Trends and Requirements for Network Processor SoC tools MP SoC School, July 2002

Pierre Paulin, Director

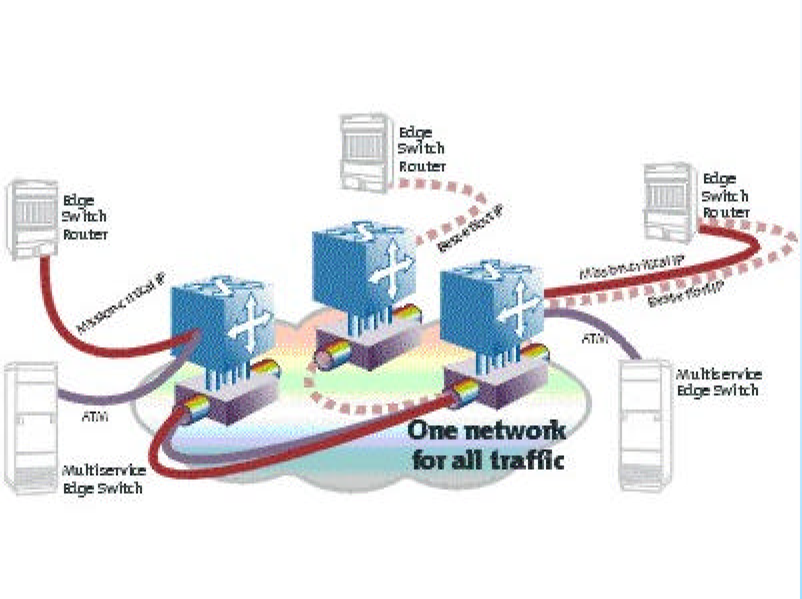
SoC Platform Automation Technologies
STMicroelectronics, Ottawa, Canada

Outline

- Network Processors
  - Why?
  - High performance: How?
  - NPU tools survey
- *StepNP*^{TM} Research NPU Platform
  - Parallel processor architecture
  - Router applications
  - SoC tools and methods
- R&D Needs Outlook
  - Which platform, which SoC tools?

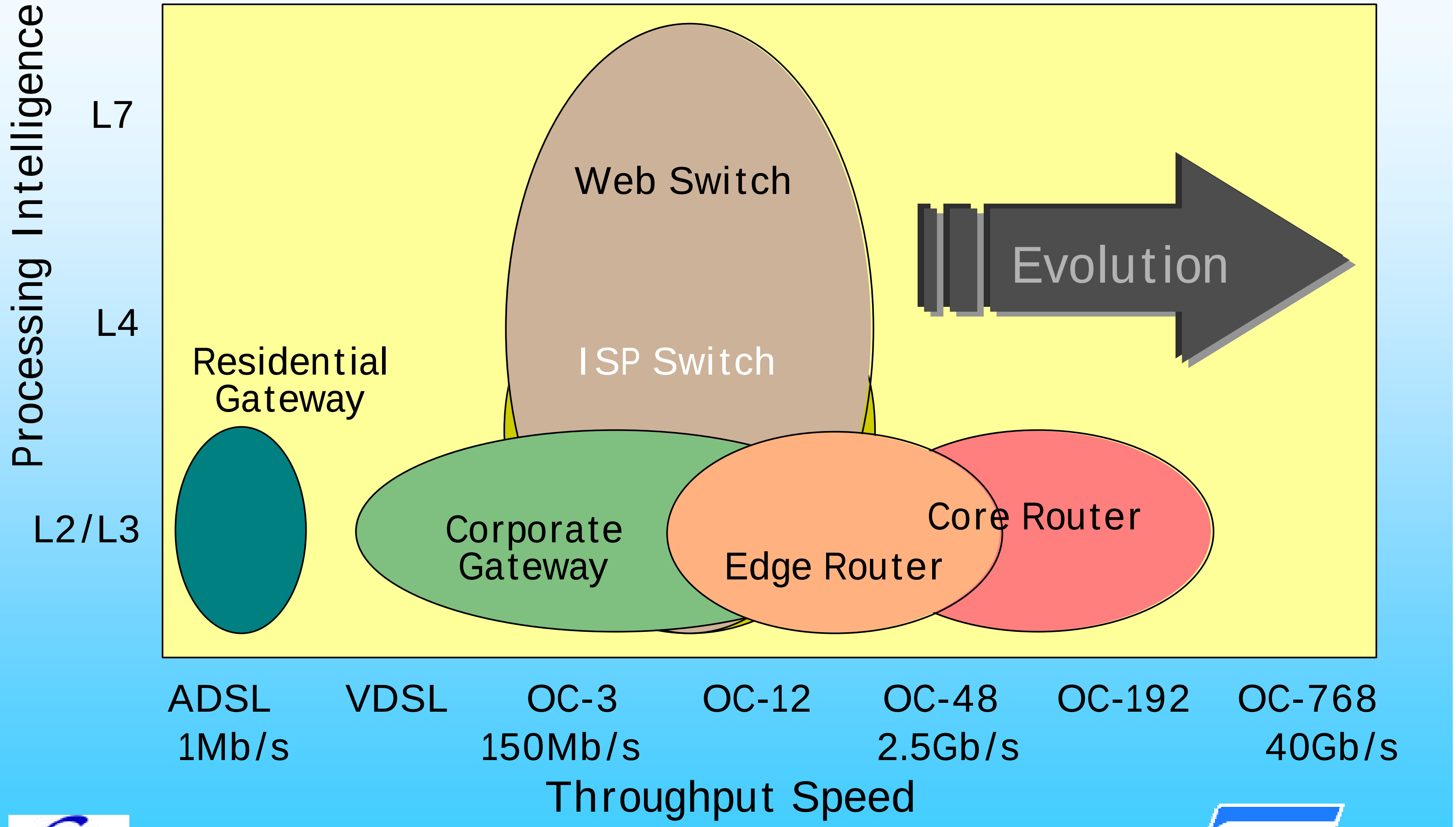
Why are NPU’s Interesting?

- ❒ Key component of emerging intelligent high-bandwidth optical network
- ❒ Commercial perspectives
- ❒ First real industrial parallel processing application

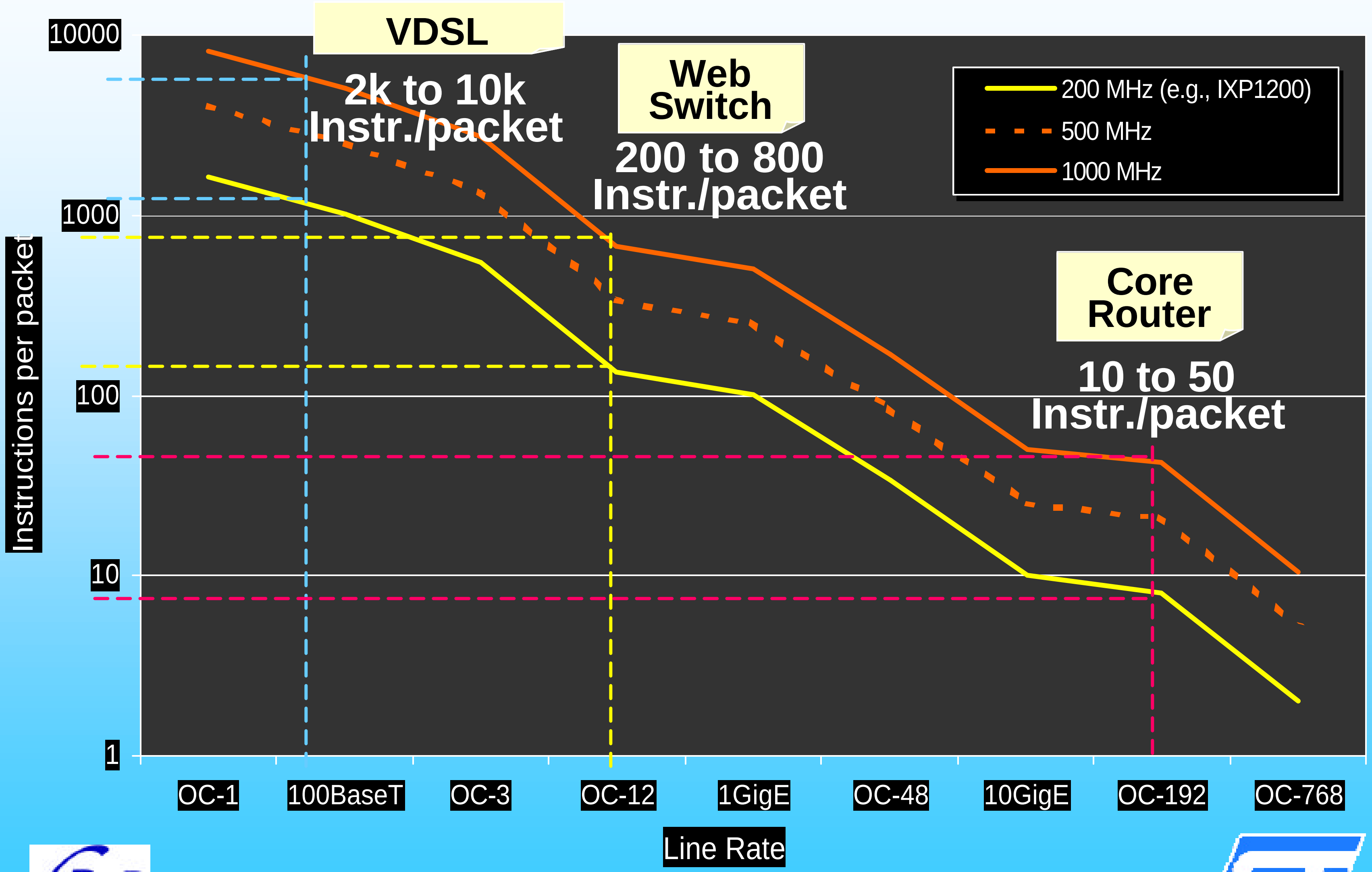


- ❒ Cost-effective & useable parallelism
- ❒ Perfect driver for next generation SoC tools
  - ➢ Embedded software & System-level design tools
  - ➢ Multi-processor tools

Router/Switch Space (Today)



NPU Instructions / Packet



VDSL

2k to 10k Instr./packet

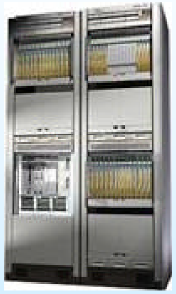
Web Switch

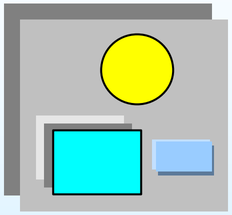
200 to 800 Instr./packet

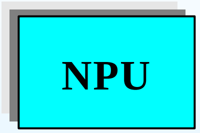
Core Router

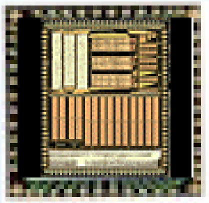
10 to 50 Instr./packet

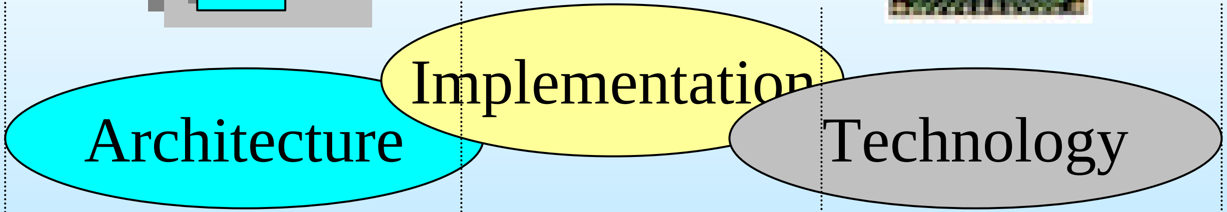
Enabling High Performance











Architecture

Implementation

Technology

Performance = $$\frac{\text{Application}}{\#\,\text{Instructions}} \times \frac{\#\,\text{Instructions}}{\text{Cycles}} \times \frac{\text{Cycles}}{\text{Second}}$$

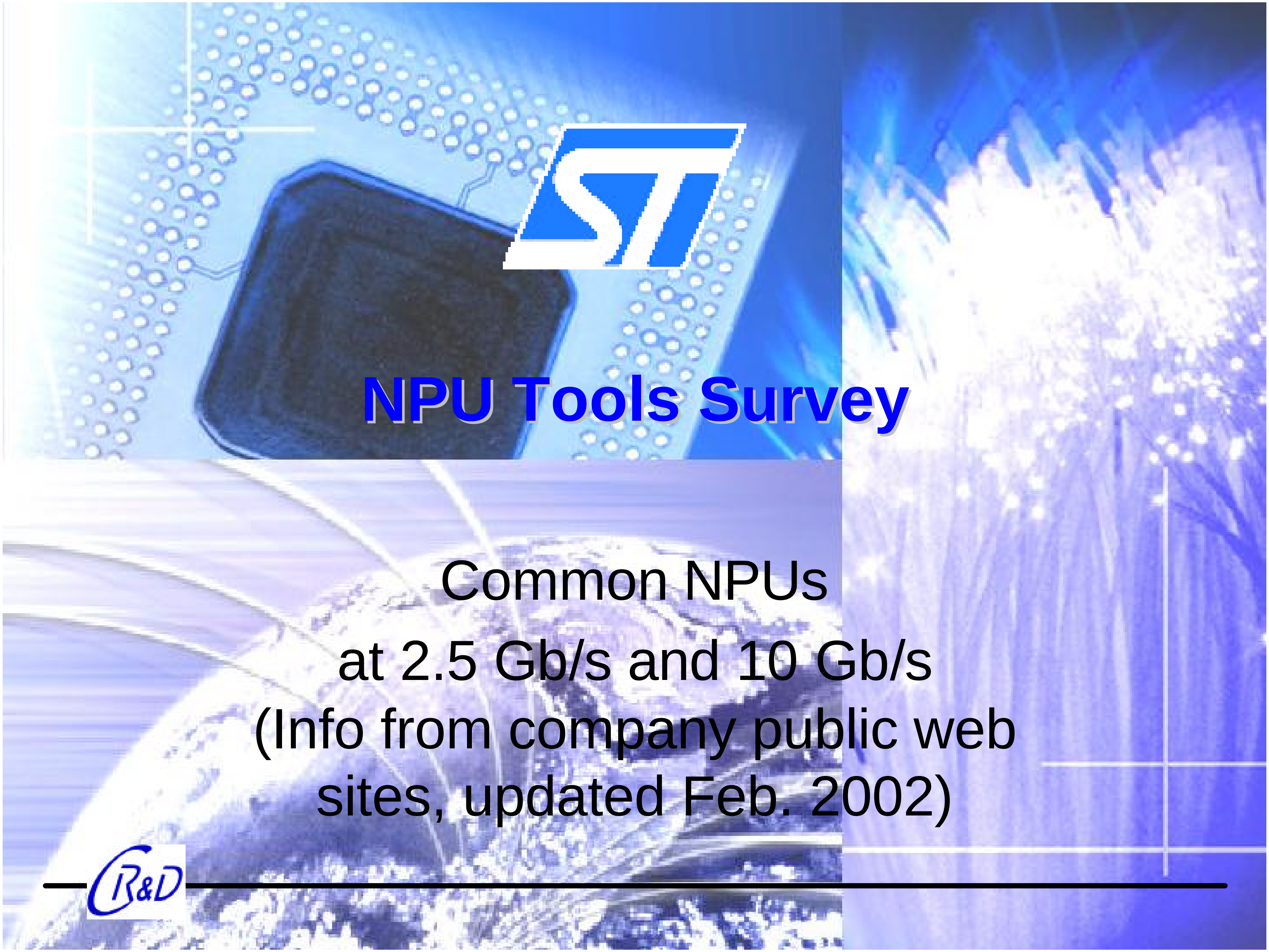
OC48: 3-12 GIPs
OC192: 12-50 GIPs

Power < 10W

- **Packet-oriented ISA**
- **H/W co-processors**
- **Network-on-Chip**
- **Parallel Processors**
- **Pipelined packet processing**

- **Pipelining**
- **Multi-issue, VLIW**
- **Special H/W**
- **Hardware Multi-threading**

- **Deep submicron CMOS**
- **500 to 1000 MHz**



NPU Tools Survey

Common NPUs

at 2.5 Gb/s and 10 Gb/s

(Info from company public web sites, updated Feb. 2002)

NPU **(2.5 Gb)**	**HLL**	**Debug**	**ISS**	**System Model**	**Lib**	**Other**
IBM PowerNP	C: ppc only VxWorks	sim,oce Fr inject	Yes	Yes Tcl/Tk I/F	Yes	Perf. analysis Ref. Board
Agere Payload+	FPL, C VxWorks	Yes IDE	Yes	Throughput modelling	Yes	Packet gen. Ref. Board
Motorola C-5	GCC VxWorks	GDB	Yes	Perf. Model (250X HDL)	C-ware	Perf. analysis Packet gen.
Intel IXP1200	ueC, symb. macro-asm NPOS, OS9	Cycle-acc dbg IDE	Yes	Cycle-acc. FL-API	Teja uWare ACE …	Perf. analysis Packet gen.
Conexant Mxt4400	GCC (2001?)	Yes	Yes	Cycle-acc.	Yes	Perf. analysis Packet gen.
Sitera IQ2000		GDB Sim, jtag oce	Yes	Functional, HDL cosim, Tcl/Tk I/F	Yes	Perf. analysis Packet gen. Ref. board

NPU **(10 Gb)**	**HLL**	**Debug**	**ISS**	**System Model**	**Lib**	**Other**
Lexra Netvortex	CC VRTX OS	m-p/m-thr IDE	Yes	Cycle-based		OCE (JTAG)
AMCC / MMC nP7510	m-p CC h/w OS VxWorks	XRay IDE: WindRiver	Yes		Yes	OCE (JTAG) Ref board
Clear Speed	m-p CC	m-p dbug IDE	Yes	Yes	Yes	Perf. analysis
Si Access	iAtom CC	m-p/m-thr IDE	Yes (C++)	Cycle-based (C++), API	Yes	Perf. analysis Packet gen.
Cisco Toaster	CC (CCC) Cisco IOS	Yes	Yes	Cycle-based	Yes	
Agere Payload+	FPL, ASL VxWorks, Linux OS	Yes, IDE	Yes	Cycle-based, C/Java API's	Yes	Perf. analysis Packet gen. Ref. Board

NPU Tools Summary

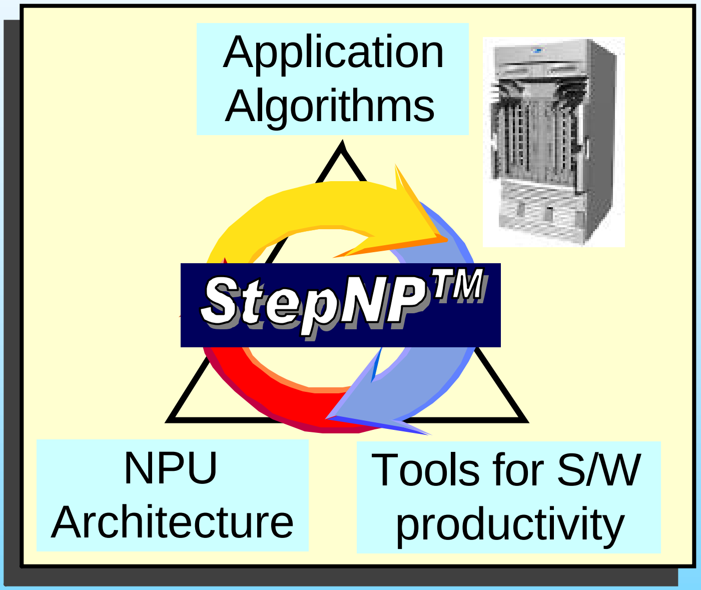
	2.5G	10G
❒ ISS	6/6	6/6
❒ Source-level debugger	4/6	6/6
❒ System-model (cycle-based)	4/6	4/6
❒ O/S support (mostly on ctrl proc)	4/6	4/6
❒ Packet generation	5/6	2/6
❒ HLL compiler	4/6	6/6
❒ Performance analysis	4/6	3/6
❒ Multi-processor compilation	0/6	2/6

Outline

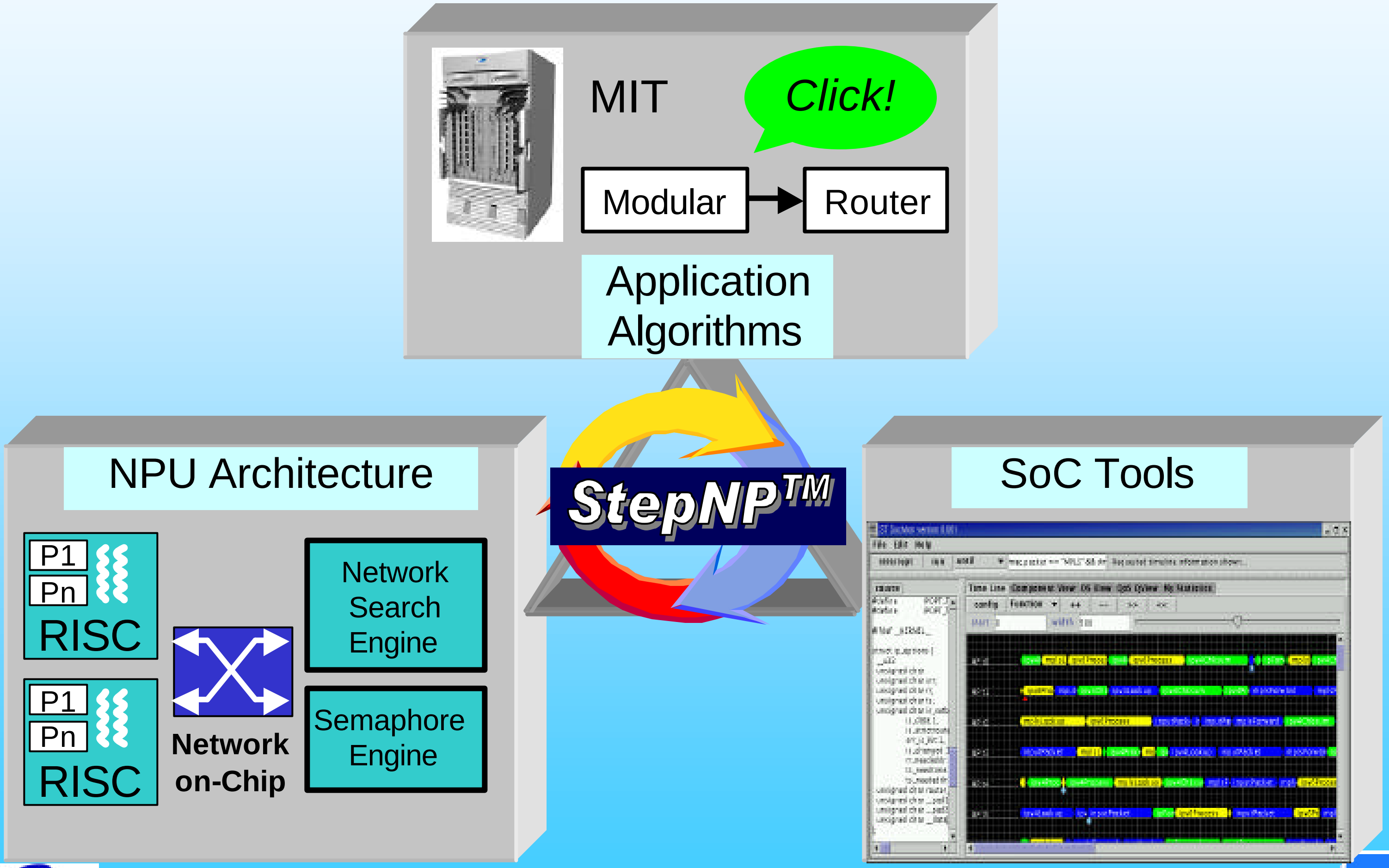
- Network Processors
- ***StepNP^{TM} Research NPU Platform***
  - Overview
  - Parallel processor architecture
  - Router application software
  - SoC tools and methods
- R&D Needs Outlook

StepNP™ Reference Platform

- ❒ System-level Telecom Exploratory Platform for Network Processing
- ❒ For Academic Partners
  - ➢ Canada, International
- ❒ For Commercial CAD partners
- ❒ For ST system design, embedded systems and platform automation R&D teams
  - ➢ Challenging internal driver
- ❒ For ST customers
  - ➢ Reference platform for communication IP
  - ➢ Tool driver for real NPU's



STMicroelectronics StepNP^{TM}
Reference Platform



MIT

Click!

Modular → Router

Application Algorithms

StepNP^{TM}

NPU Architecture

P1
Pn
RISC

P1
Pn
RISC

Network on-Chip

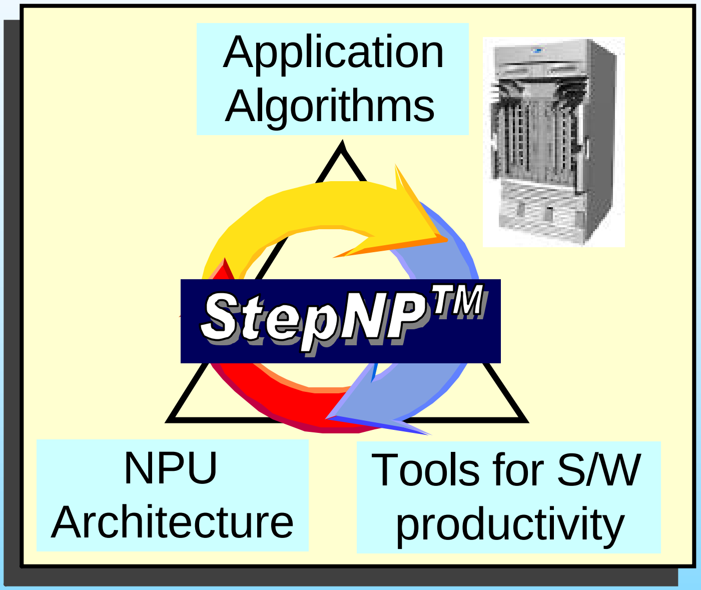
Network Search Engine

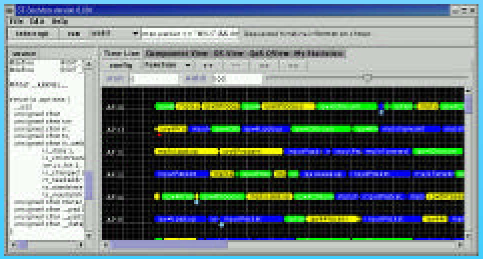
Semaphore Engine

SoC Tools

StepNP^{TM} Reference Platform

- Application S/W
  - MIT Click modular router
- Architecture
  - Processor array: ARM , Tensilica, PowerPC, DLX
  - Interconnect: Split-transaction bus, Octagon, Spin, Amba, ring, crossbar
  - Network search engine, co-proc., semaphore engine, memory, I/O
- Tools
  - ST NPU SoC tools prototype

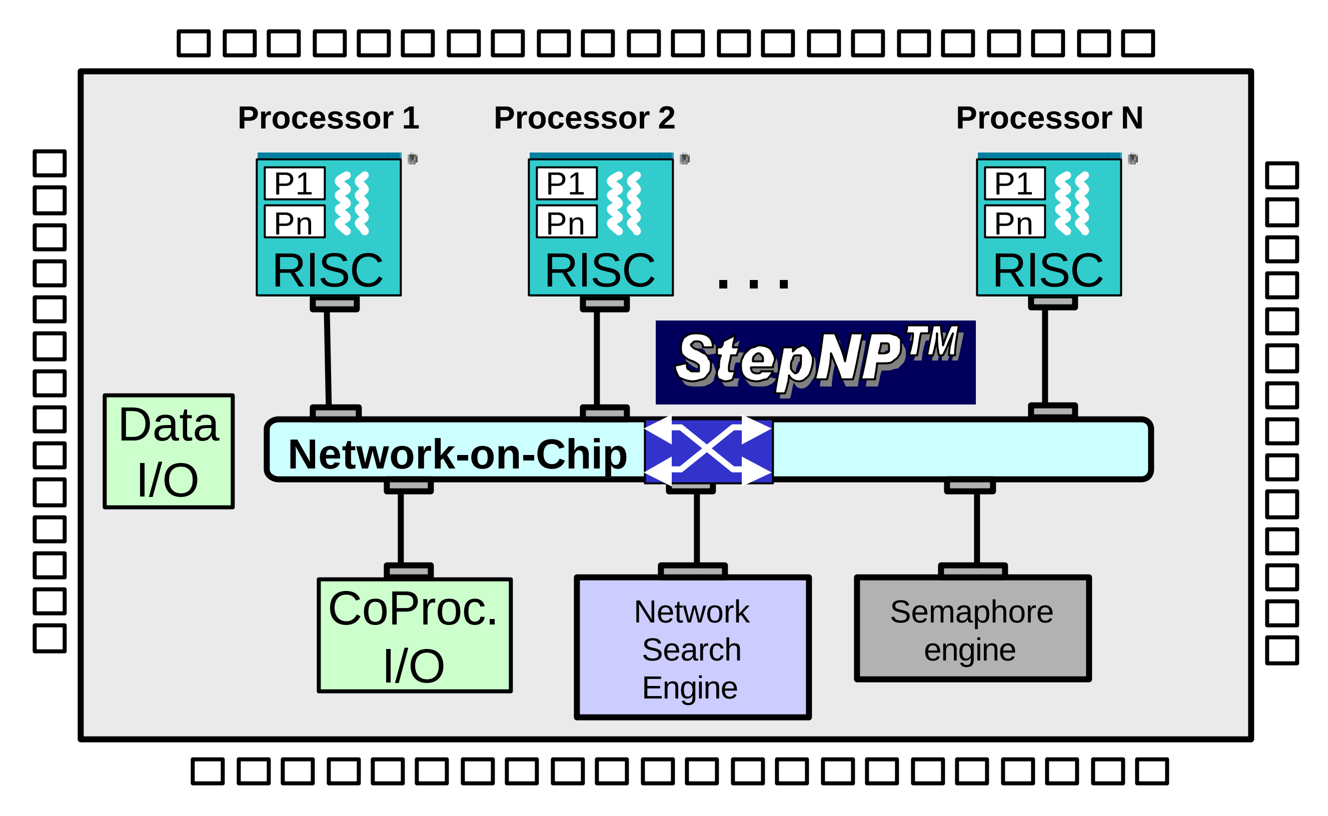




StepNP^{TM} Outline

- ❒What is StepNP^{TM}?
- ❒***Hardware Architecture***
  - ➢Multi-threaded processors
  - ➢Interconnect
- ❒Routing software
- ❒Tools and Methodologies

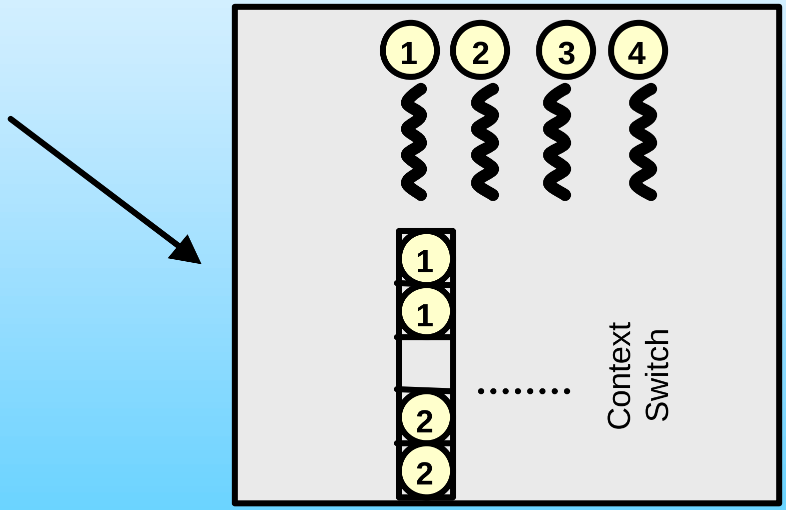
StepNP™ H/W Architecture



- Programmable Multithreaded Network Processor Platform
- Easy replacement of processors: ARM, PowerPC, Tensilica

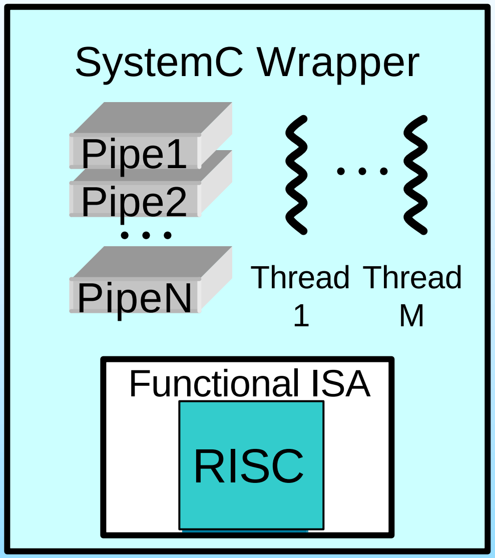
H/W Multi-threaded Processor

- Principal approach :
  - Block interleaving technique: The instructions of a thread are executed successively until an event occurs that may cause latency (ie: read memory data)

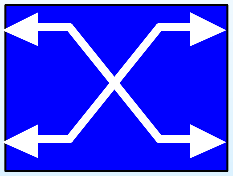


StepNP^{TM} Reference Processor

- Hardware multi-threaded processor
  - Standard approach for network processors
    - Hide latency via zero overhead thread context switch
  - Configurable number of hardware threads per processor (default 8)
  - Configurable pipeline depth (default 4)
- Each thread executes functional instruction set (e.g. ARM, PowerPC)
  - StepNP primary focus not on instruction set of NPU
- SystemC cycle-accurate model
  - Encapsulation of functional ISS (ARM, DLX, PowerPC, etc.)

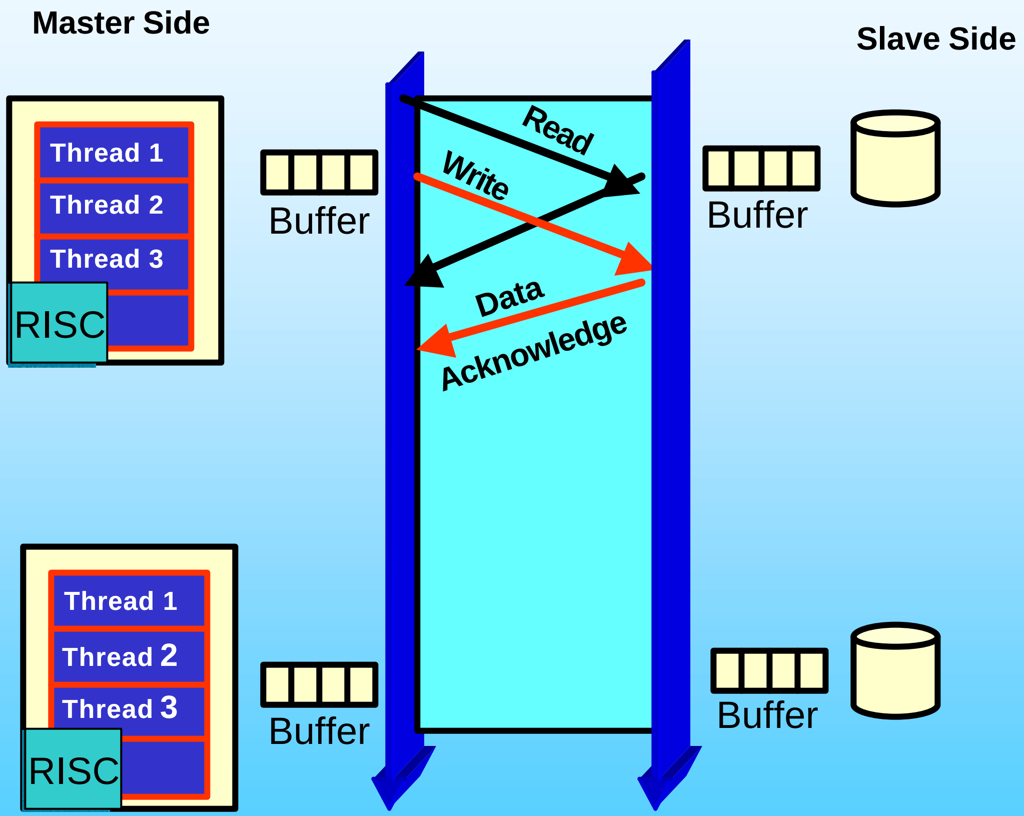


StepNP^{TM} Interconnect Models



- ❒ A general framework for interconnects, to allow architecture exploration
- ❒ Multi-level modelling
  - ➢ Transaction-level to cycle-accurate models
- ❒ Various interconnect architectures
  - ➢ Split-transaction bus, SPIN (Paris LIP6), Octagon (ST), ring, crossbar, Amba bus, Sonics
  - ➢ Currently developing various channel implementations
    - ➪ Functional, transactional, cycle-based
  - ➢ Distributed simulation is another form of implementation

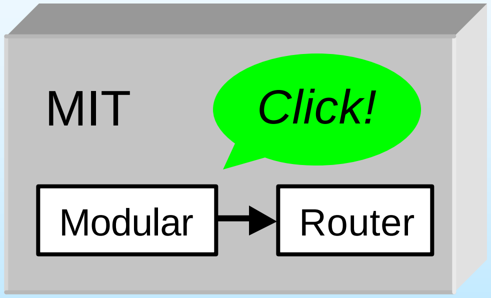
Split Transaction Bus Model



StepNP^{TM} Outline

- ❒ What is StepNP^{TM}?
- ❒ Hardware Architecture
  - ➢ Multi-threaded processors
  - ➢ Interconnect
- ❒ ***Routing software***
- ❒ Tools and Methodologies

StepNP^{TM} Routing Software

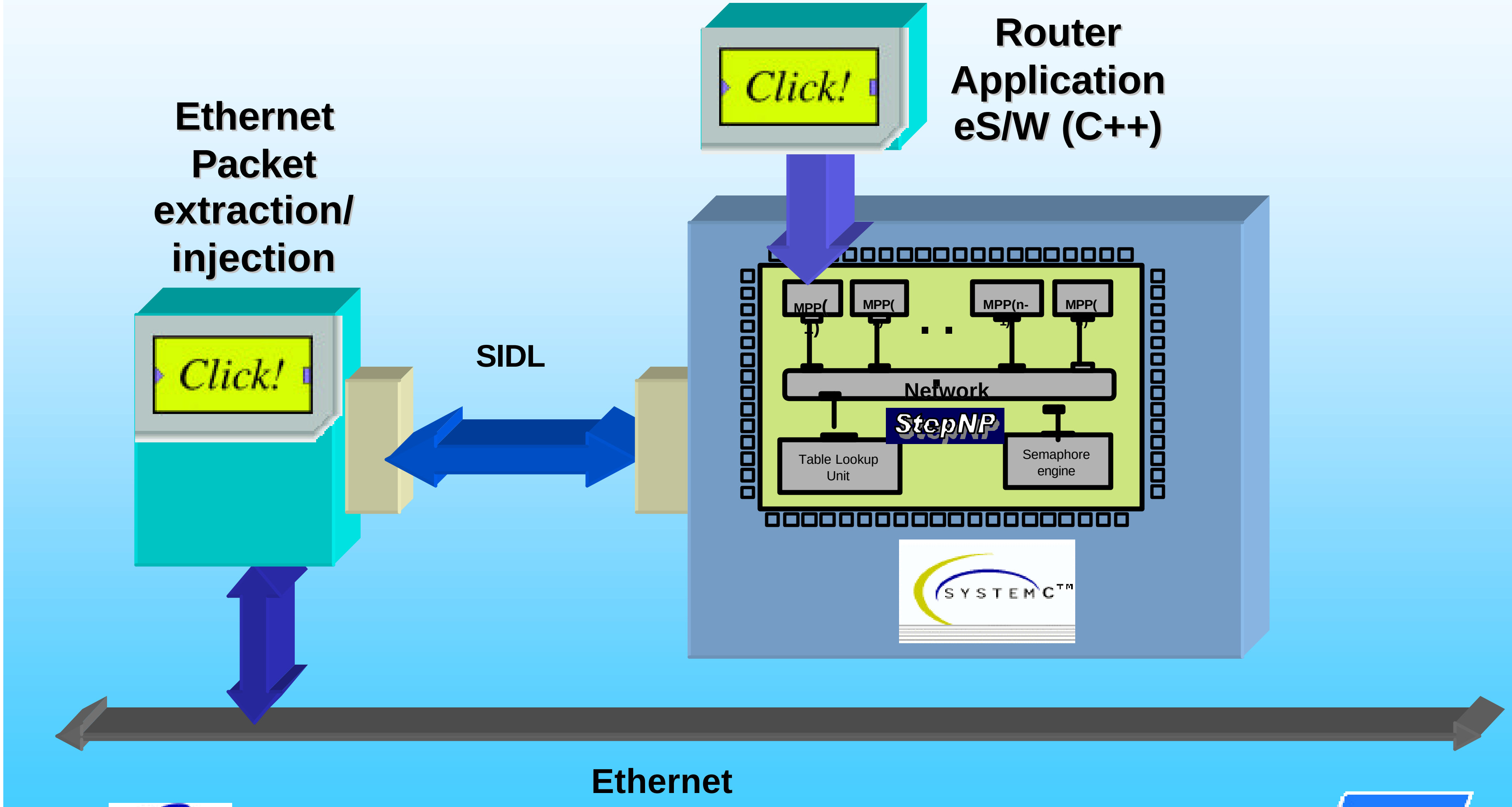


- ❒ Uses the MIT Click router software
  - ➢ http://www.pdos.lcs.mit.edu/click
- ❒ Click is
  - ➢ Available under open-source license
  - ➢ modular, flexible, configurable
- ❒ Fine-grained modularity of Click allows efficient parallel execution.
- ❒ Router configurations expressed in a simple language

StepNP^{TM} Routing Application

- ❐ A simple network translation (NAT) application
- ❐ Easy to set up testbench environment
- ❐ Emulates a “virtual host”, by capturing packets off real ethernet, performs NAT algorithms, and injects back on to ethernet.
- ❐ Can set up real network applications (telnet or web browsing) with the virtual host, and watch StepNP simulator processing packets.

Application and Architecture Testbench



Router Application eS/W (C++)

Ethernet Packet extraction/ injection

SIDL

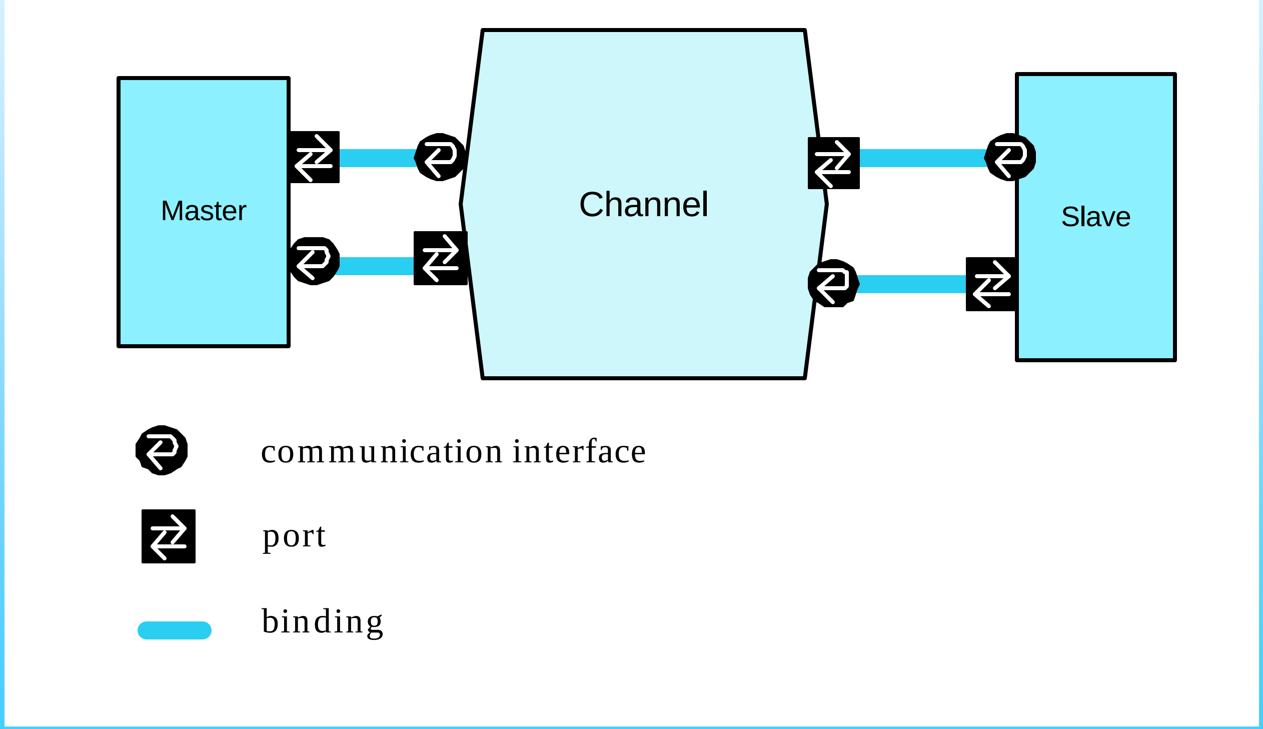
Ethernet

StepNP^{TM} Outline

- ❒What is StepNP^{TM}?
- ❒Hardware Architecture
  - ➢Multi-threaded processors
  - ➢Interconnect
- ❒Routing software
- ❒***Tools and Methodologies***

SOCP Transaction-Level Model Channel Interface

- ❒ Based on OCP (VCI) semantics
  - ➢ Transaction-level interface (no pins, signals, clk)

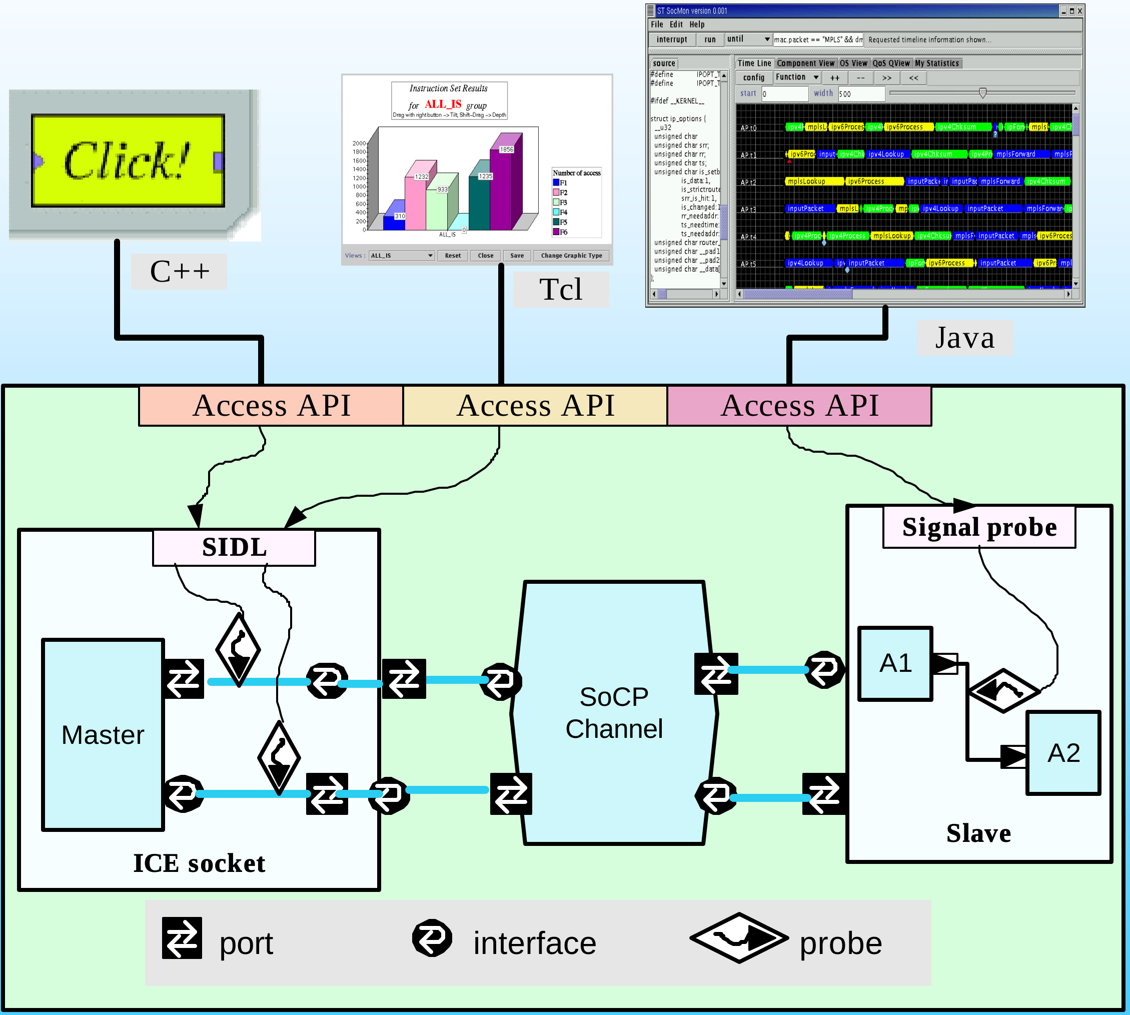


SOCP Channel Interface (contd.)

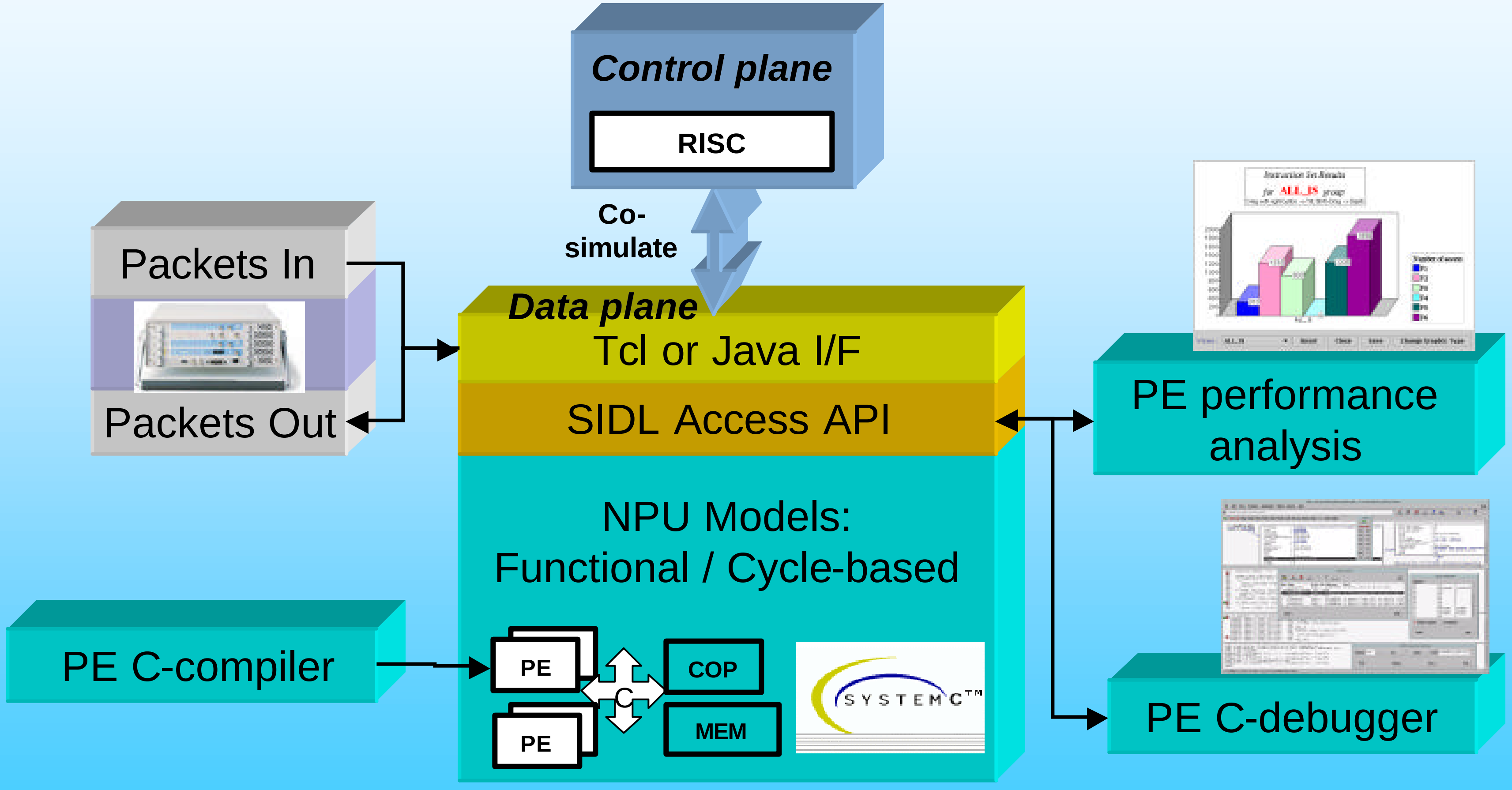
- ❐ Simulation speed results for functional SOCP implementation
  - ➢Unix Ultra80 (450 MHz) 250 KHz (to 1.2 MHz)
  - ➢Linux PC (800 MHz) 1.5 MHz (to 1.5 MHz)
- ❐ Distributed simulation using SOCP
  - ➢Special SOCP channel uses TCP/IP, SystemC modules assigned manually to W/S
  - ➢Lower bound on simulation speed (if limited only by TCP/IP communication)
    - ➪30 KHz over TCP/IP 100Mb Ethernet
  - ➢Exploring Myrinet implementation (U. of Montreal)

SIDL (SystemC IDL)

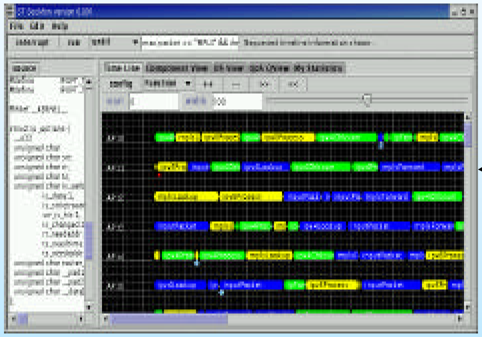
- Similar to CORBA IDL (lighter, SysC based)
  - Language-specific APIs generated automatically: C++, Java, Tcl, XML (FwLib)
- API types
  - Instrospection API
  - SOCP ICE API
  - Domain-specific API
- Part of instrumentation methodology
  - FlexPerf2, SysProbe links

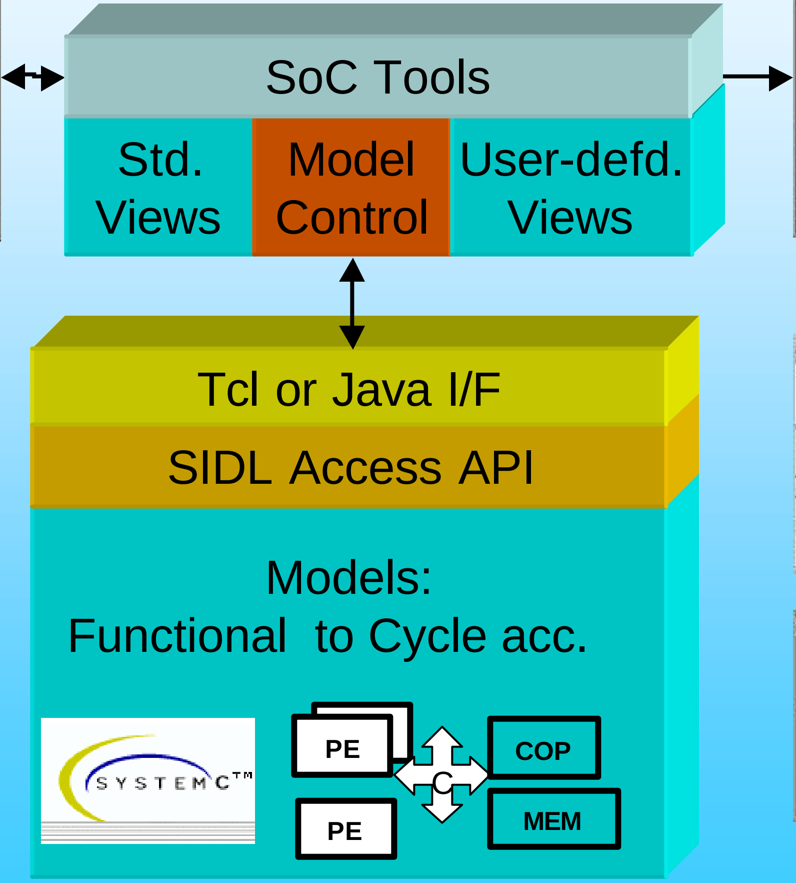


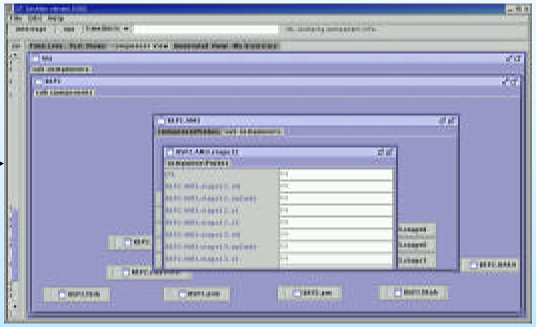
NPU eS/W Tools

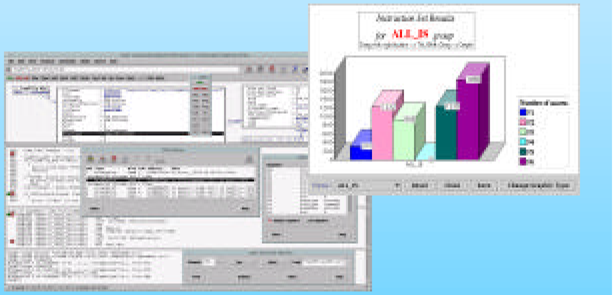


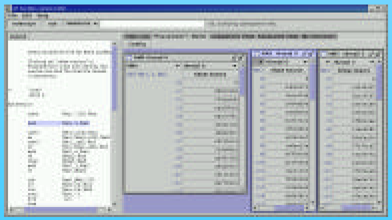
StepNP^{TM} Tool Platform







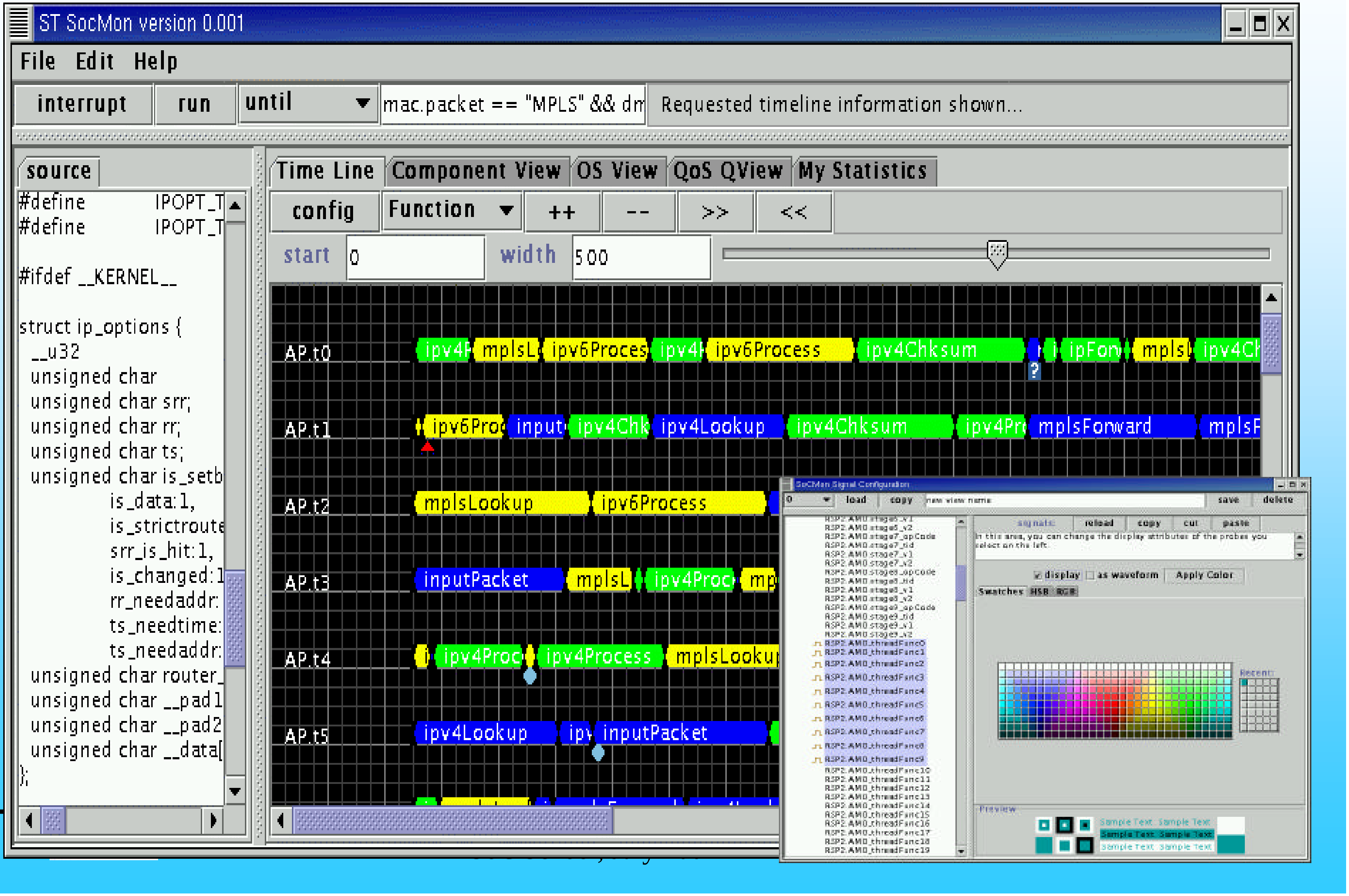




SoC-level User-extendable platform

- ❒ SoC execution debug
  - ➢multi-threaded support, backtracking features
- ❒ SoC execution analysis
  - ➢Logical, temporal, spatial perspectives
  - ➢Range of abstraction levels supported
- ❒ SoC execution control
  - ➢Well-defined and powerful interface to SoC model
  - ➢Enables integration with other system software
  - ➢Interfaces for C++ and scripting languages

SoC Tools – Timeline View



until: mac.packet == "MPLS" && dm

Requested timeline information shown...

start 0

width 500

```
#define          IPOPT_T
#define          IPOPT_T

#ifdef __KERNEL__

struct ip_options {
  __u32
  unsigned char
  unsigned char srr;
  unsigned char rr;
  unsigned char ts;
  unsigned char is_setb
                is_data:1,
                is_strictroute
                srr_is_hit:1,
                is_changed:1
                rr_needaddr:
                ts_needtime:
                ts_needaddr:
  unsigned char router_
  unsigned char __pad1
  unsigned char __pad2
  unsigned char __data[
};
```

0

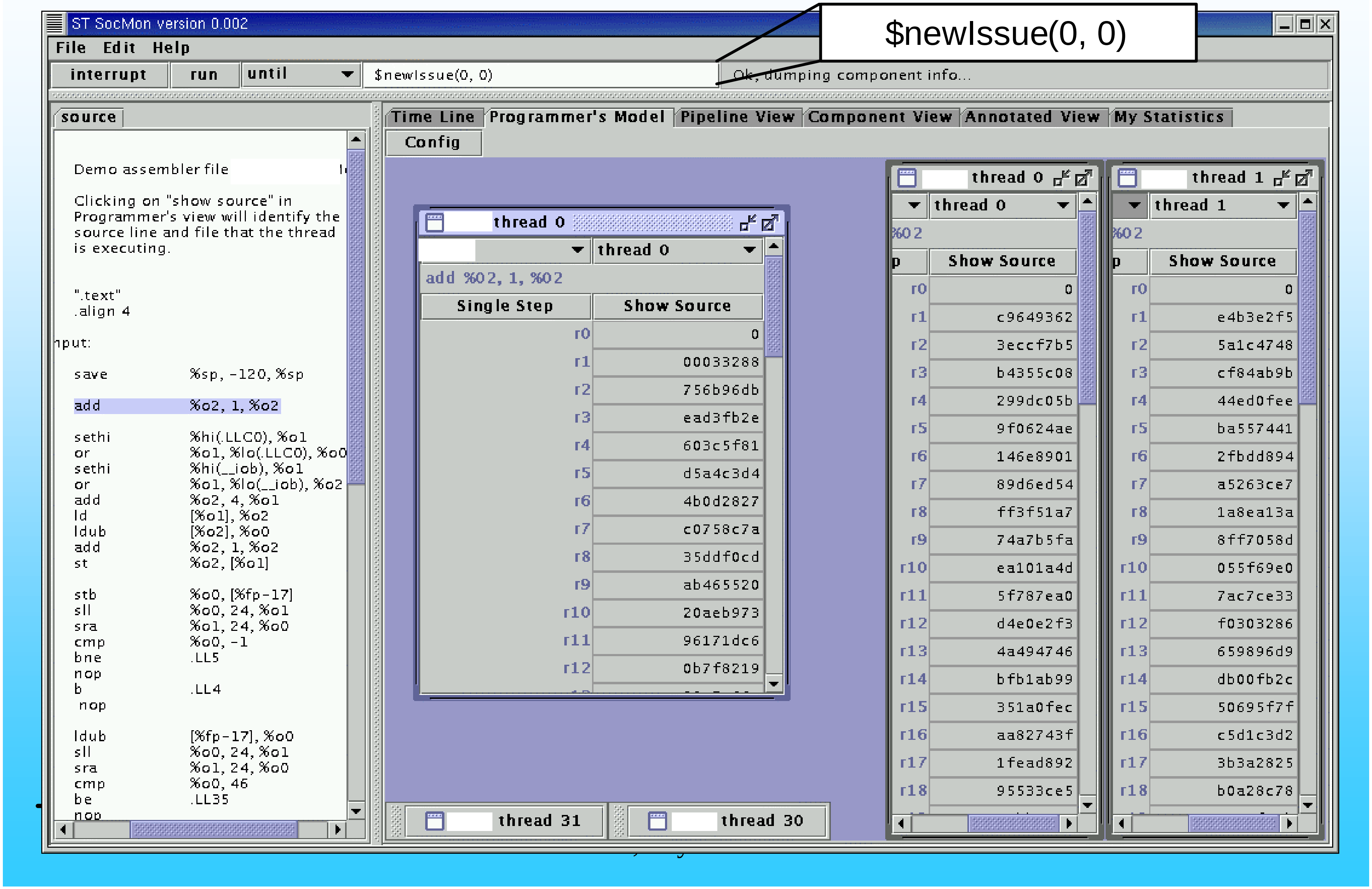
new view name

```
RSP2.AMO.stage6_v1
RSP2.AMO.stage6_v2
RSP2.AMO.stage7_opCode
RSP2.AMO.stage7_tid
RSP2.AMO.stage7_v1
RSP2.AMO.stage7_v2
RSP2.AMO.stage8_opCode
RSP2.AMO.stage8_tid
RSP2.AMO.stage8_v1
RSP2.AMO.stage8_v2
RSP2.AMO.stage9_opCode
RSP2.AMO.stage9_tid
RSP2.AMO.stage9_v1
RSP2.AMO.stage9_v2
RSP2.AMO_threadFunc0
RSP2.AMO_threadFunc1
RSP2.AMO_threadFunc2
RSP2.AMO_threadFunc3
RSP2.AMO_threadFunc4
RSP2.AMO_threadFunc5
RSP2.AMO_threadFunc6
RSP2.AMO_threadFunc7
RSP2.AMO_threadFunc8
RSP2.AMO_threadFunc9
RSP2.AMO_threadFunc10
RSP2.AMO_threadFunc11
RSP2.AMO_threadFunc12
RSP2.AMO_threadFunc13
RSP2.AMO_threadFunc14
RSP2.AMO_threadFunc15
RSP2.AMO_threadFunc16
RSP2.AMO_threadFunc17
RSP2.AMO_threadFunc18
RSP2.AMO_threadFunc19
```

In this area, you can change the display attributes of the probes you select on the left.

☑ display ☐ as waveform

SoC Tools – Programmer’s View



$newIssue(0, 0)

```
$newIssue(0, 0)
```

Ok, dumping component info...

```
Demo assembler file

Clicking on "show source" in
Programmer's view will identify the
source line and file that the thread
is executing.


".text"
.align 4

nput:

save            %sp, -120, %sp

add             %o2, 1, %o2

sethi           %hi(.LLC0), %o1
or              %o1, %lo(.LLC0), %o0
sethi           %hi(__iob), %o1
or              %o1, %lo(__iob), %o2
add             %o2, 4, %o1
ld              [%o1], %o2
ldub            [%o2], %o0
add             %o2, 1, %o2
st              %o2, [%o1]

stb             %o0, [%fp-17]
sll             %o0, 24, %o1
sra             %o1, 24, %o0
cmp             %o0, -1
bne             .LL5
nop
b               .LL4
 nop

ldub            [%fp-17], %o0
sll             %o0, 24, %o1
sra             %o1, 24, %o0
cmp             %o0, 46
be              .LL35
nop
```

thread 0

add %o2, 1, %o2

	Show Source
r0	0
r1	00033288
r2	756b96db
r3	ead3fb2e
r4	603c5f81
r5	d5a4c3d4
r6	4b0d2827
r7	c0758c7a
r8	35ddf0cd
r9	ab465520
r10	20aeb973
r11	96171dc6
r12	0b7f8219

thread 0

%o2

p	Show Source
r0	0
r1	c9649362
r2	3eccf7b5
r3	b4355c08
r4	299dc05b
r5	9f0624ae
r6	146e8901
r7	89d6ed54
r8	ff3f51a7
r9	74a7b5fa
r10	ea101a4d
r11	5f787ea0
r12	d4e0e2f3
r13	4a494746
r14	bfb1ab99
r15	351a0fec
r16	aa82743f
r17	1fead892
r18	95533ce5

thread 1

%o2

p	Show Source
r0	0
r1	e4b3e2f5
r2	5a1c4748
r3	cf84ab9b
r4	44ed0fee
r5	ba557441
r6	2fbdd894
r7	a5263ce7
r8	1a8ea13a
r9	8ff7058d
r10	055f69e0
r11	7ac7ce33
r12	f0303286
r13	659896d9
r14	db00fb2c
r15	50695f7f
r16	c5d1c3d2
r17	3b3a2825
r18	b0a28c78

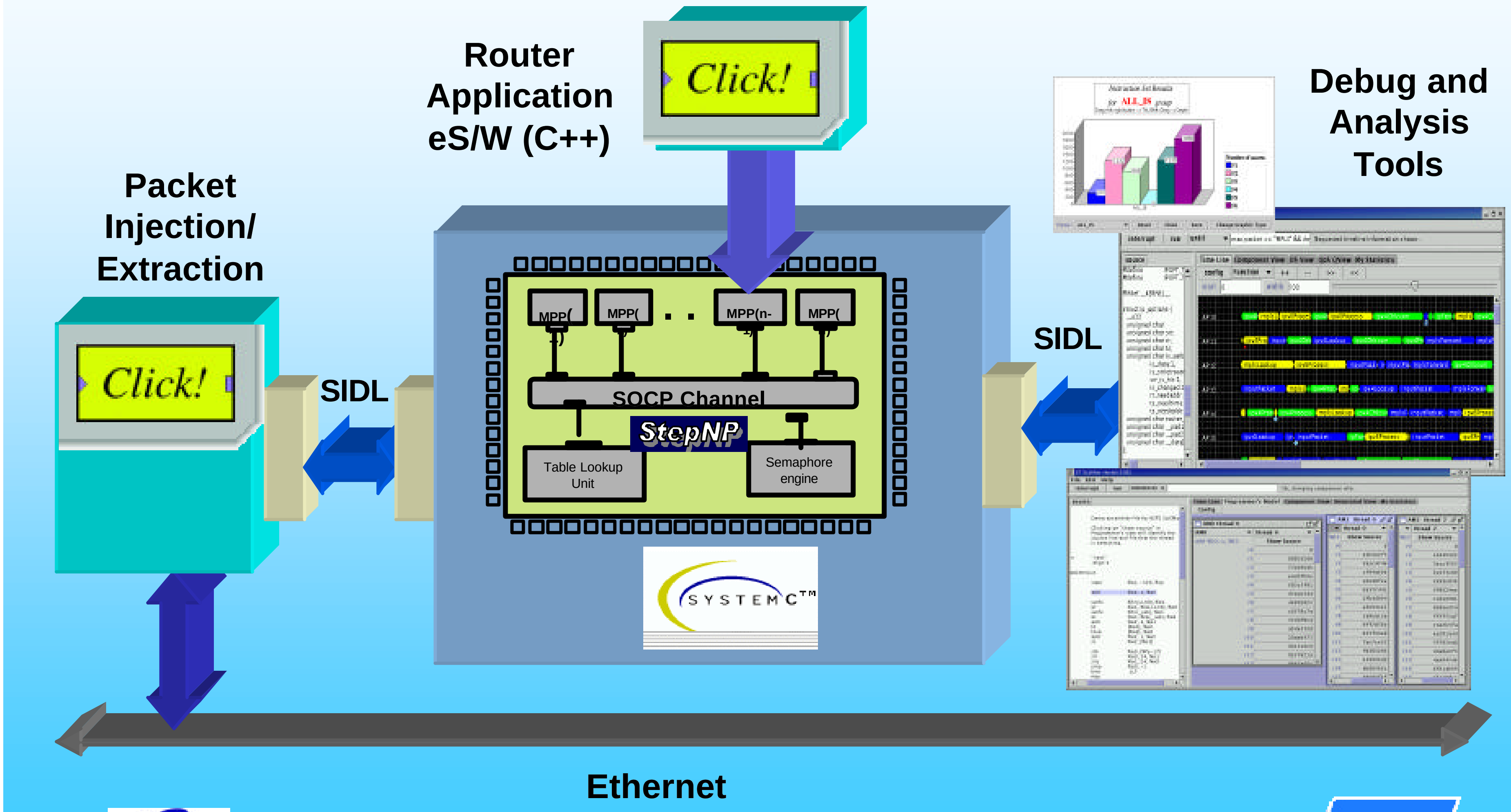
SoC Tools - Component View

- Shows layout of model components in 2D format
- Model component structure automatically obtained from SystemC model API
- Component graphical representation augmented with configurable information from:
  - Model signals, attribute values
  - software variable values
  - user-defined function of these primitive values
  - user-defined functions driving presentation objects

Soc Tools - User Defined Views

- ❒ Develop framework that allows end-user to construct application-specific views
- ❒ Two levels: scripting and Java-level
- ❒ Scripting (“easy” extensions): User can write scripts accessing model state, which drives provided presentation objects
- ❒ Java-level (power user): Provide Java SoC Monitor IDE environment and framework classes, Java Beans, etc, for extending the user interface
- ❒ Framework classes: for model access, control, and presentation

StepNP^{TM}: Putting it all Together



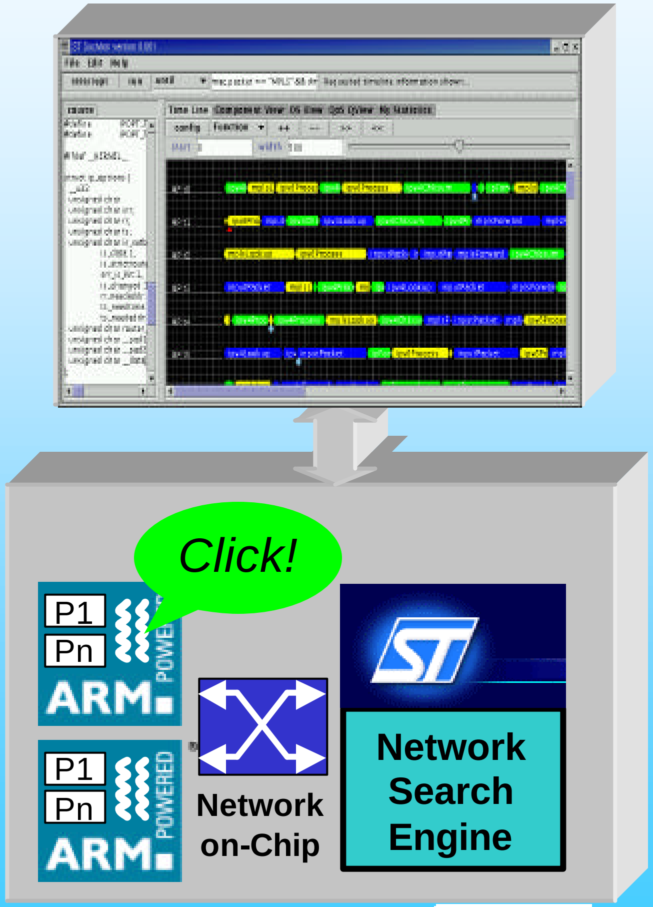
Router Application eS/W (C++)

Packet Injection/ Extraction

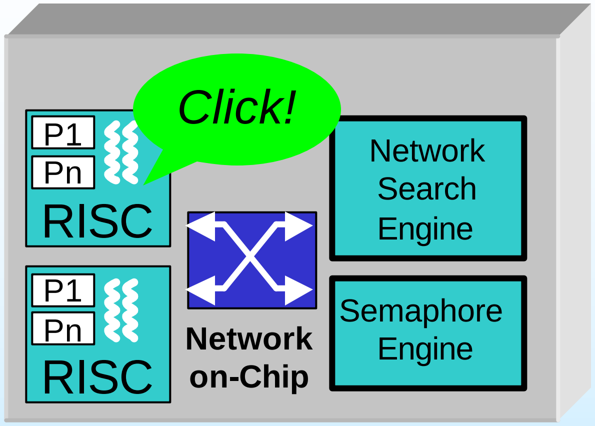
Debug and Analysis Tools

StepNP^{TM} Application: NSE Reference Platform

- Customer learning tool
- Illustrate programming and use of NSE
- Example usage for typical routing applications
  - Applications developed using Click Router (NAT, IPv4)
  - BGP packet traces
- Visualization and control environment

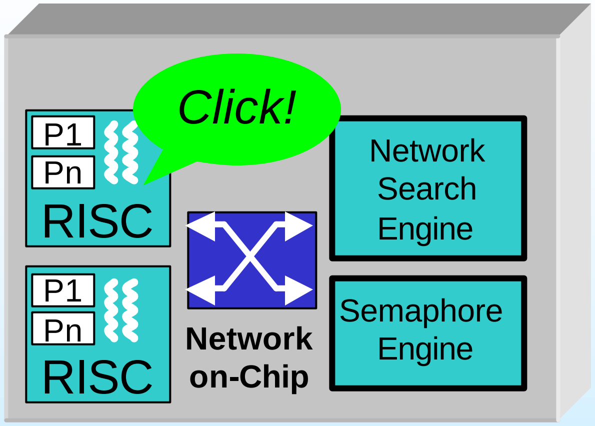


StepNP^{TM} Benefits



- ❐ Driver for ST SoC tools/methods
  - ➢ Transaction-level modeling
  - ➢ Working on standard instrumentation approach
- ❐ Driver for ST FlexWare eS/W tools
  - ➢ FlexPerf2 performance analysis
  - ➢ SoC-level debug, FlexCC-based NPU C compiler, ISS tools
- ❐ Reference platform for customers
  - ➢ Including models of ST IP:
    Network Search Engine, Octagon network-on-chip
- ❐ Environment reused for customer NPUs
- ❐ Vehicle for university interaction

StepNP^{TM} Benefits (2)



- ❒ Network Processing training
  - ➢ Architecture: Semaphore, H/W multi-threading, etc.
  - ➢ Application: Click router application
- ❒ Commercial CAD tool evaluation
  - ➢ Modelling tools: SystemC, ISS, System-level modeling
  - ➢ Verification tools
- ❒ Emerging CAD tool driver:
  - ➢ Combines two key elements of next gen. SoC's
    1. Multi-processor
    2. Network on Chip

StepNP^{TM} Current Work

- ❒ Integration with ST Network Search Engine
  - ➢Extend SoC debug/analysis tool
- ❒ SystemC instrumentation methodologies
- ❒ SOCP transaction-level I/F validation
  - ➢Integrate with instrumentation methodology
- ❒ Beta release targeted for Sept. 2002
  - ➢To selected Univ. and CAD partners
- ❒ Open Source in 2003

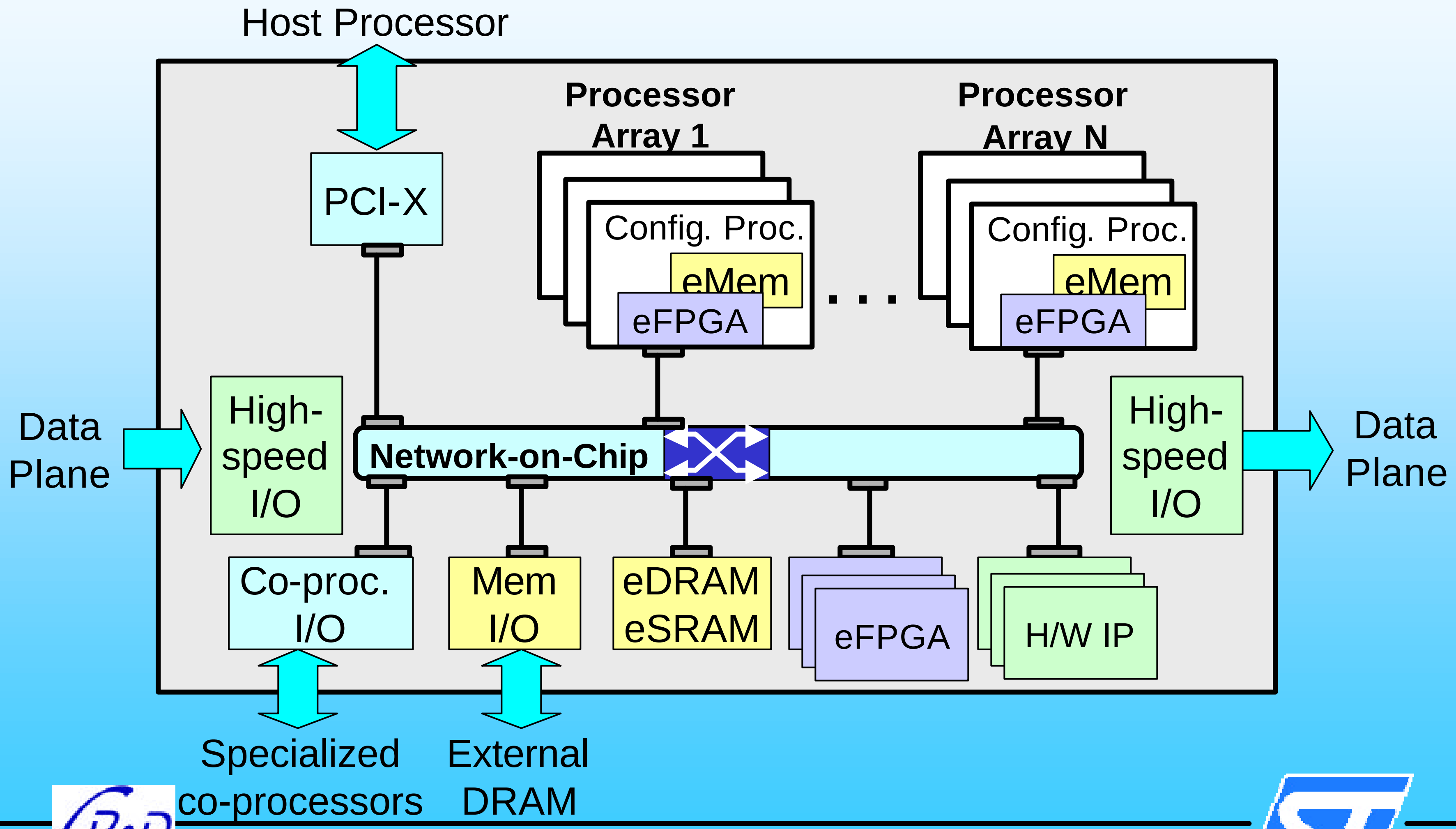
Next ‘Steps’

❐ New Processors,co-processors	
➢ Tensilica (+ bit-manip. instrns.)	Ec. Poly. Montreal
➢ LisaTek model integration	Univ. Aachen, ST
❐ New Interconnect	
➢ SPIN	LIP6 lab
➢ ST Octagon, AMBA bus	Univ. Montreal
❐ Application S/W	
➢ 4~6 representative applications, manually mapped onto threads/processors	ST/Ottawa, LIP6
❐ SoC Tools	
➢ Continue development current prototype	ST/Ottawa
➢ SystemC instrumentation methods	ST/Ottawa

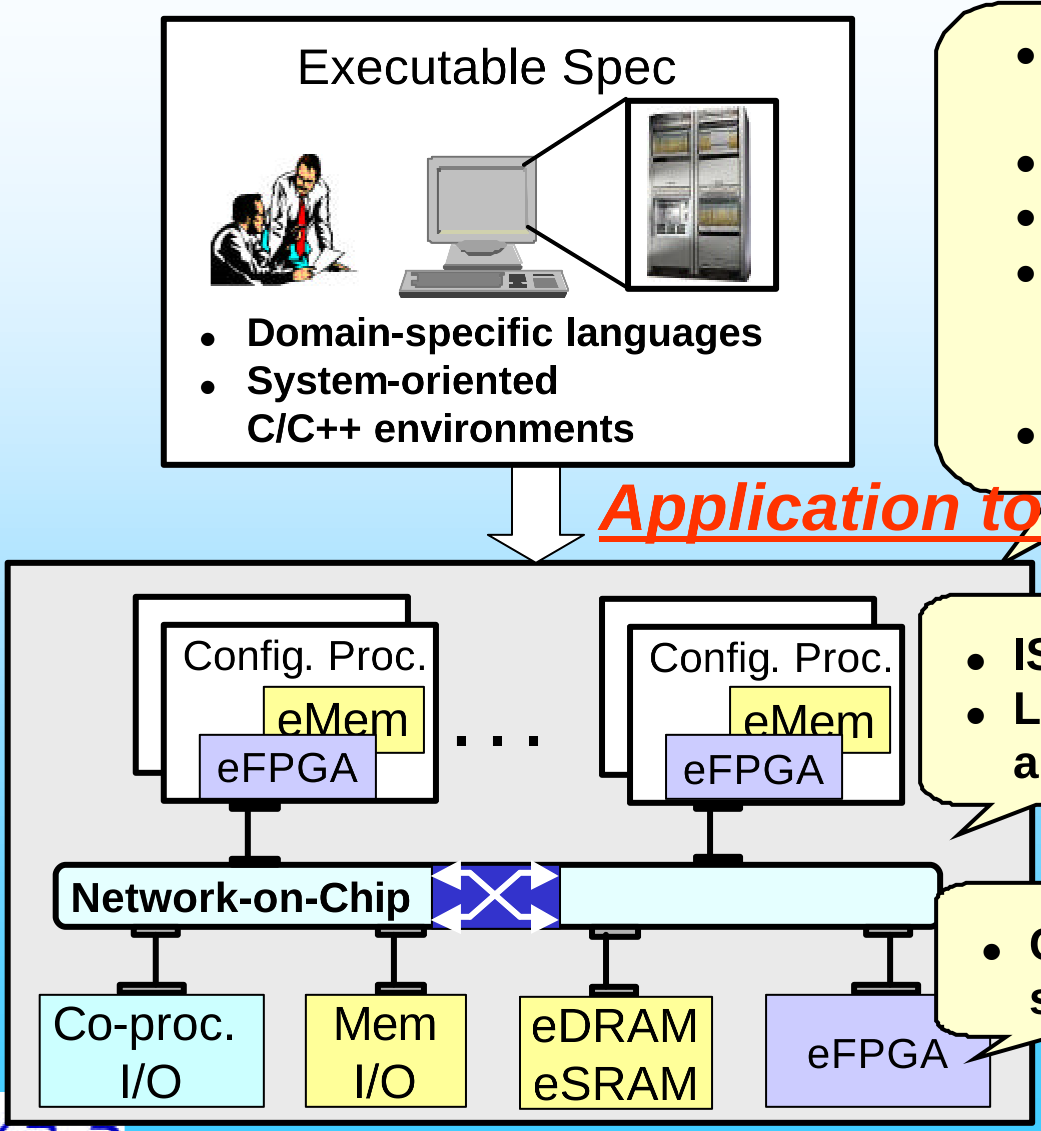
Outline

- ❒ Network Processors
- ❒ *StepNP*^{TM} Research NPU Platform
- ❒ ***R&D Needs Outlook***
  - ➢ Communication processor architecture of the future?
  - ➢ Key automation technologies
  - ➢ University interaction

Communication Processor Architecture of the Future?



Key Automation Technologies



Executable Spec

- Domain-specific languages
- System-oriented C/C++ environments

- Multi-processor partitioning and allocation
- Memory allocation
- Communication allocation
- High-performance S/W compilation for domain-specific processors
- Multi-processor RTOS

Application to platform mapping

Config. Proc.

eMem

eFPGA

. . .

Config. Proc.

eMem

eFPGA

Network-on-Chip

Co-proc. I/O

Mem I/O

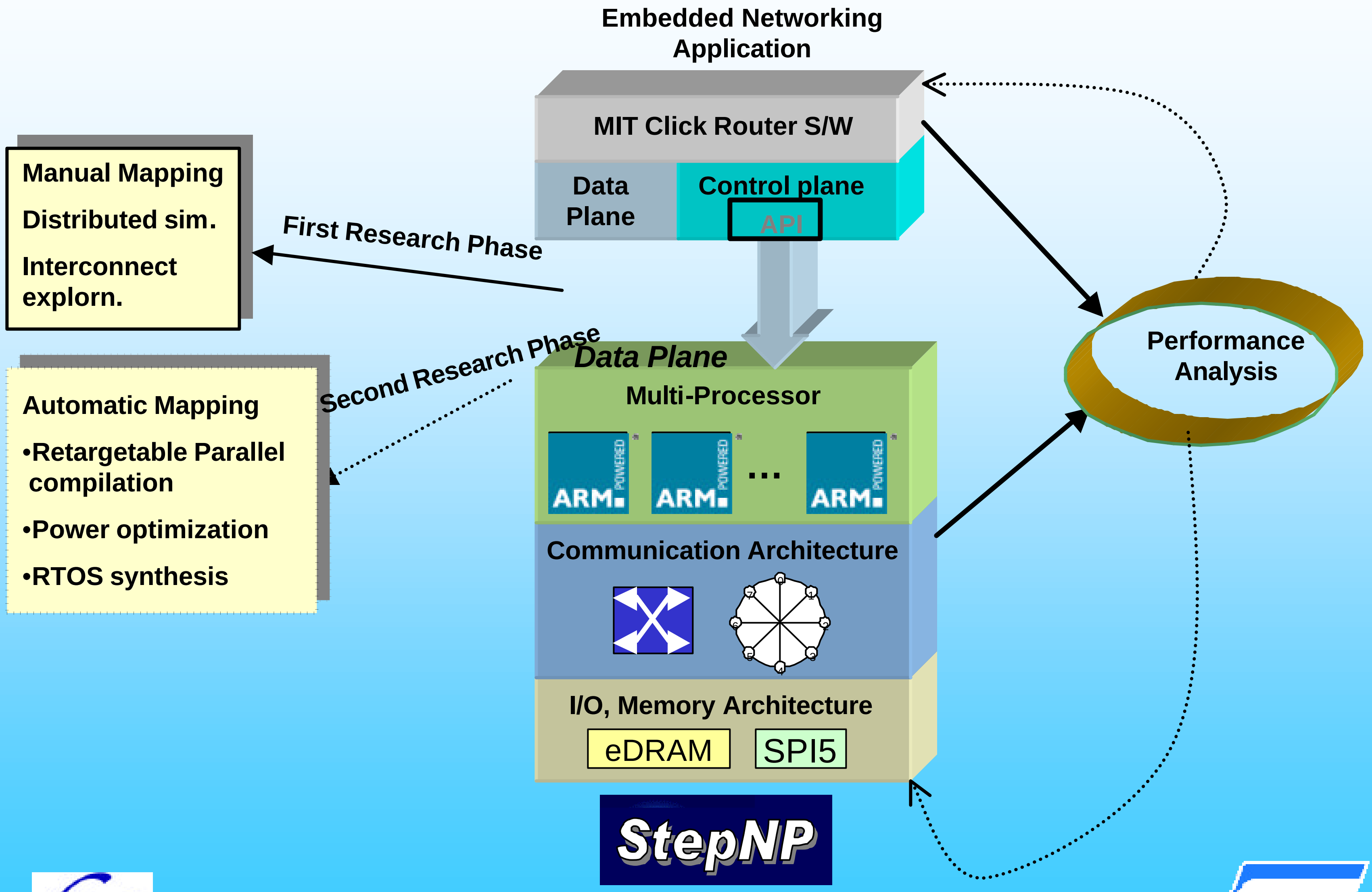
eDRAM eSRAM

eFPGA

- ISA synthesis
- Library of base applications

- Co-processor synthesis

University Research Framework



Embedded Networking Application

MIT Click Router S/W

Data Plane

Control plane

API

First Research Phase

Manual Mapping

Distributed sim.

Interconnect explorn.

Second Research Phase

Automatic Mapping

- Retargetable Parallel compilation
- Power optimization
- RTOS synthesis

Data Plane

Multi-Processor

Communication Architecture

I/O, Memory Architecture

eDRAM

SPI5

Performance Analysis

StepNP

StepNP^{TM} Cooperation

- ❐ Academic cooperation
  - ➢ Univ. Montreal and Ecole Polytechnique Montreal
  - ➢ U. Paris LIP6 (SPIN SoC packet-based network)
  - ➢ U. Aachen (ISS/SystemC cosim, eS/W tools)
  - ➢ U. Toronto (Multi-processor compilation)
  - ➢ Polit. Milan (Netw.-on-chip power optimization)
  - ➢ CMC (Canada-wide StepNP distribution)
  - ➢ Others tbd
- ❐ CAD vendor cooperation
  - ➢ The usual suspects

Conclusion

- ❒ NPU's are excellent driver for next generation SoC tools & methods
- ❒ StepNP^{TM} reference platform
  - ➢SystemC architecture model framework
  - ➢MIT Click Router application software
  - ➢SoC methods (SOCP, SIDL) and tool framework
  - ➢Plan for open source in 2003
- ❒ Many R&D challenges
  - ➢Network-on-Chip architectures
  - ➢Multi-processor compilation