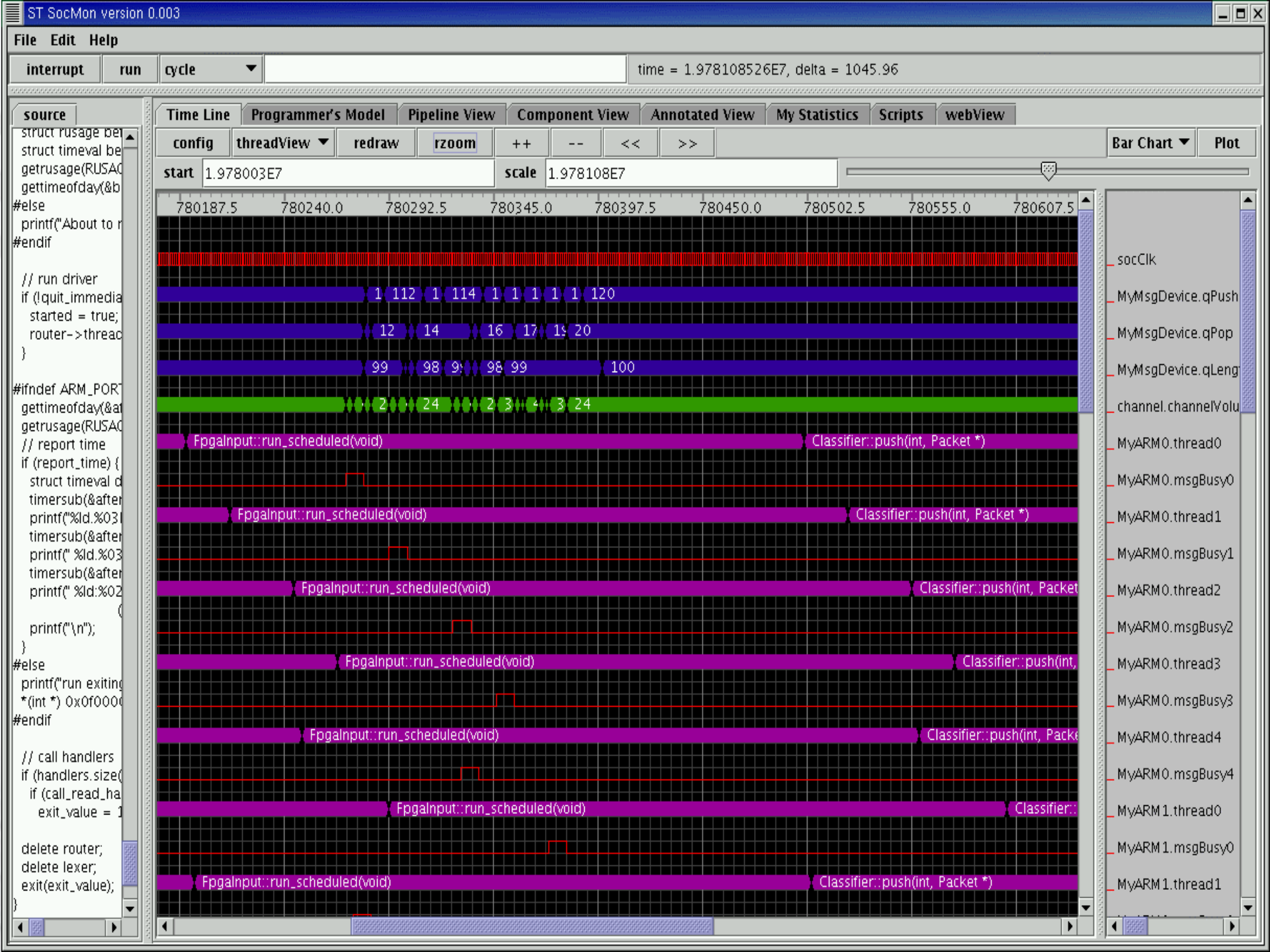
Processor
H/W
Mem



Graph Partitioning and Allocation

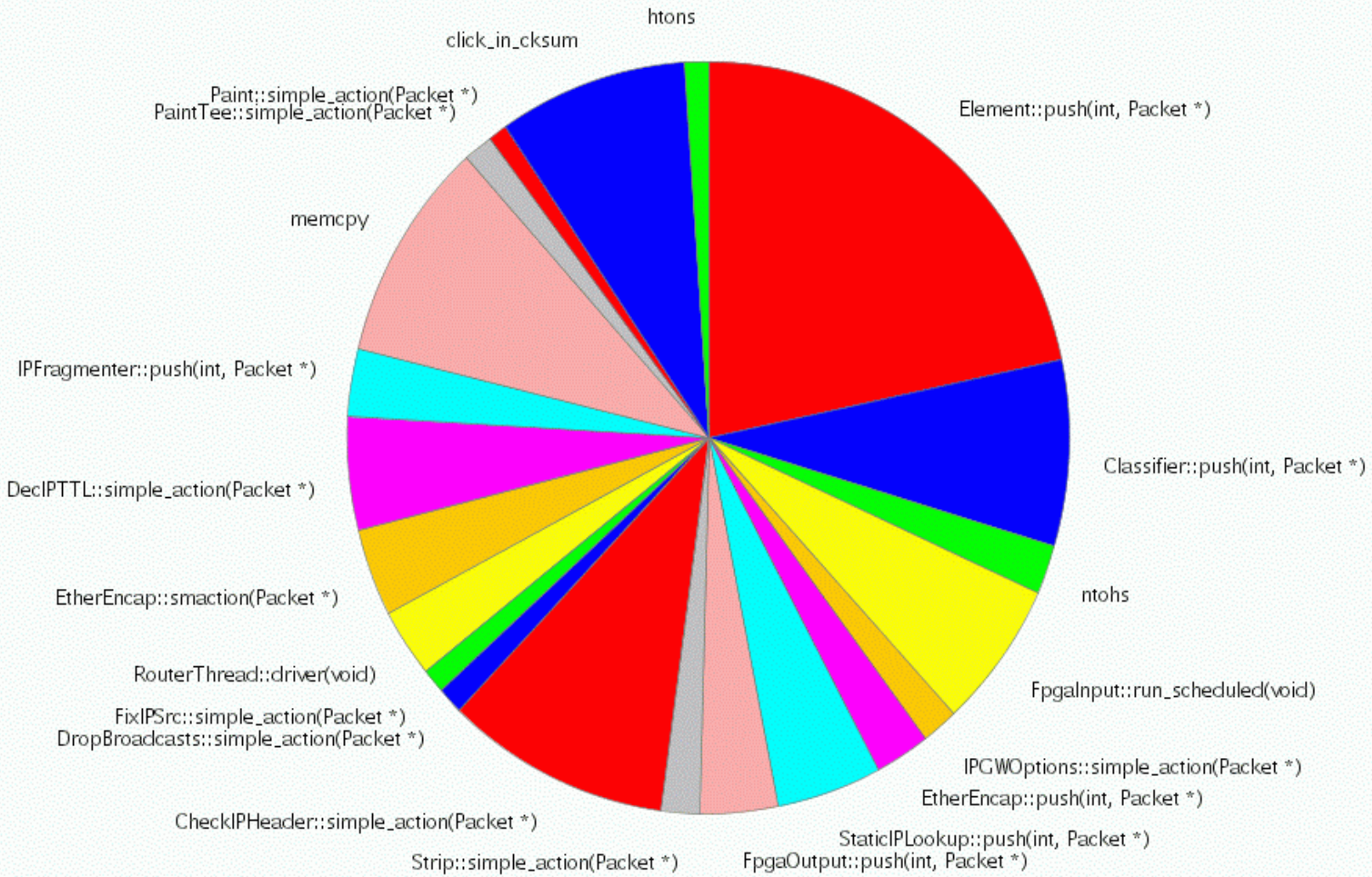


- Encapsulation of Click into DSOC model
- Static graph partitioning at compile time
- Dynamic allocation of graph segments to H/W threads by H/W scheduler

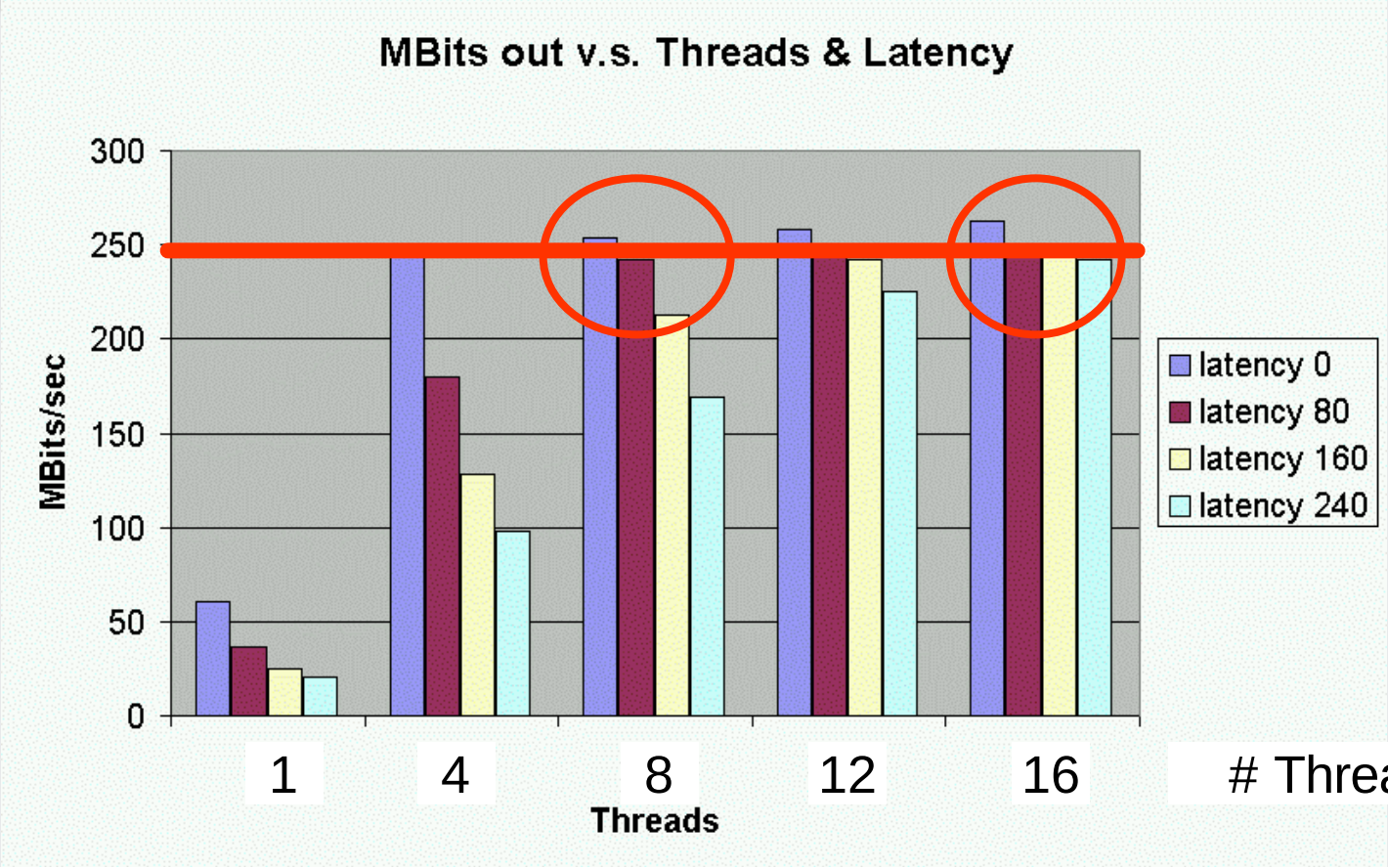

ST SocMon version 0.003
File Edit Help
interrupt
run
cycle
time = 1.978108526E7, delta = 1045.96
source
Time Line
Programmer's Model
Pipeline View
Component View
Annotated View
My Statistics
Scripts
webView
config
threadView
redraw
rzoom
++
--
<<
>>
Bar Chart
Plot
start 1.978003E7
scale 1.978108E7
780187.5
780240.0
780292.5
780345.0
780397.5
780450.0
780502.5
780555.0
780607.5
socClk
MyMsgDevice.qPush
MyMsgDevice.qPop
MyMsgDevice.qLeng
channel.channelVolu
MyARM0.thread0
MyARM0.msgBusy0
MyARM0.thread1
MyARM0.msgBusy1
MyARM0.thread2
MyARM0.msgBusy2
MyARM0.thread3
MyARM0.msgBusy3
MyARM0.thread4
MyARM0.msgBusy4
MyARM1.thread0
MyARM1.msgBusy0
MyARM1.thread1
FpgaInput::run_scheduled(void)
Classifier::push(int, Packet *)

Signal Plot

Signal Value Distribution

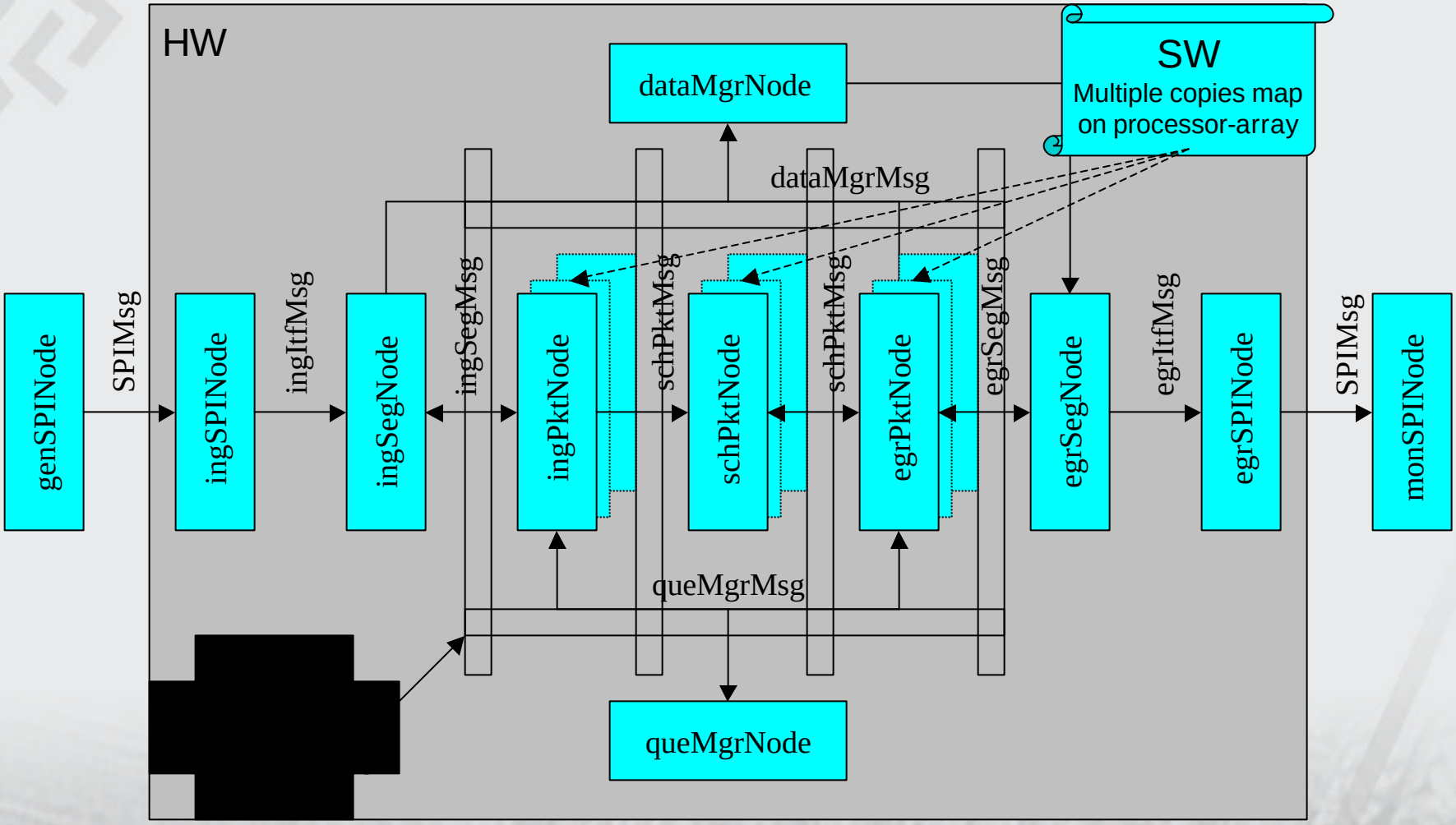


IPv4 Simulation Results



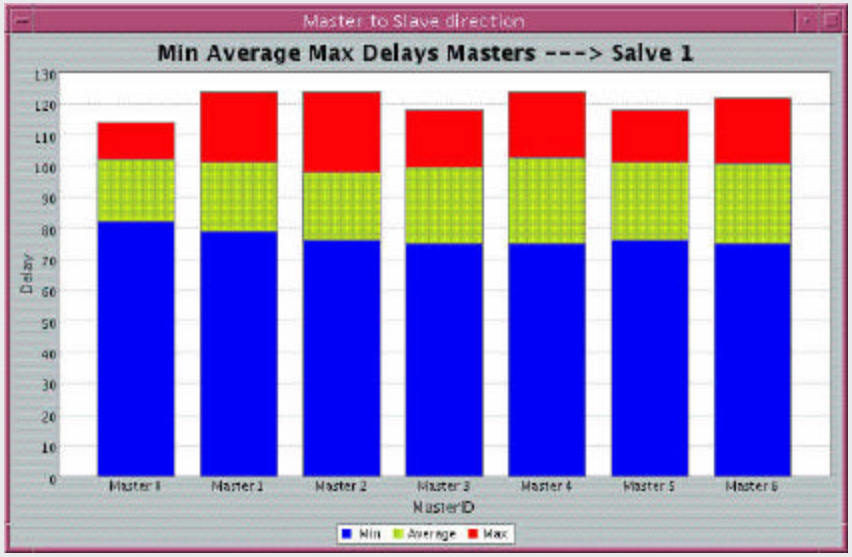
- Normalized results for 1 ARM
- 250 Mb/s (80%): 8 threads absorbs 80ns NoC latency
  16 threads absorbs 240ns NoC latency

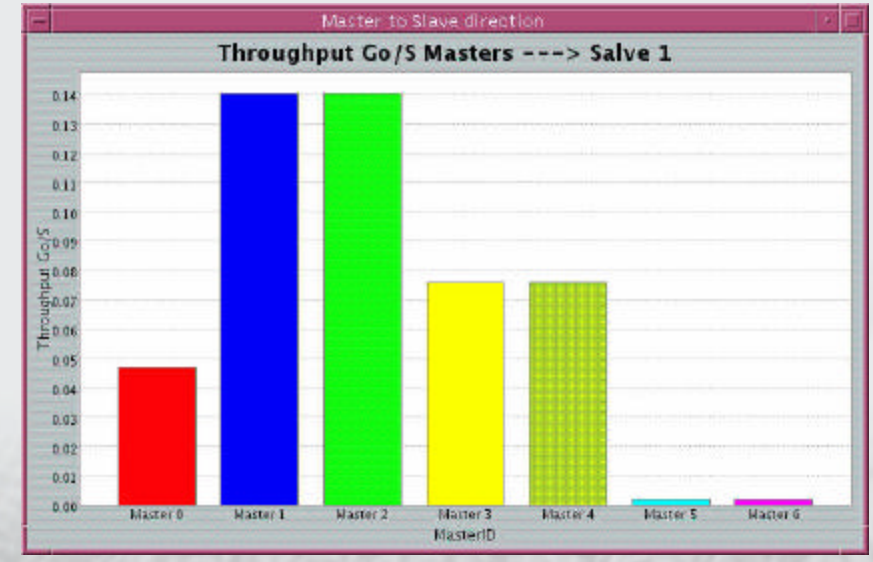
Traffic-Manager: Target Architecture



NoC Analysis Visualization

- Traffic Mgr. mapped on 7 ARMs with NoC XBar 100+/-25ns Latency





Outline

- Key SoC trends
- Heterogeneous multi-processor platforms
- ST MP-SoC R&D
- ***Research challenges***

Research Challenges

- Refinement of programming models
  - Mixed DSOC and SMP models
  - Evaluate for multimedia, consumer, wireless applications
- Heterogeneous system partitioning
  - RISC + Config. Processor + H/W and/or eFPGA
- DSOC H/W compilation
  - Synthesis of H/W interface logic
- StepNP implementation prototype
- NoC architecture evaluations
  - DSOC partitioning optimized for various NoC

New: SMP Support

- Addition of shared-memory multi-processing option
  - Subset of objects can be declared SMP
    - More appropriate MP model for Multimedia
  - O/S (in C++ with H/W assist):
  - Java/C# style concurrency primitives implemented
    - Priority scheduler
    - Monitors, conditions, semaphores
- Communication with DSOC objects via message passing

Summary

- Increased flexibility needs: ‘soft’ SoC platforms
  - *Need clean programming model for mp-s/w+c/w+h/w*
- DSOC MP programming model
  - Distributed concurrent objects: HL message passing
  - H/W-based object request broker, msg passing
    - *Very low overhead (10 MHz MP msg rate)*
  - High-end (10Gb/s) applications demonstrated
  - 80% processor utilization achieved
- Natural extension: more Corba features (S/W)
  - Fault tolerance, load balancing
  - Natural to add new scheduling priorities
- Shared Memory (SMP) model where needed