STP Engine, a C-based Programmable HW Core featuring Massively Parallel and Reconfigurable PE Array:

Its Architecture, Tool, and System Implications

2009/04/17

Masato Motomura
1st SoC Operations Unit
NEC Electronics Corporation

Outline

- Needs for programmable HW core
- STP engine
  - Architecture
  - Execution model
  - Tool
- XBridge: a 90nm system LSI with STP engine
- Demonstration
- Conclusion

As On-Chip System Integration Evolves ...

- An LSI development project
  - Requires longer time,
  - Demands larger amount of engineers,
  - Costs more expense,
  - Faces severer risks, (or failure cost)

  generation by generation

- As such, an SoC (System-on-Chip) requires larger volume of shipments
- But the system integration itself naturally means dedication to some specific product category

  => **"SoC dilemma"**

To be HW, or be SW …

NEC

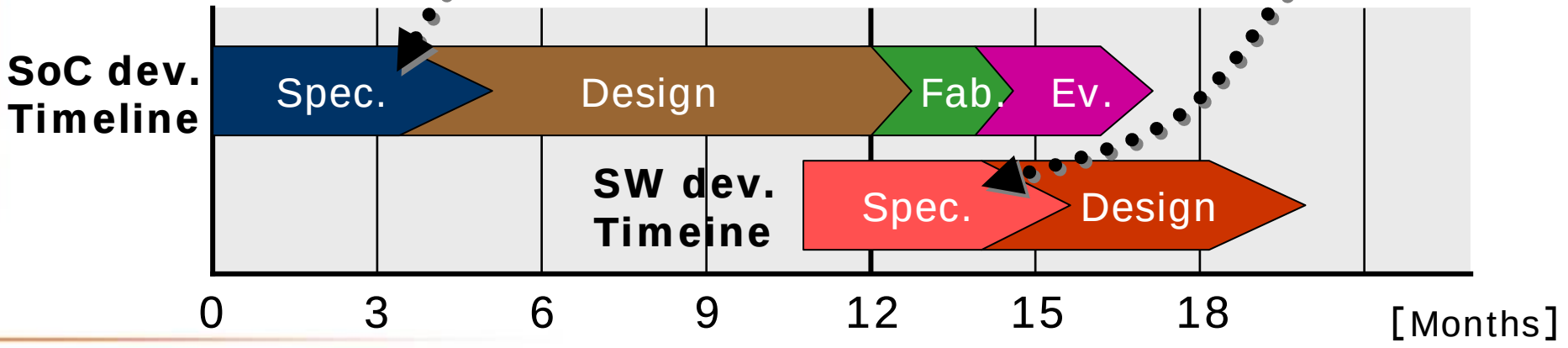
“We need THIS fantastic function in our set to beat competitors!”

Ok, let’s put it in HW!

- Good performance
- Occupies small Si area
- Needs more than a year
- Lacks flexibility

… Make it as SW instead.

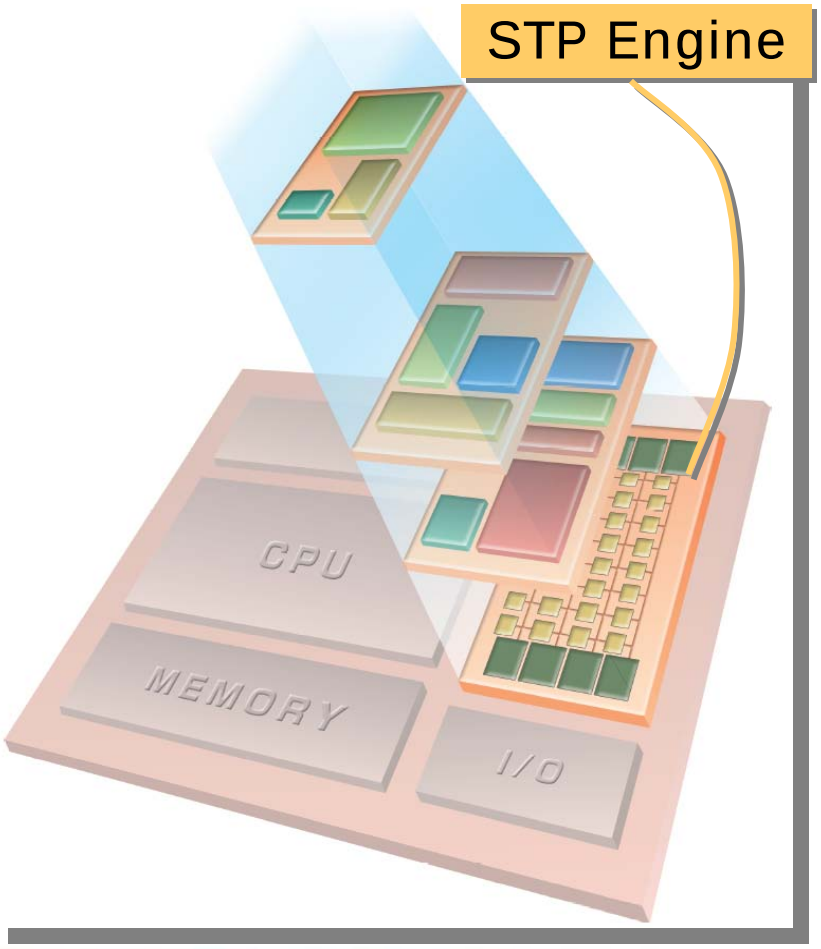
- Requires high-speed CPU
- Still lacks performance
- Needs only a few months
- Can put newest algorithm



Our Solution: STP Engine

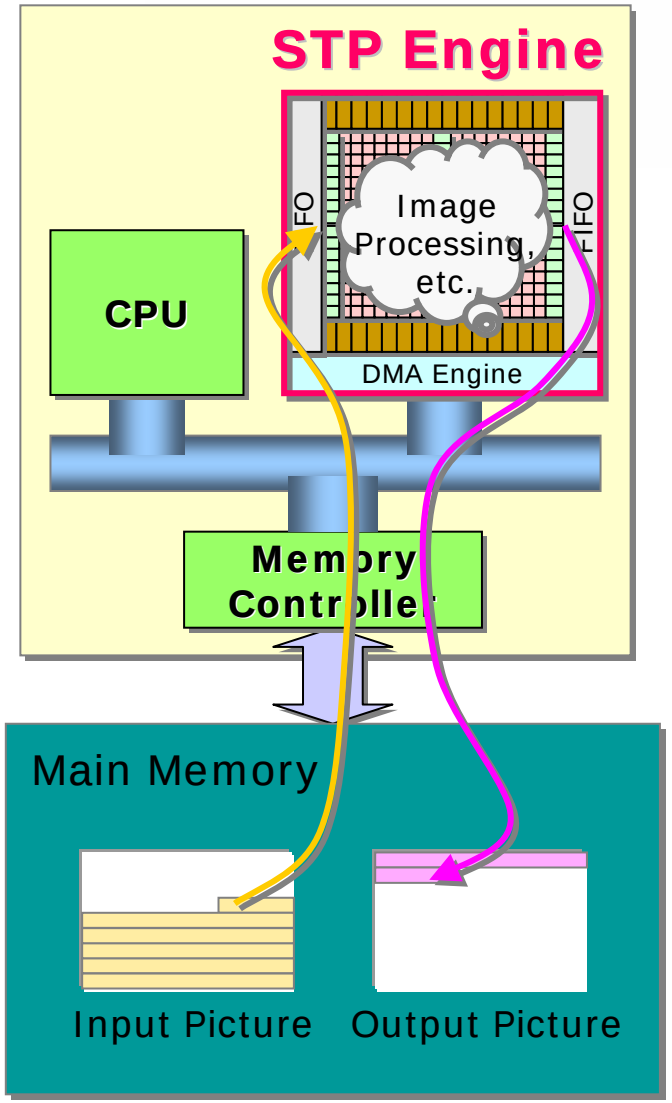
NEC

⇒ To solve the “SOC dilemma”
⇒ To fill the gap between HW and SW



- It is a programmable HW core integrated in an SoC
- It can load a set of pseudo HW configurations
- It reconfigures itself to one of those configurations during runtime
- Programming tool is C-based
  - C is the only common language for HW and SW engineers

STP Engine: Concept



DRP Array

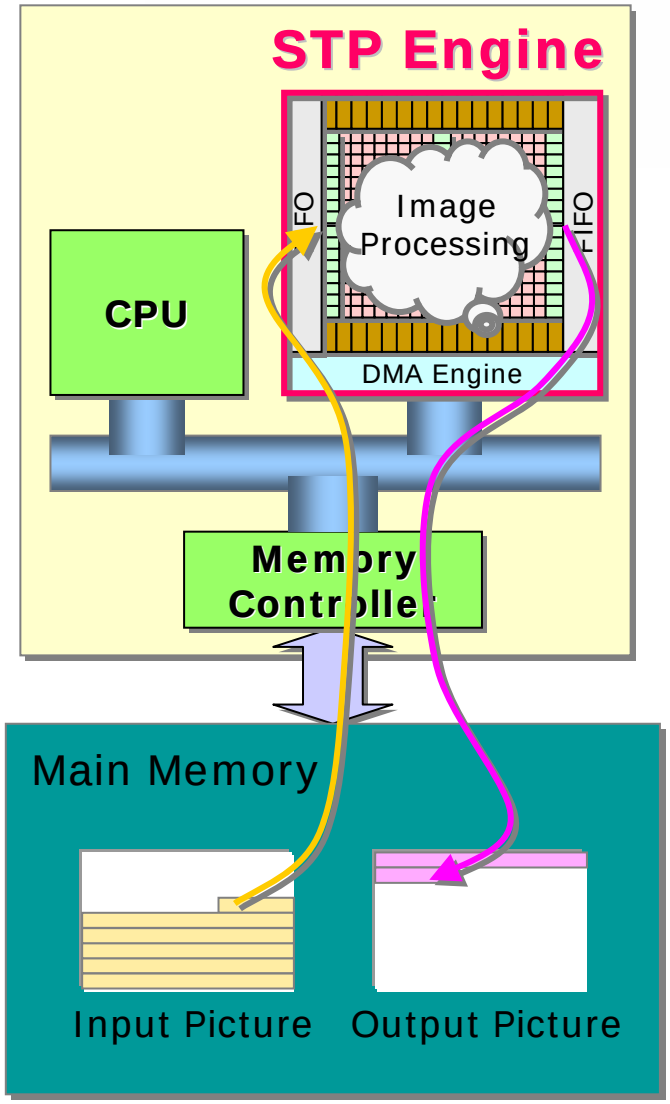
- Numerous numbers of processing and memory elements
- Tasks are mapped onto the array both in space and time domains
- Parallel and customized processing leads to high performance

Tightly coupled to hide mutual latency

DMA Engine

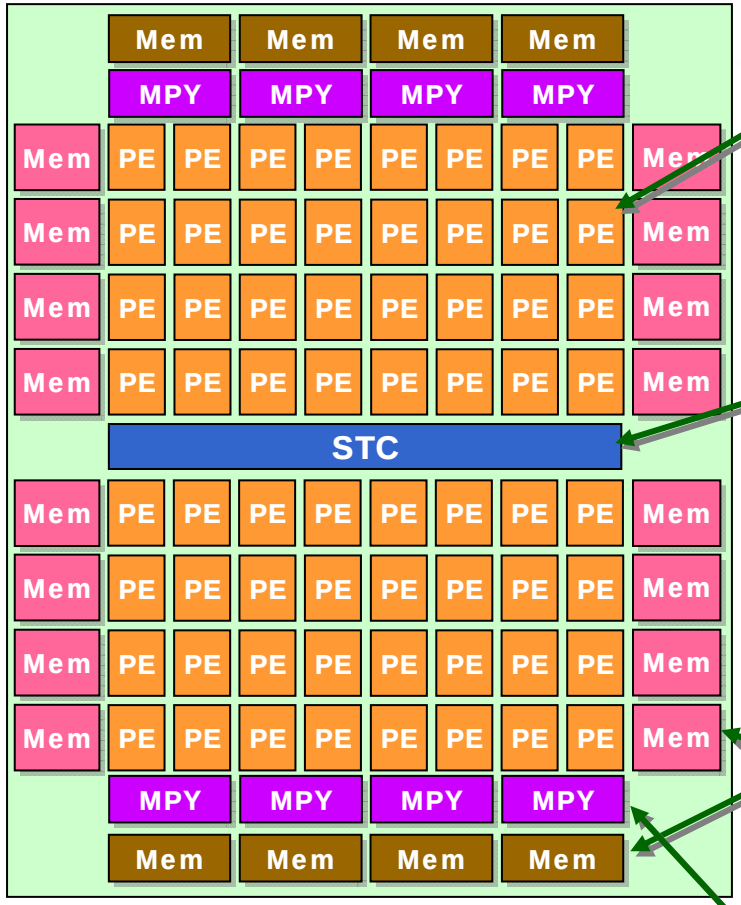
- Autonomous mechanism to stream input/output data
- Under the control of both CPU and DRP array through a set of API
  - Start, halt, re-load, read, etc.

STP Engine: Reasoning



- STP Engine := Stream Transpose Engine
  - An Engine that can process, convert, manipulate, etc. stream data, e.g., still image/video data and NW packets
- DRP Array := Dynamically Reconfigurable Processor Array
  - Base architecture was presented in MPF 2002 and SOC 2003
- DMA Engine plays a key role
  - SW tasks that require HW assists tend to be stream-oriented
  - It makes DRP array programming a lot easier
  - It provides a familiar look of "intelligent DMA" to CPU

DRP Array



Single Tile

Processing Element (PE)

- Byte-oriented ALUs
- Byte-width vertical/horizontal buses and registers
- Several tens of configuration sets

State Transition Controller (STC)

- Controls “dynamic reconfiguration”

Data Memory (Mem)

- Dual port
- Single port

16b Multiplier (MPY)

Execution Model (1): Spatial Mapping

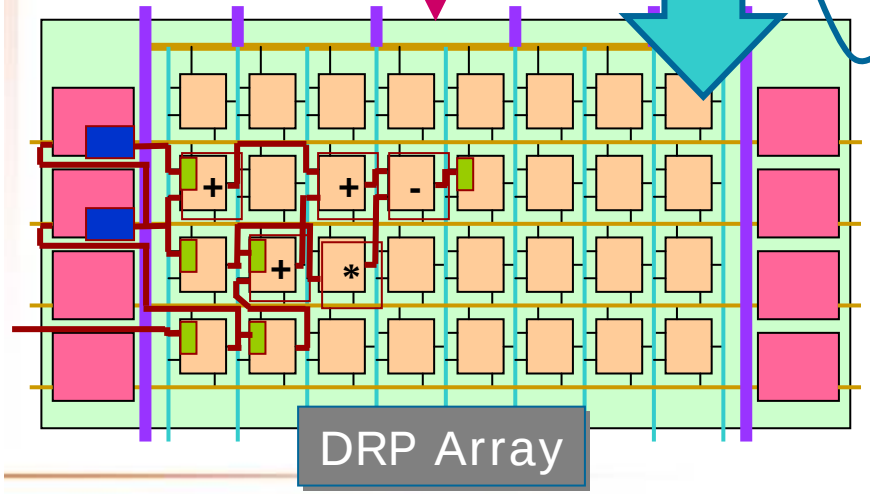
NEC

Example: 3x3 Filter

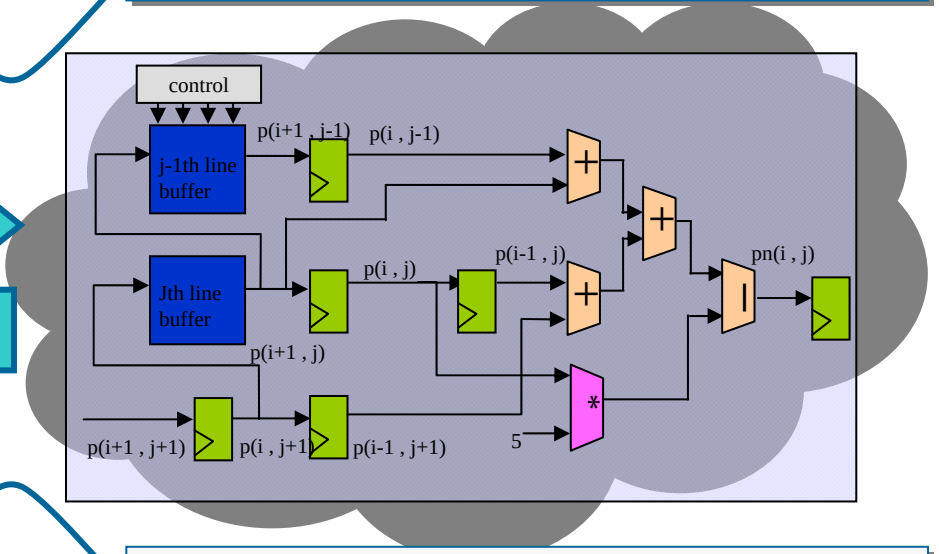
```
for( i = 0; i < N; i++ ){
    for( j = 0; j < N; j++){
        fn(i, j) = 5*f(i, j) – f(i, j-1) – f(i-1, j)
                   – f(i+1, j) – f(i, j+1);
    }
}
```

Source Code in C-langage

STP Tool

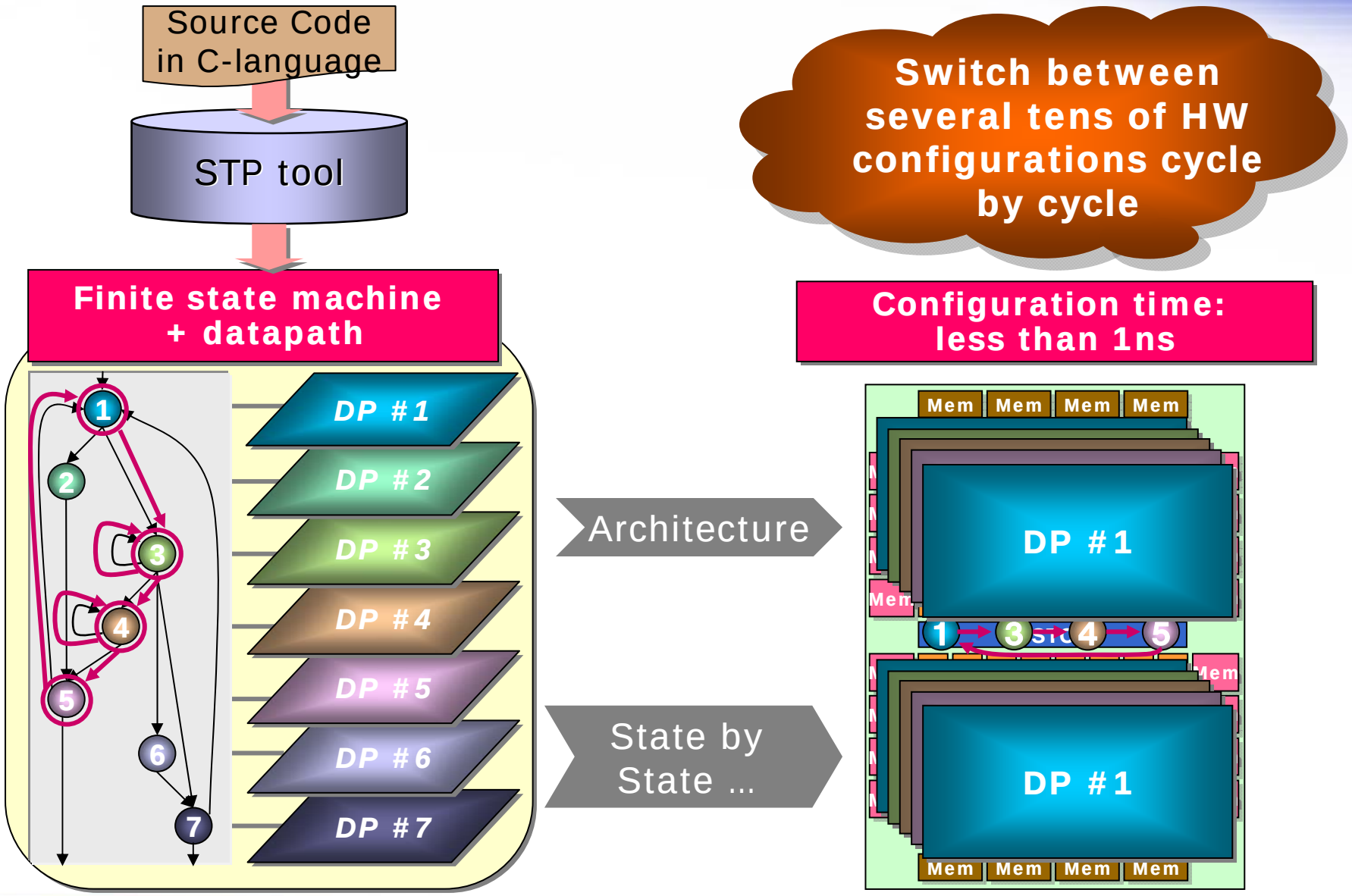


1. Generates a HW config. from the source code

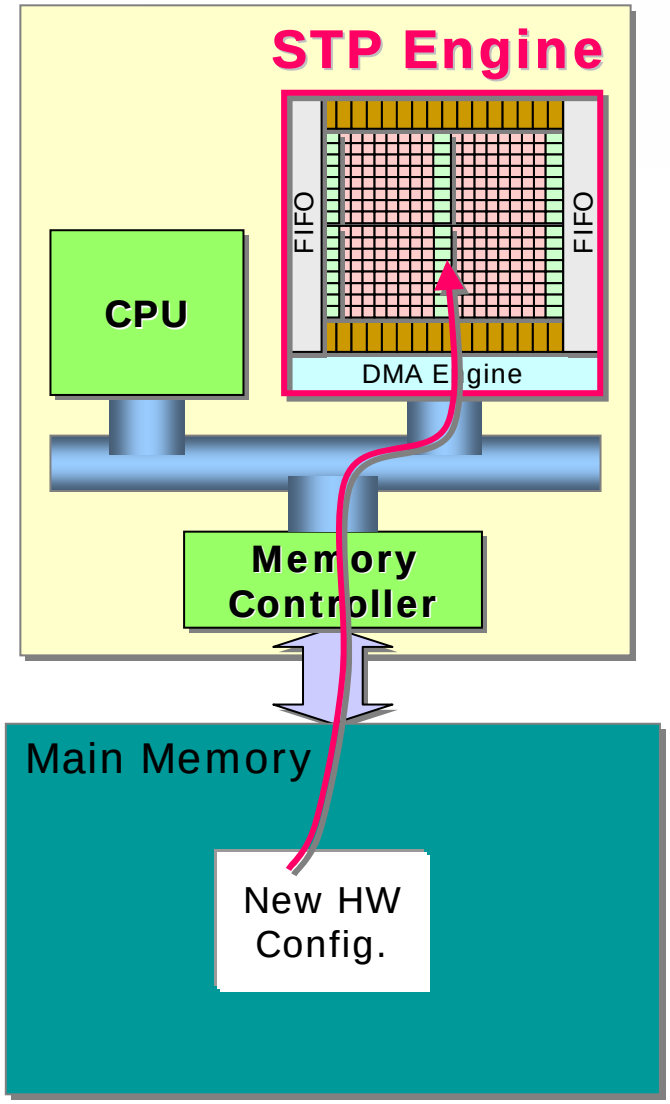


2. Spatially maps onto the array

Execution Model (2): Dynamic Reconfiguration



Execution Model (3): Dynamic Loading



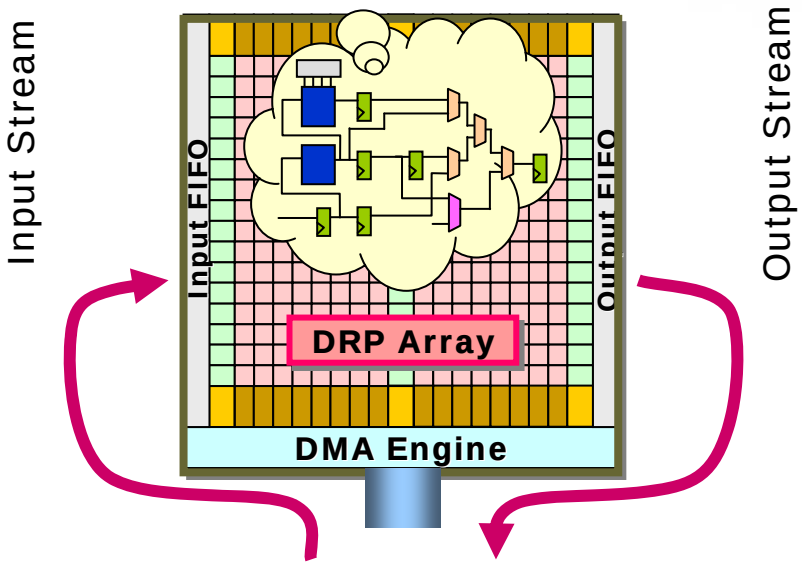
- Reload HW configurations from external memory
  - Can change into totally different HW
  - Can timeslice a huge application into a chunk of configuration sets
  - Can virtualize HW, in other words
  - Etc.

Configuration Time: Order of 100us

Application Domain

NEC

STP Engine



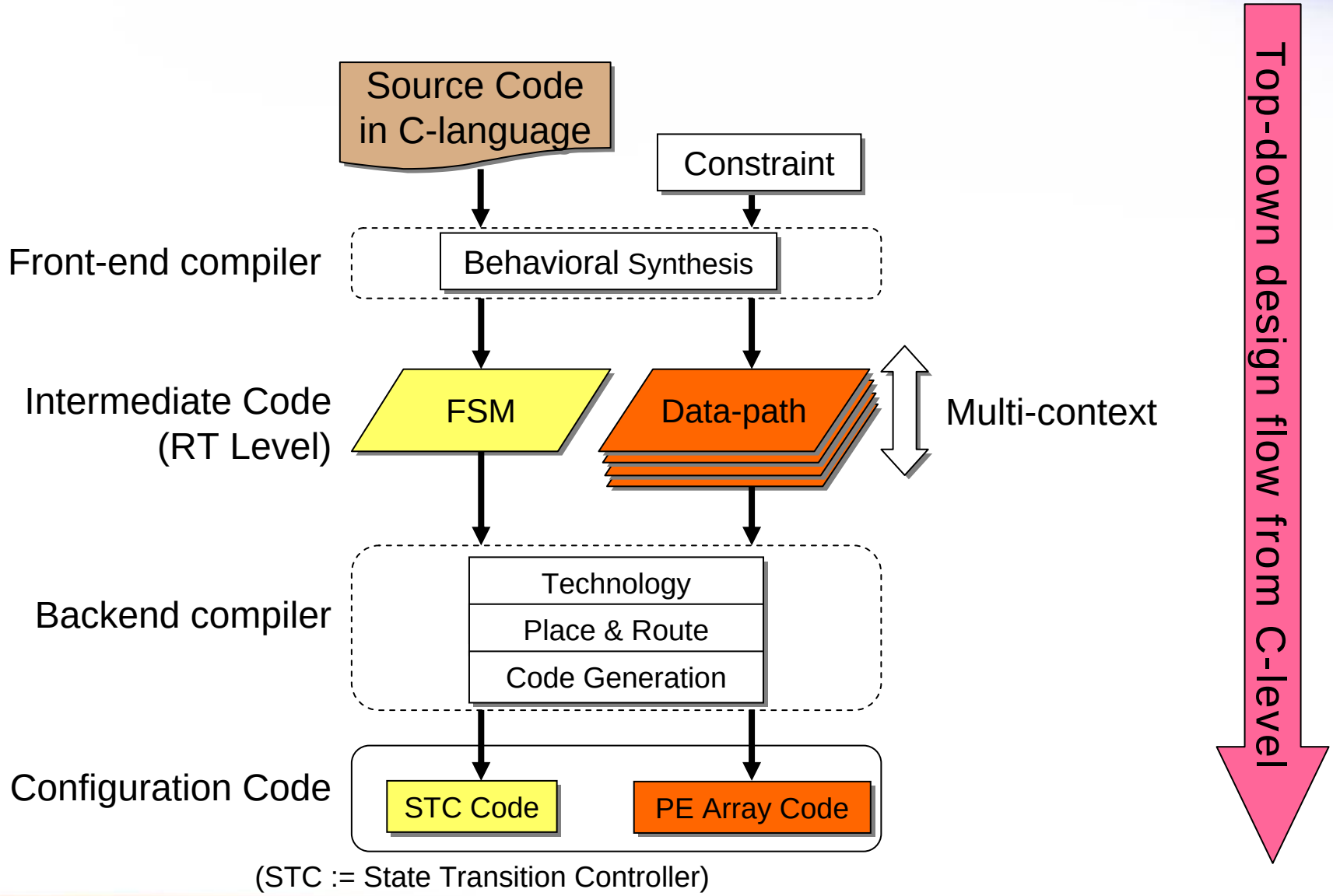
- Parallel Processing
  - Data parallel, pipeline, task parallel, etc
- Customized Processing
  - Datapath customized for a given task under area-performance tradeoff

Major Application Categories

- Still image processing
  - Image filter, transformation
  - Image scaling
  - Compression/decompression
  - Image recognition
- Video processing
  - Video stream multiplexing
  - Compression/decompression
- Packet processing
  - Forwarding
  - Access control
  - Contents search
- Others
  - Entropy coding
  - Encryption

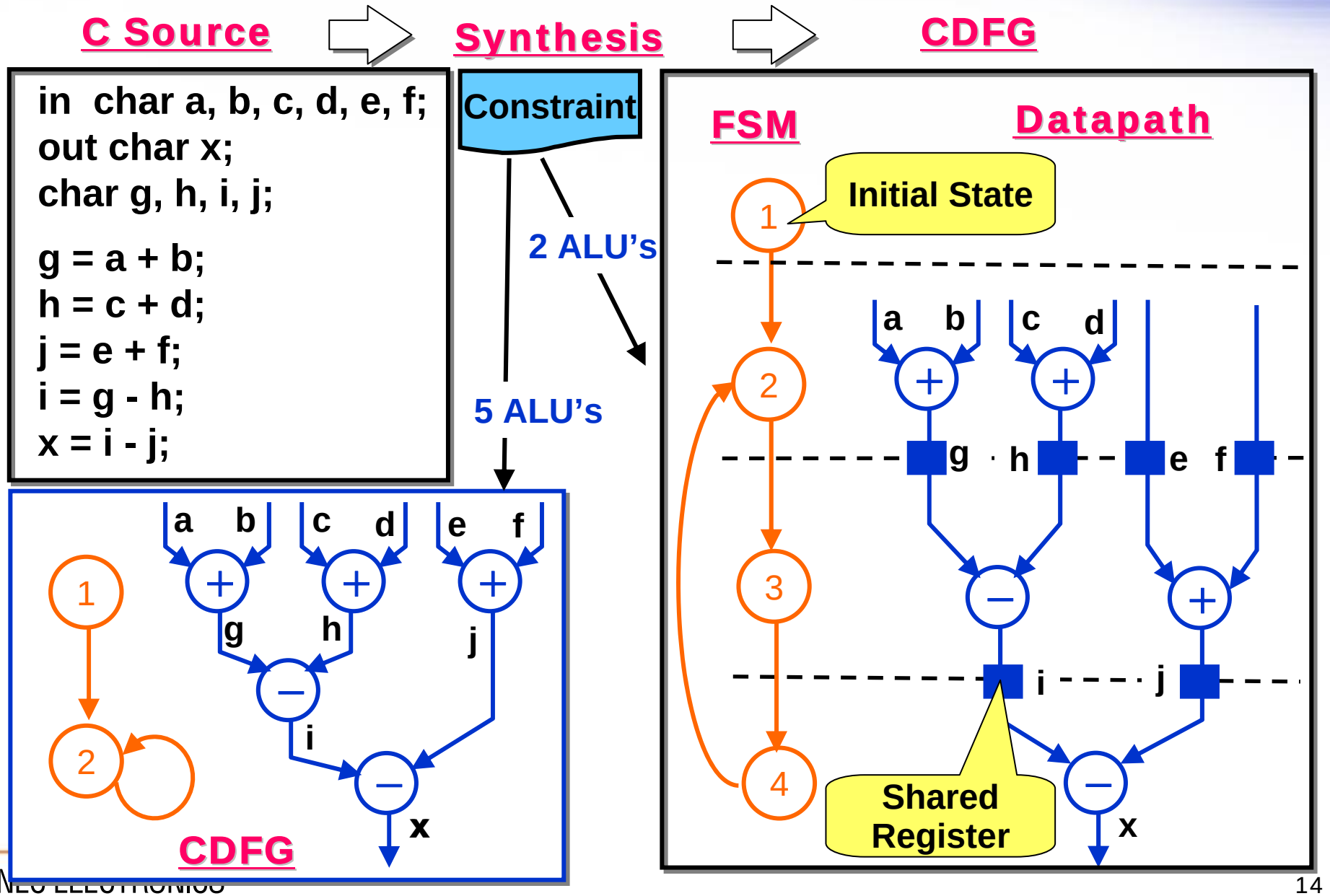
It works well when a given application is rather complex, and has mixture of different kinds of parallelism

STP Tool – Based on Behavioral Synthesis Tech.

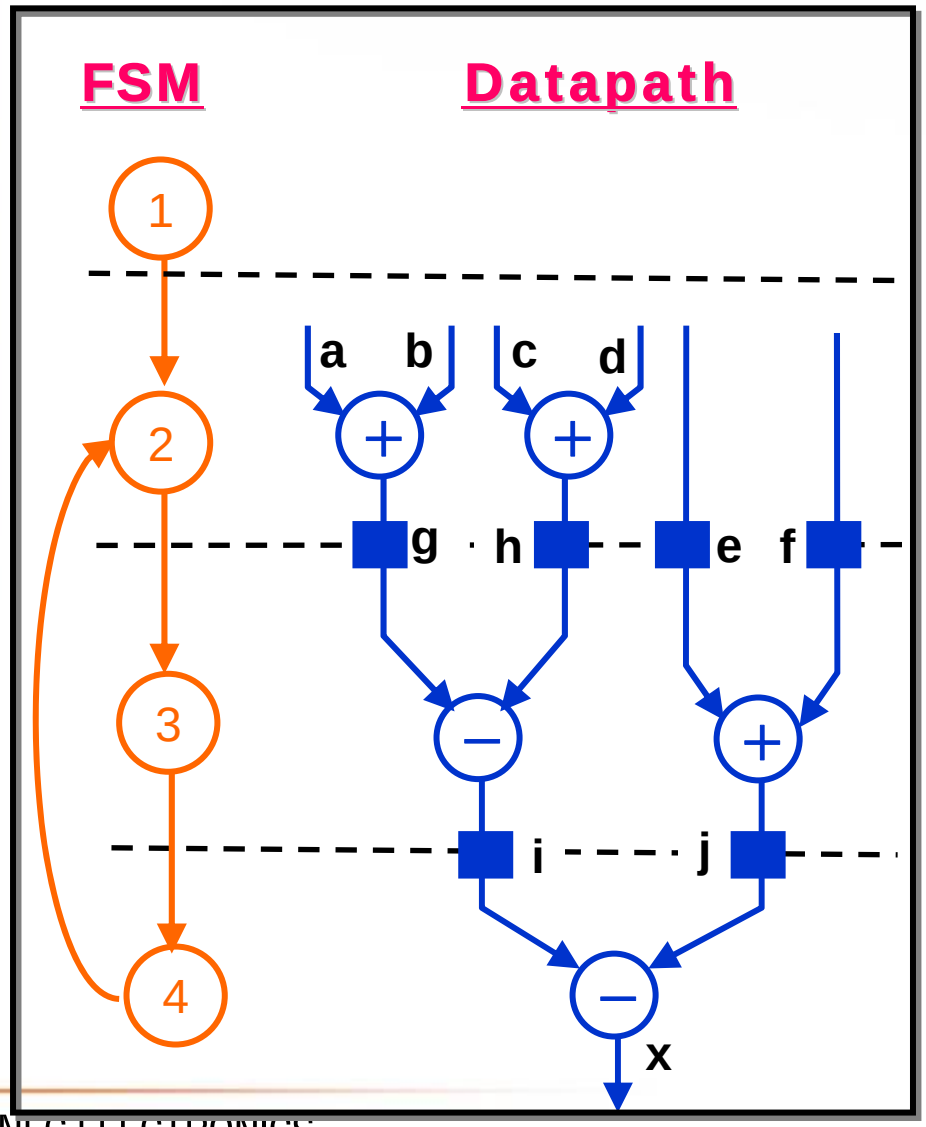


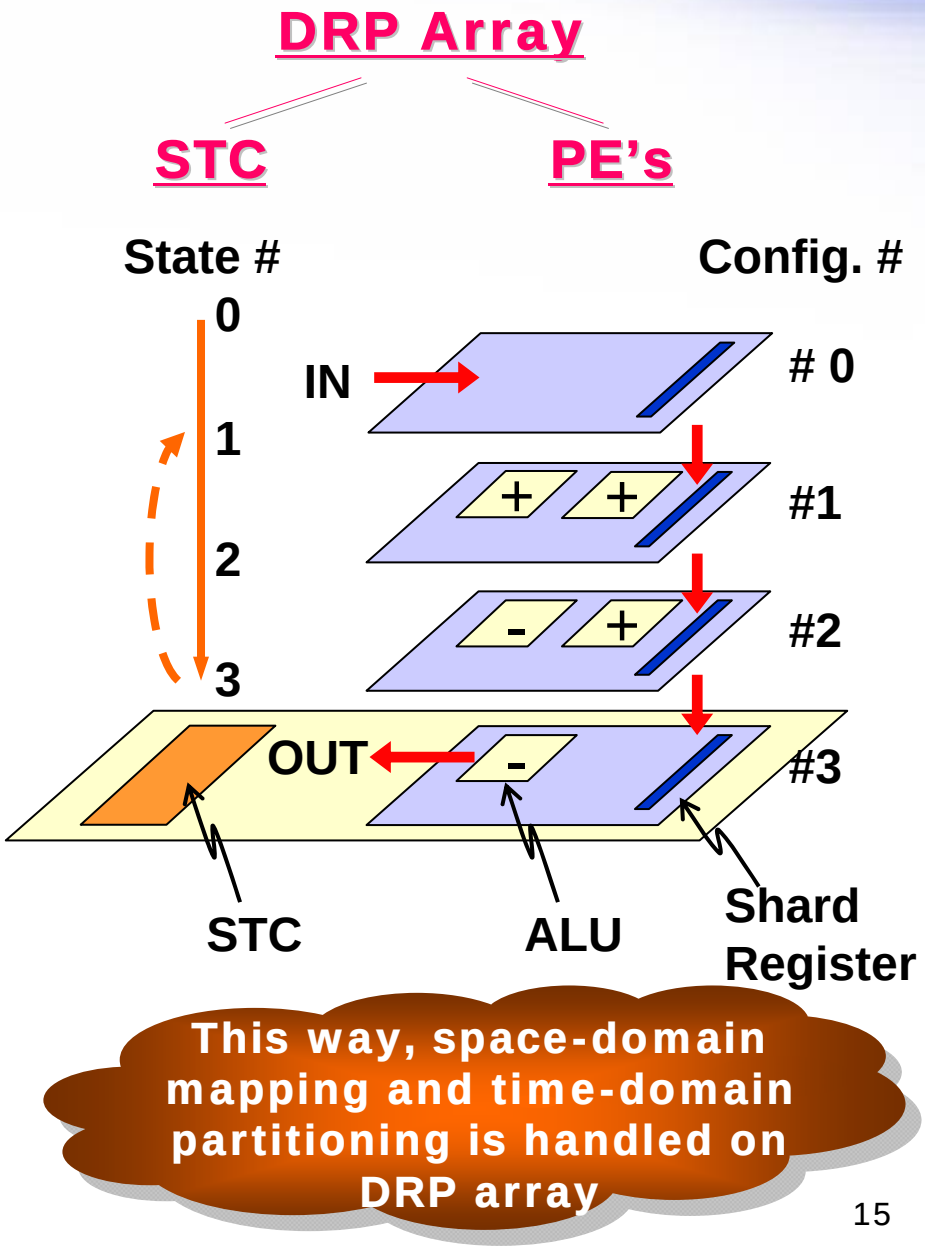
(STC := State Transition Controller)

Compilation Steps (1)

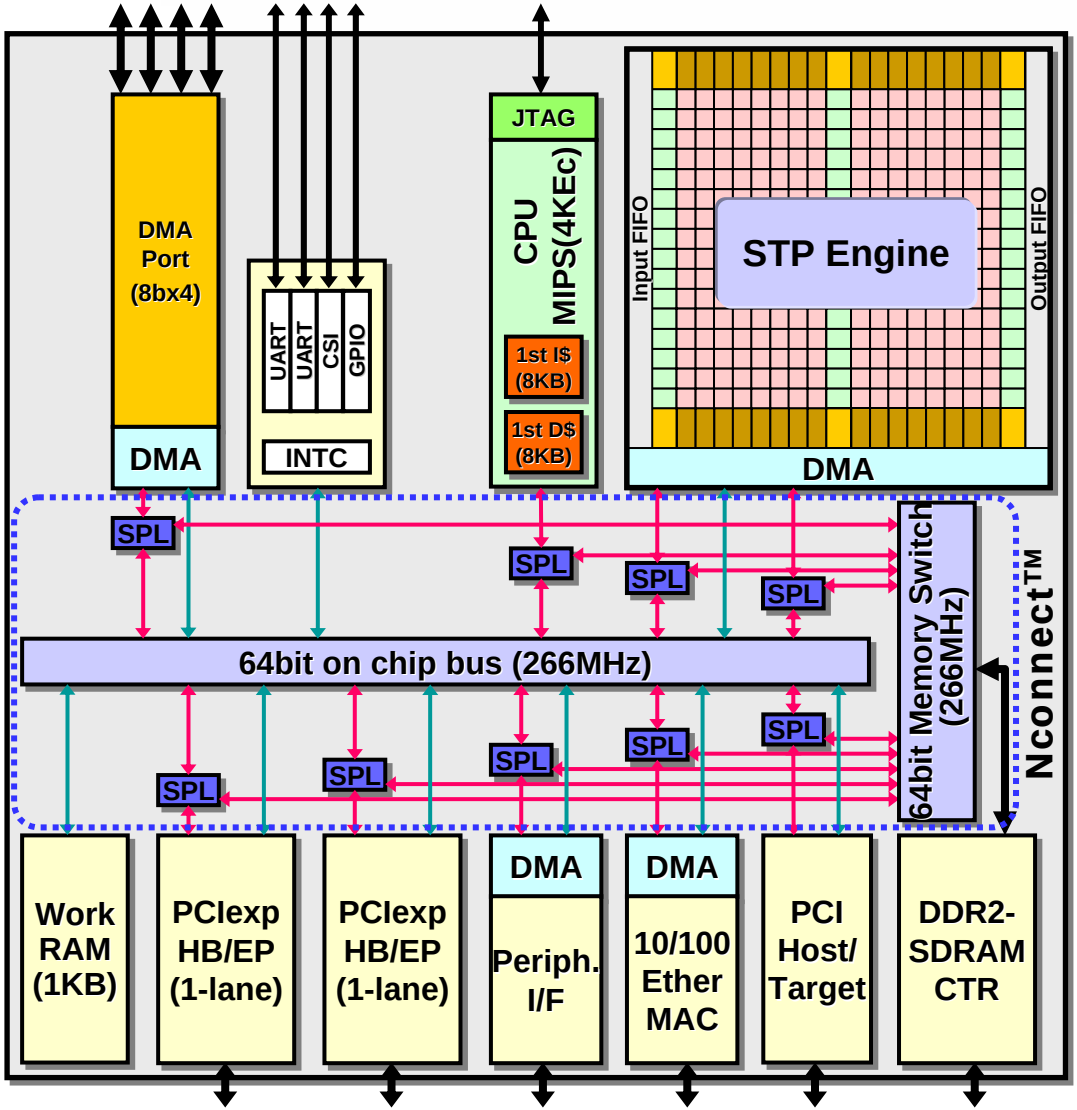


Compilation Steps (2)



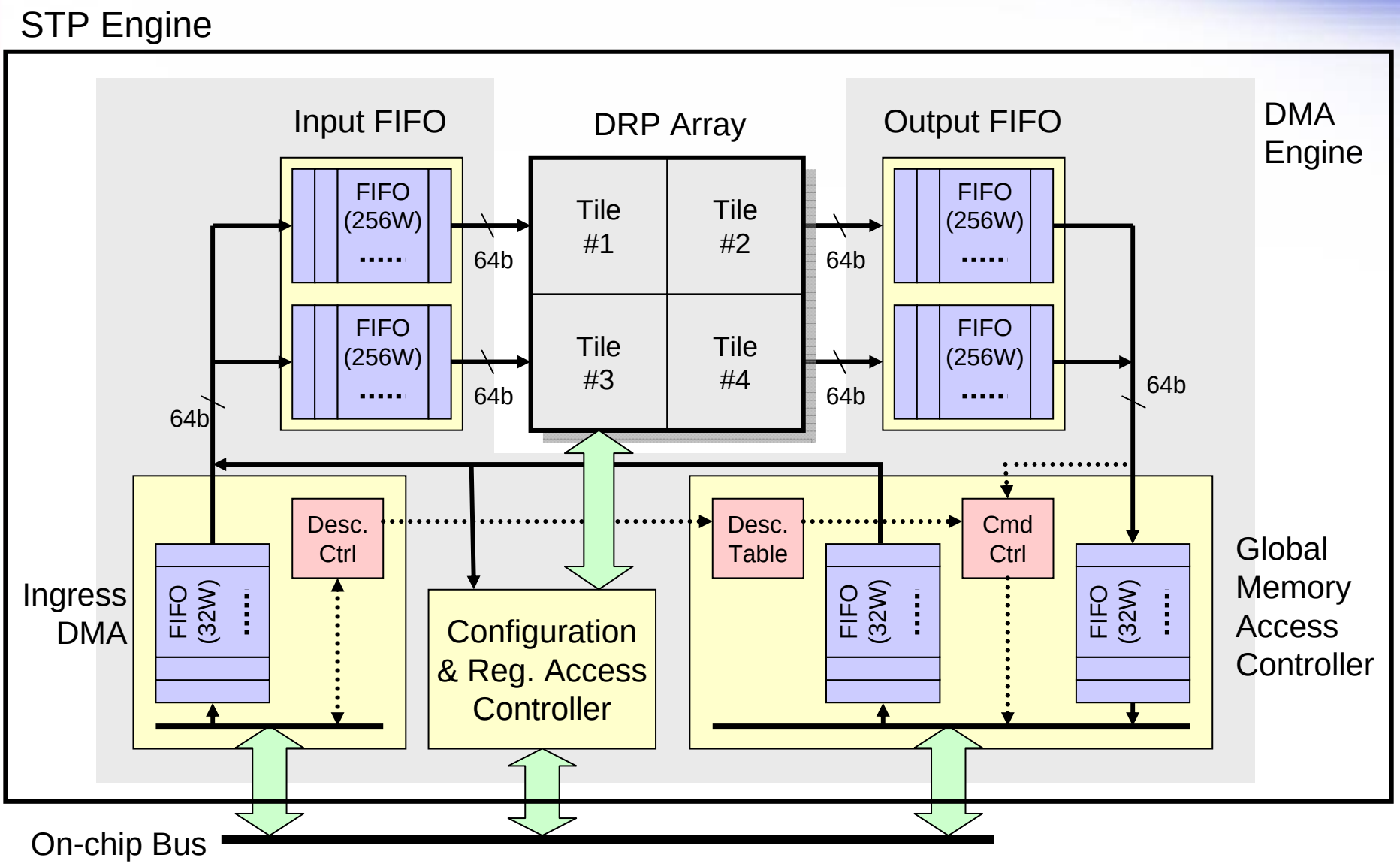


XBridge: System LSI with STP Engine

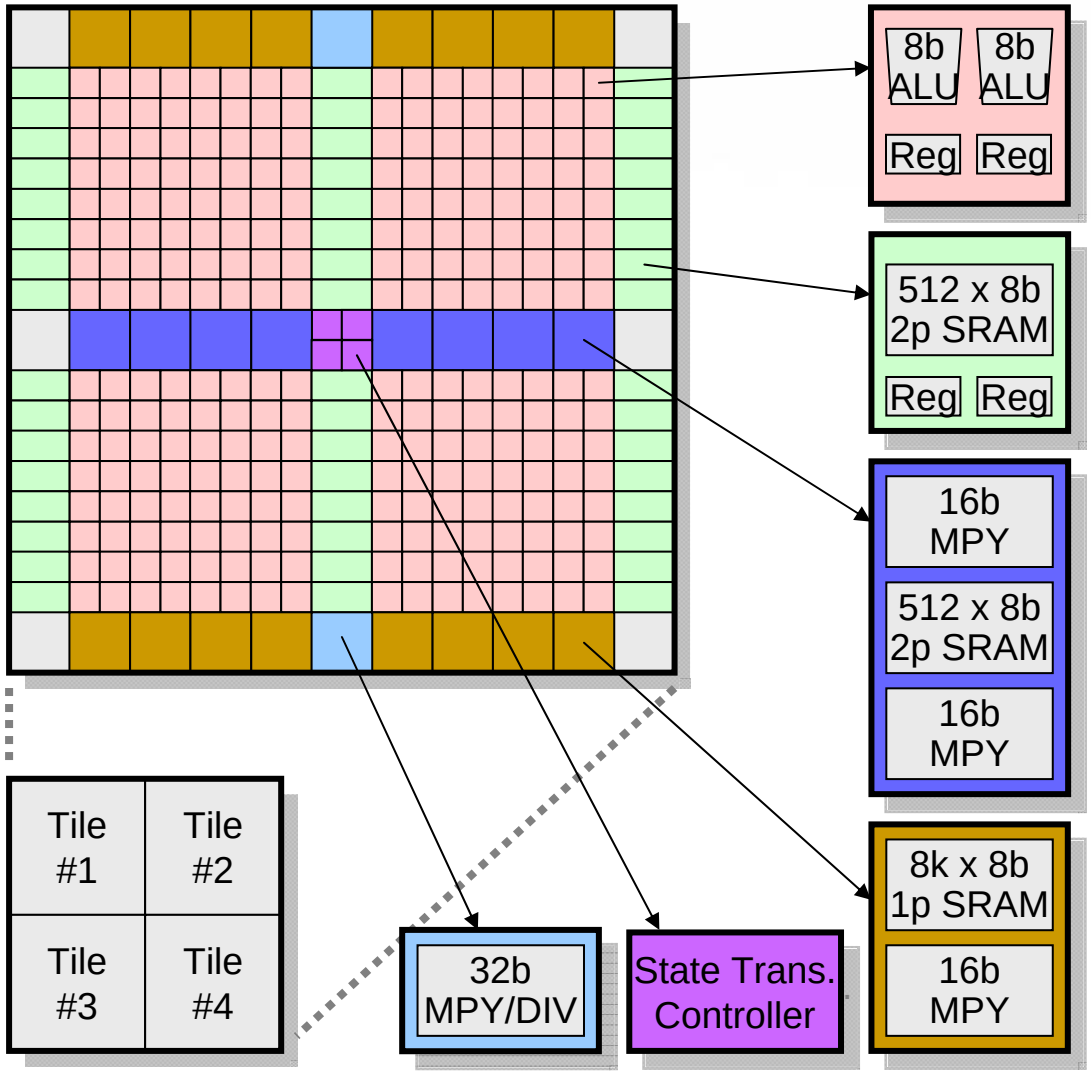


- 90nm process tech.
- 960pin 0.8mm pitch FCBGA (27mm)
- Operation Clock:
  - System: 266MHz
  - STP Engine: 33-100MHz
- Total power: 2W(Worst)

STP Engine: Detailed Block Diagram



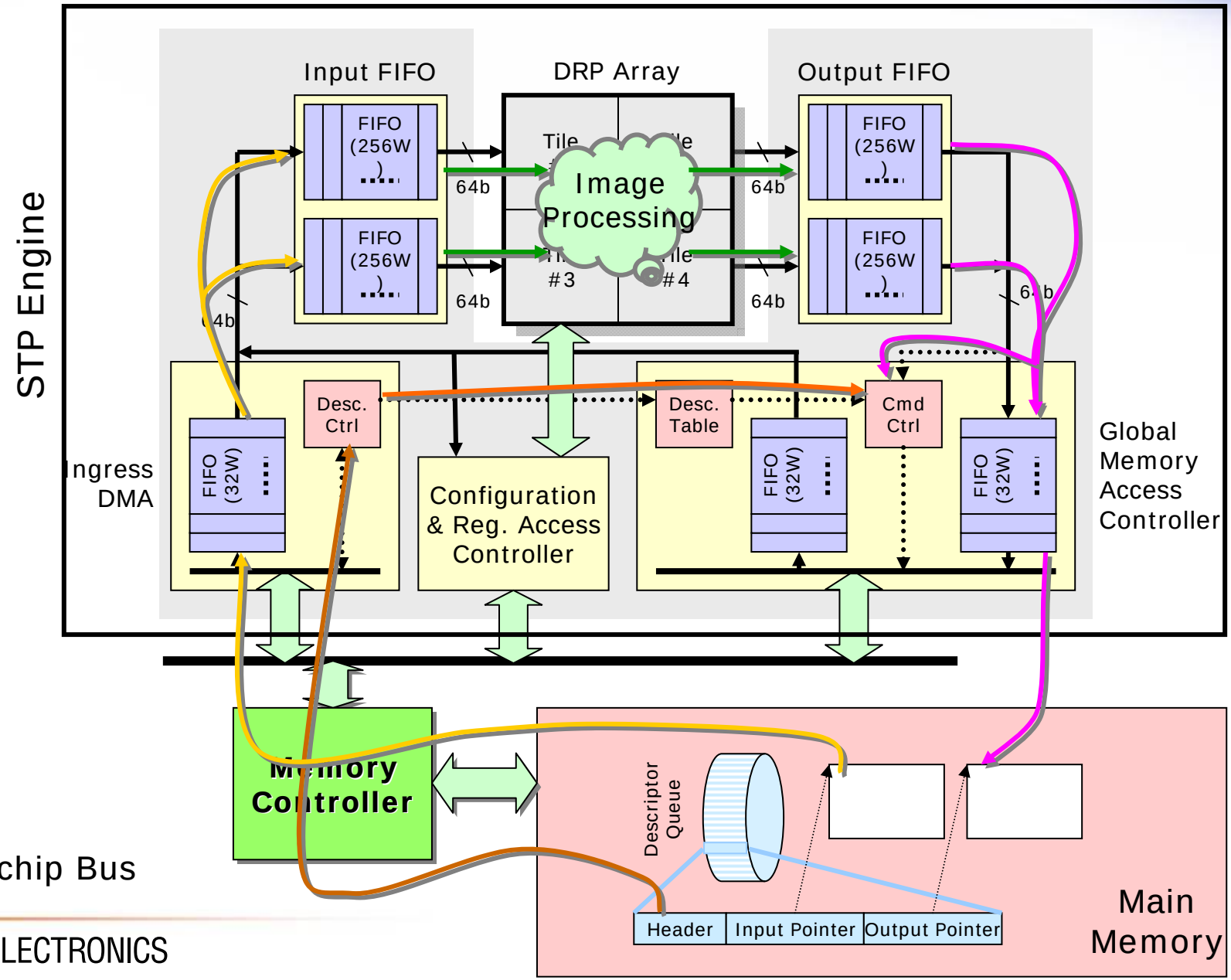
DRP Array



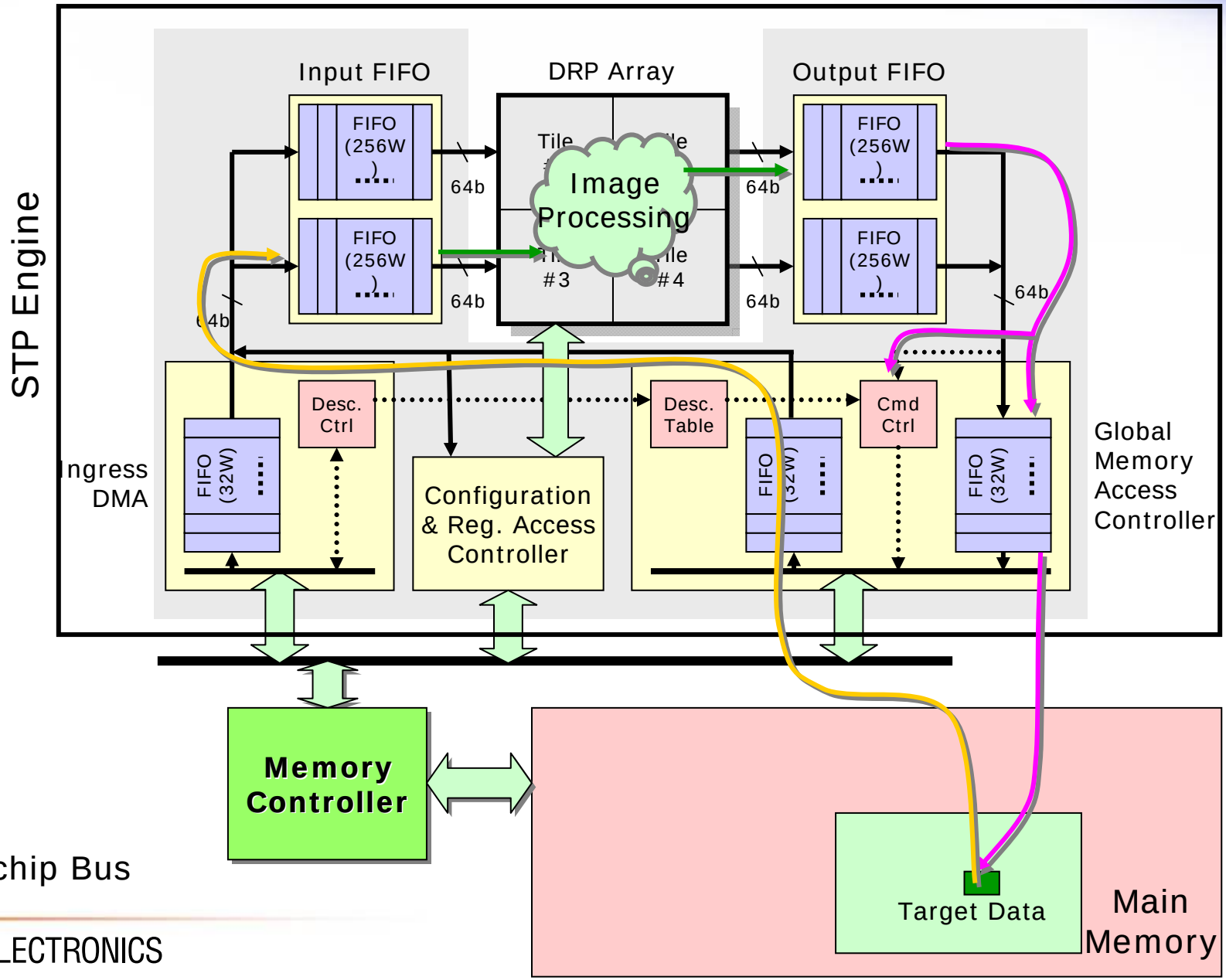
RESOURCES

- PE x 256
  - @8b ALUx2
  - @8b Regx2
- 16b MPY x 32
- 2p SRAM x 56
  - @512B
  - Total 28KB
- 1p SRAM x 16
  - @8KB
  - Total 128KB
- Contexts x 32

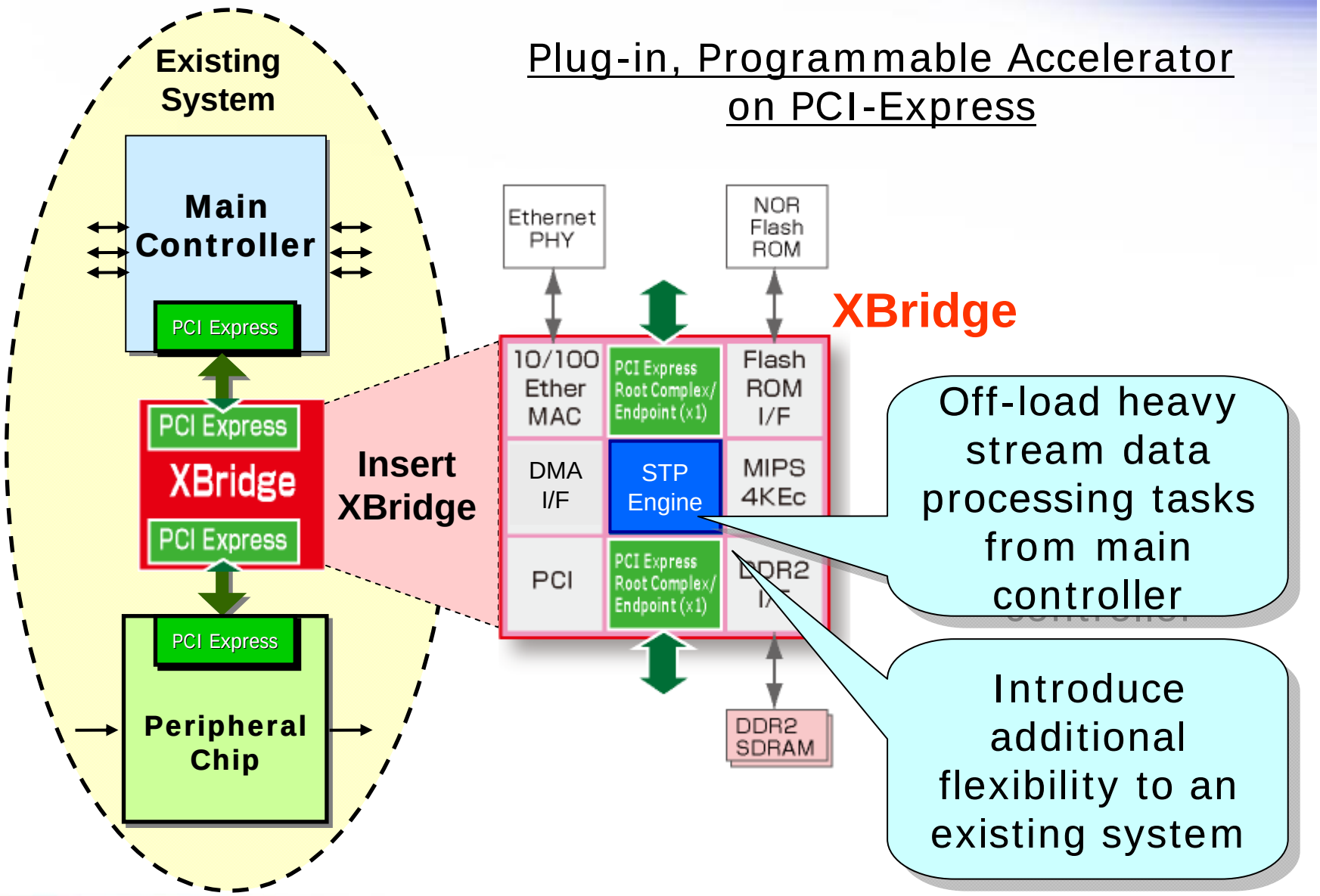
Ingress/Egress Data Streaming



External Memory Access



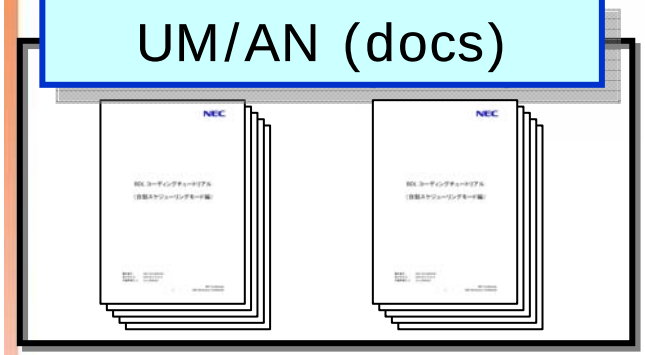
XBridge: System Application



XBridge Evaluation Kit

NEC

UM/AN (docs)



STP Tool (IDE)

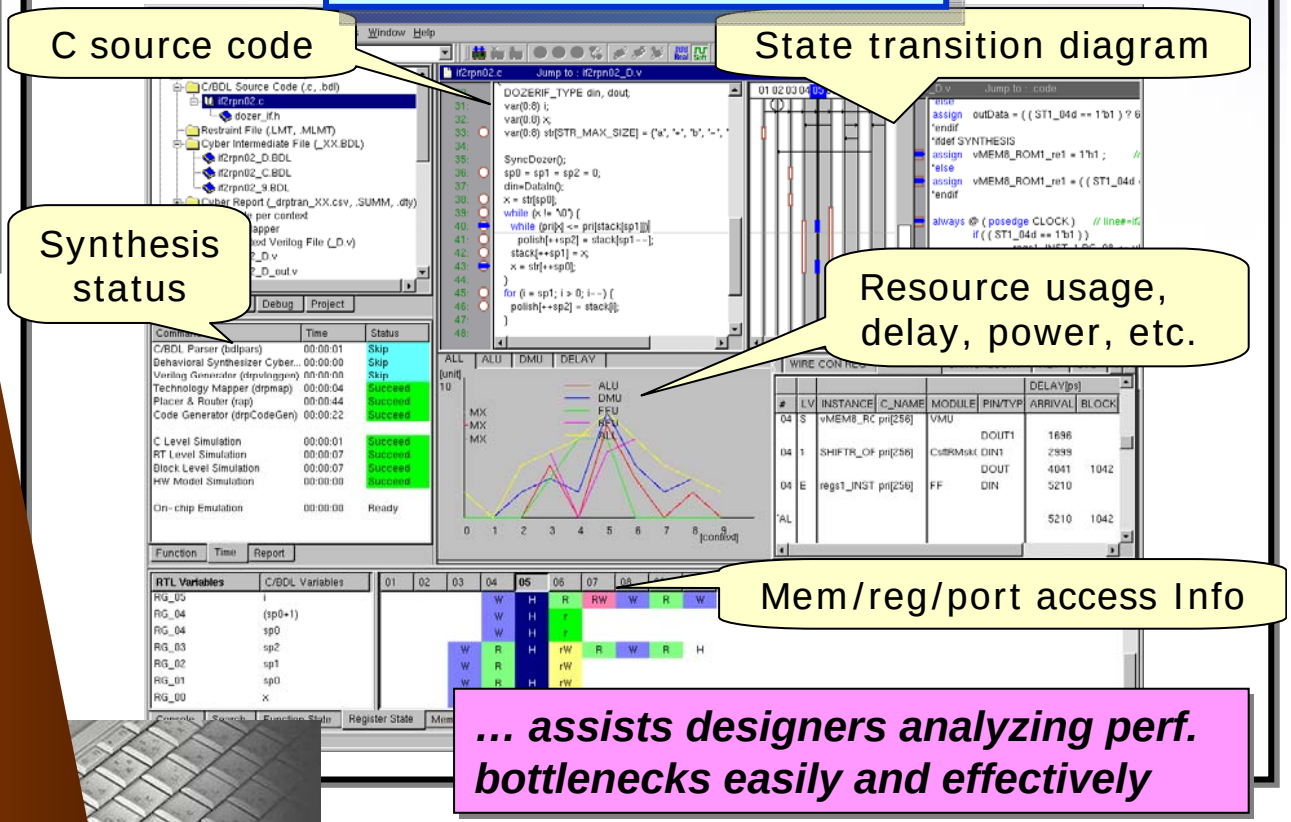
C source code

State transition diagram

Synthesis status

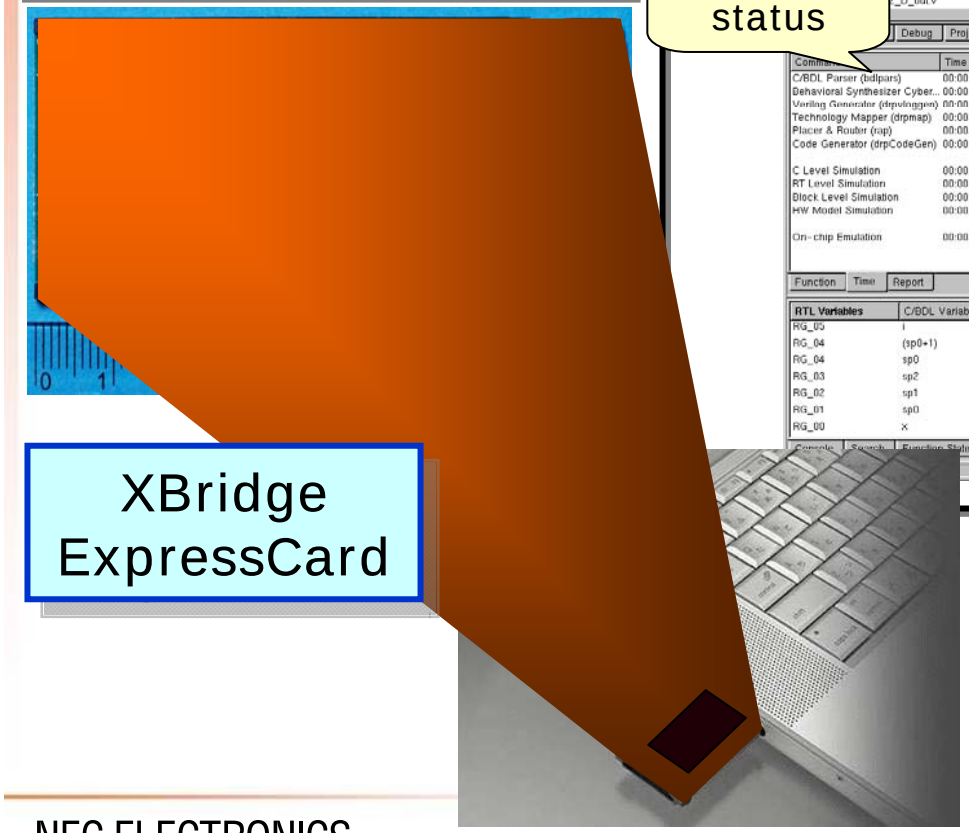
Resource usage, delay, power, etc.

Mem/reg/port access Info



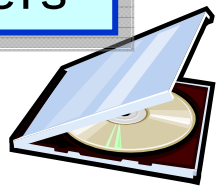
… assists designers analyzing perf. bottlenecks easily and effectively

XBridge ExpressCard



Drivers

- Windows
- Linux



Demonstration

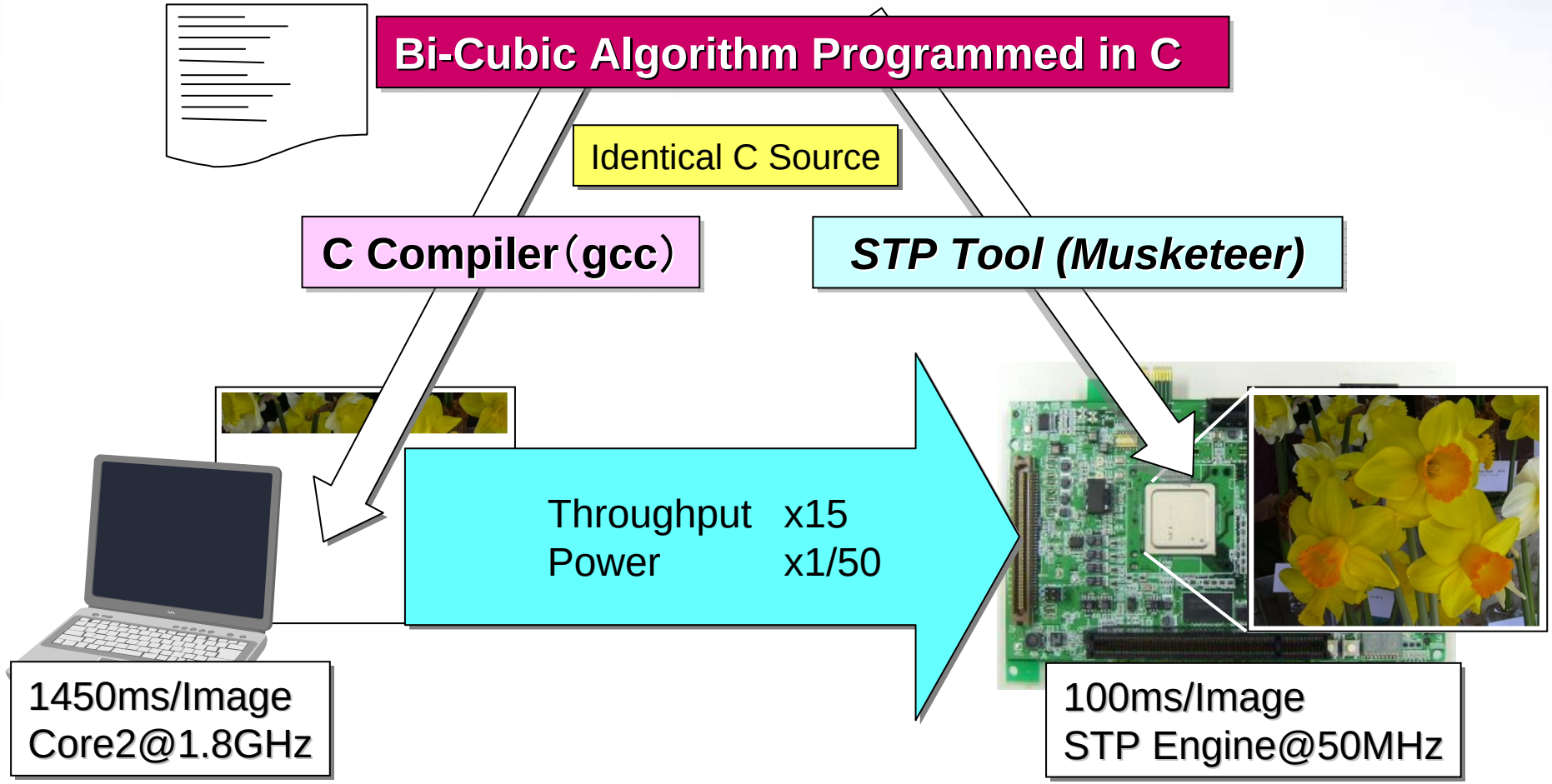
Image Scaling
(STP vs. CPU)

Error diffusion
(STP vs. CPU)

Superfine Resolution
(STP vs. CPU)

Noise Reduction
(STP vs. CPU)

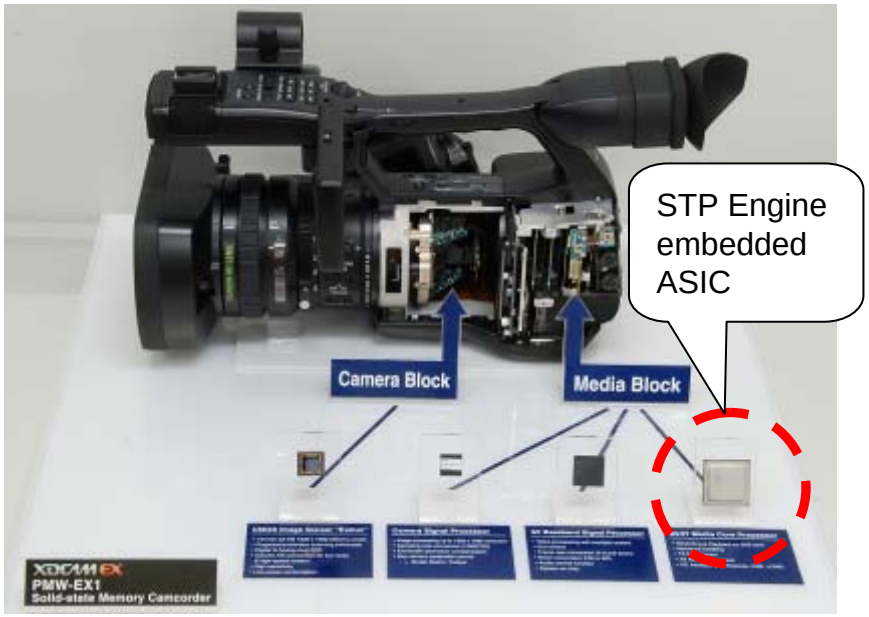
Performance Comparison



STP Engine Used in Real Products

- Integrated in our customer's ASIC for professional cam coder
  - STP Engine (90nm) is embedded as an generic accelerator of the CPU
  - Implemented functions:
    - Stream Packet Mux/DeMux
    - Audio Encode
    - Intelligent DMA
    - **Video Codec, etc**

Extended to the enhanced models by updating functions running on STP Engine



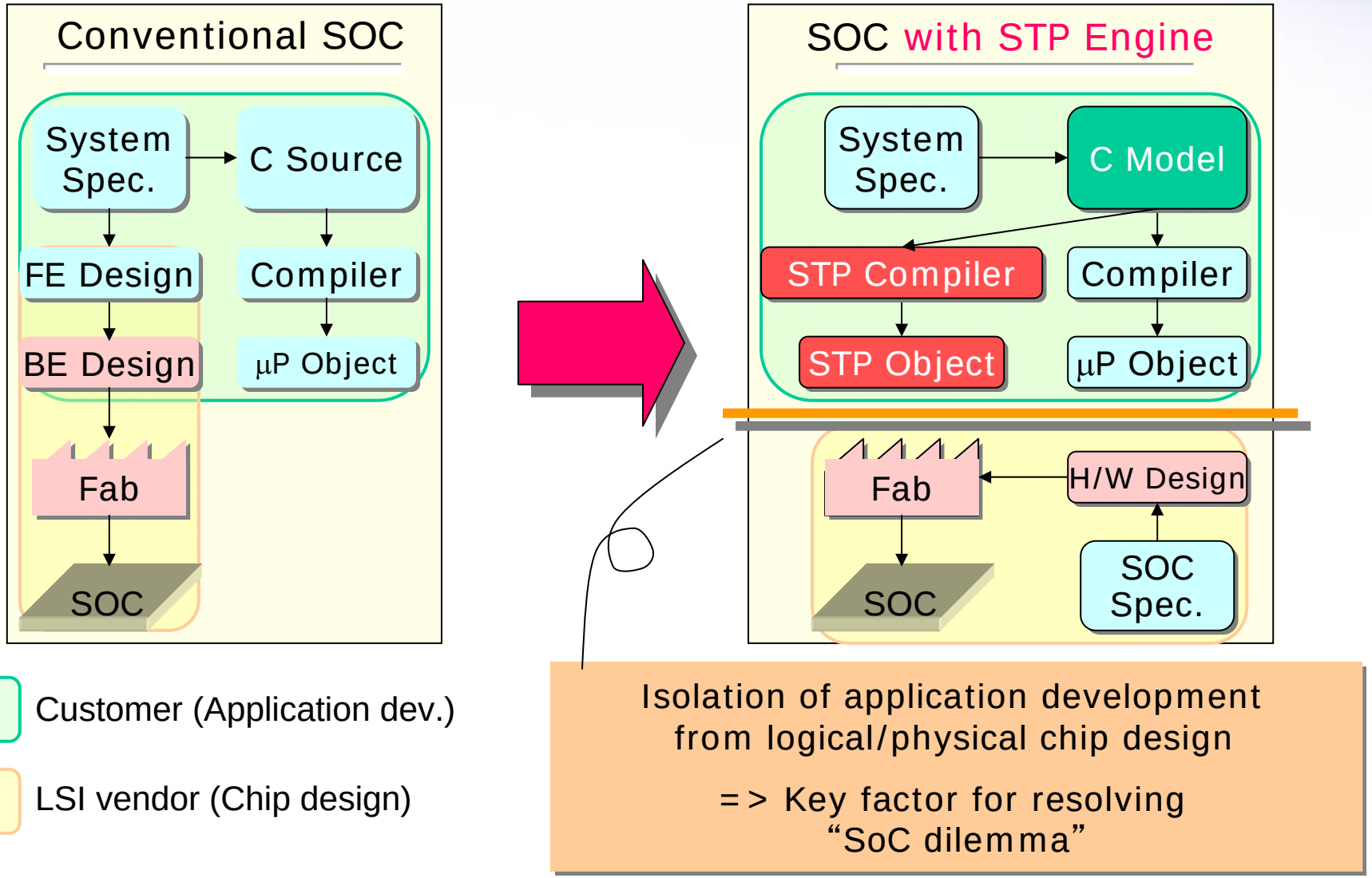
Nov. 2007

July 2008

July 2008

(Courtesy Sony Corporation)

Paradigm Shift in SoC Design



STP Engine as a “Many Core”

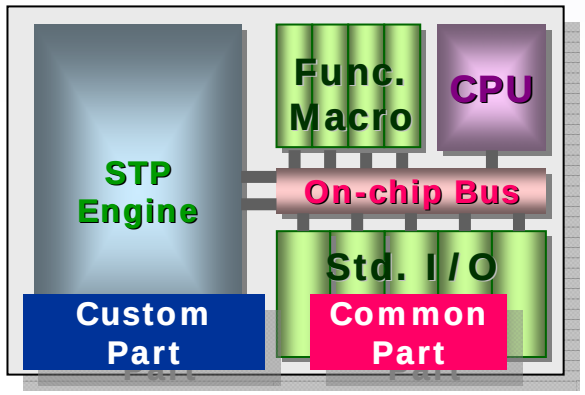
- Conventional many-core programming issues
  - Parallel programming is hard
  - Automatic parallelizing compiler is really hard
  - Rewriting, for different # of cores, is inevitable
- STP Engine challenges these issues by using the power of behavior synthesis technology
  - “Compilation into HW configuration” is the key differentiation
- When using STP Engine,
  - Parallel programming task becomes just as simple as writing a straight forward, single-process C program
  - The tool automatically parallelizes loops, branches, tasks, etc.
  - Rewriting, for different array size, is not necessary (only to put different constraint)

STP Engine: Roadmap

CY	06-07	08	09	10	11	12	13

Multi-market Application

Productized in high-end video camera



IP Core

90nm Generation → 40nm Generation → 32/28nm Generation → .....

- Cost/power reduction
- Throughput improvement
- Testing-friendly architecture

R&D

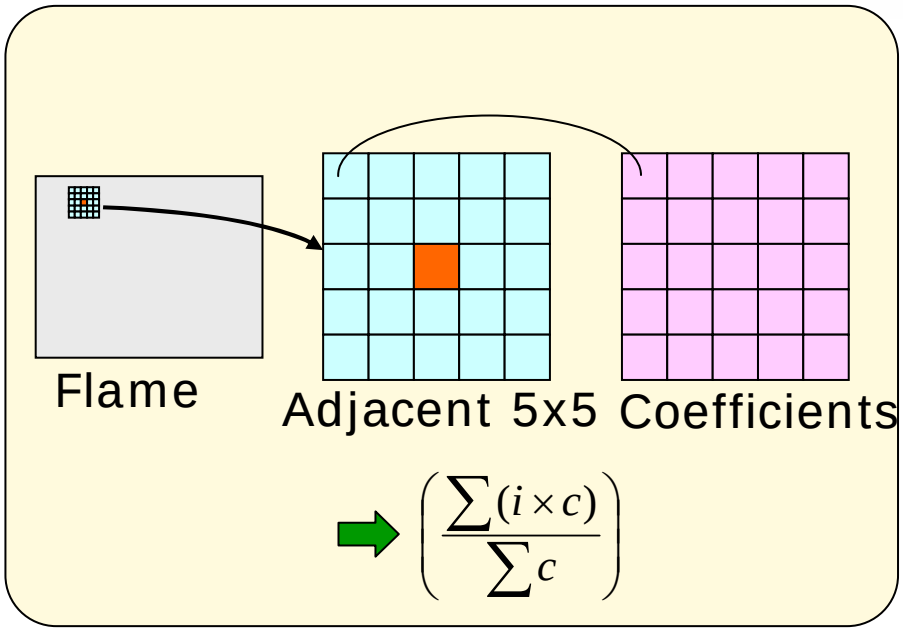
Architecture

Design Tool

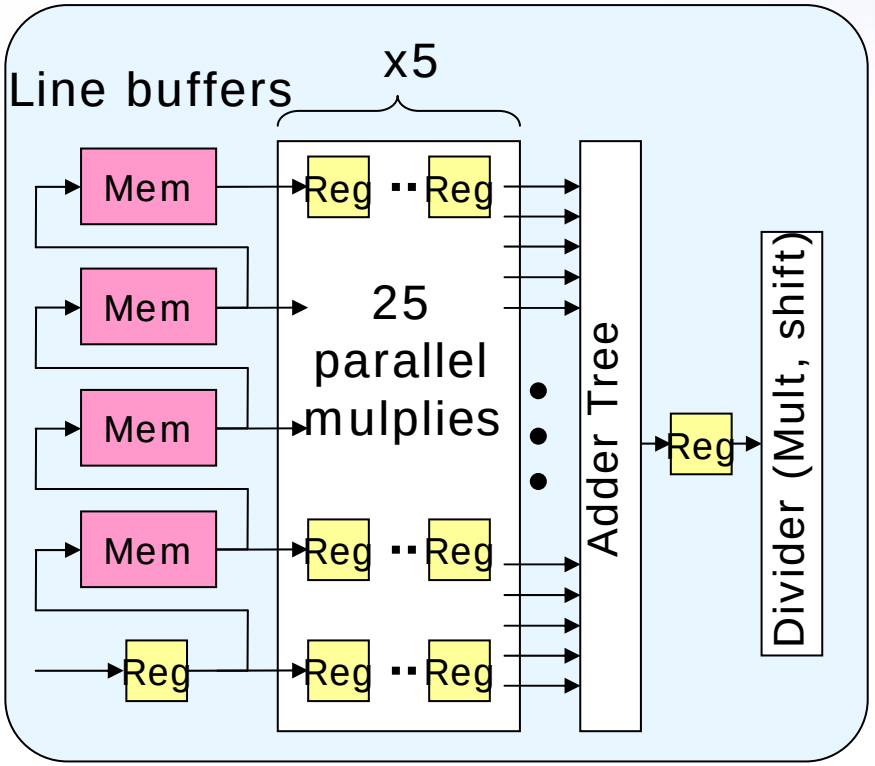
Spatial Mapping Example: Image Filter

NEC

5x5 Filter



Mapped onto DRP Array



Multiply: x25
Addition: x24
Division: x1

In one cycle

- Process 5 lines one time
- Parallel data access
- Pipelined processing
- 1 pixel per 1 cycle

参考データ: デモ画像サイズ

NEC

- Filter(9x9フィルタ+反転＋二値化)の画像情報
  - momiji.pgm　もみじ　1904x1520
  - murasaki.pgm　フリージア　640x 512
  - siro.pgm　アネモネ　768x 614
  - sakura.pgm　桜　717x 573
  - mejiro.pgm　メジロ　832x 666

- Bicubicの画像情報
  - momiji.ppm　もみじ　712x 566
  - sakura.ppm　桜　640x 512
  - siro.ppm　アネモネ　640x 512
  - tori.ppm　ユリカモメ　368x 288
  - mejiro.ppm　メジロ　640x 512
  - yakei.ppm　夜景　640x 512