Manycore Processor
for Video Mining Applications

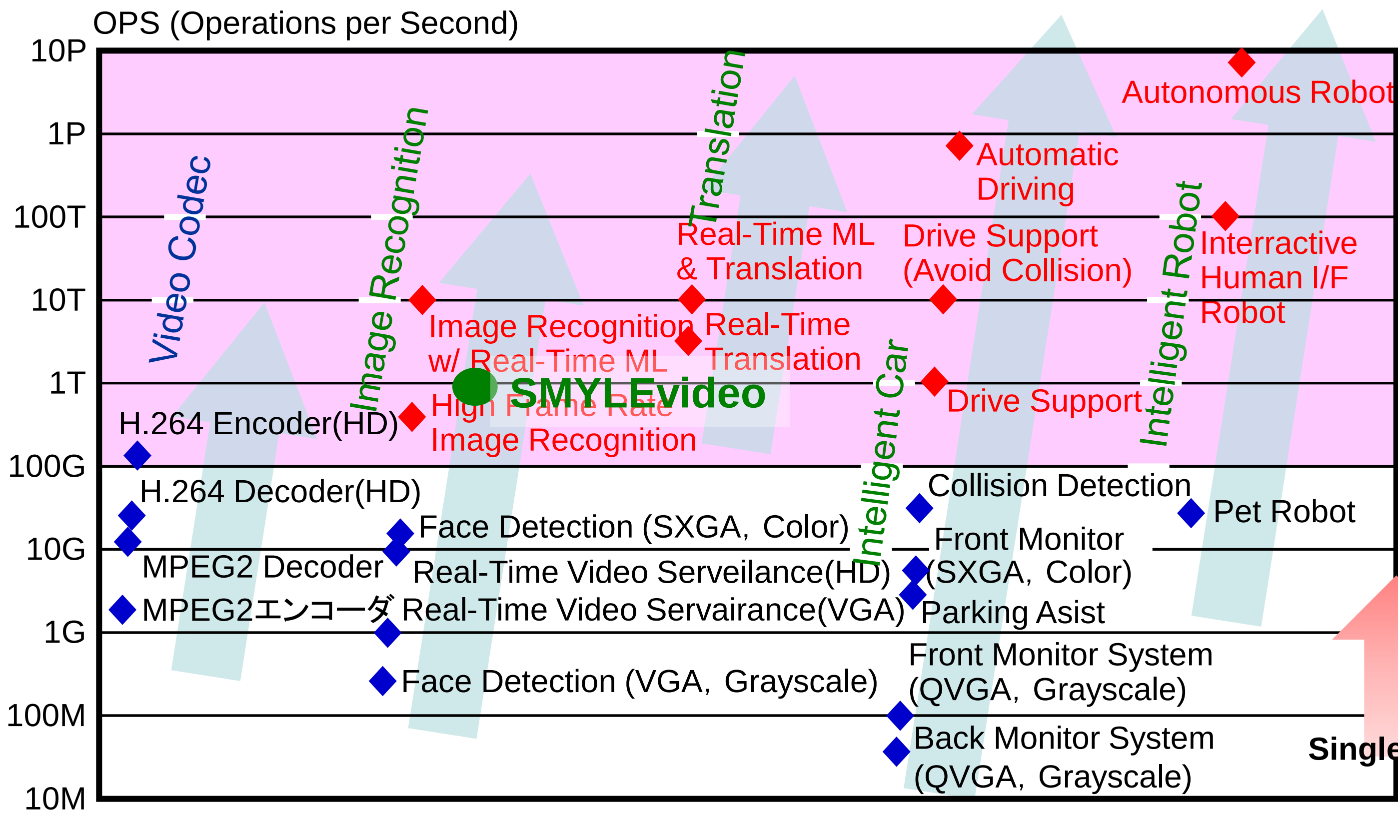
Jan. 25th 2013

Yukoh Matsumoto, Hiroyuki Uchida, Michiya Hagimoto,
Yasumori Hibi, Sunao Torii, Masamichi Izumida

TOPS Systems Corp.

More Energy-Efficiency required for Next-Generation Embedded Systems

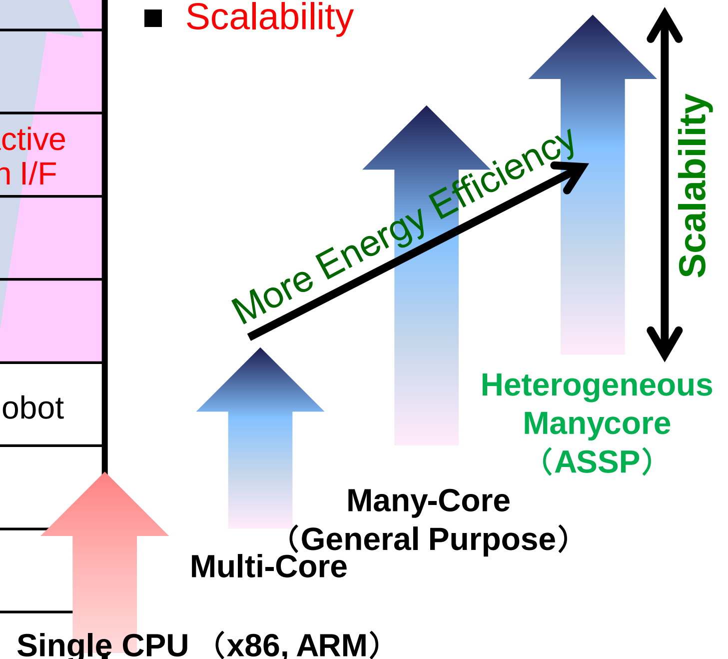
- **Next-Gen Embedded Systems :Requires more performance（100's GOPS~ 1 POPS）**
- **Power Consumption： Already reached upper limit（～ W）**



Ref： NDEO Technology Roadmap 2009, I-48p, Fig. 1-6

＜Expectations on Many-Core＞

- 10's GOPS/W ⇒ 100's GOPS/W
- General Purpose ⇒ ASSP
- Scalability

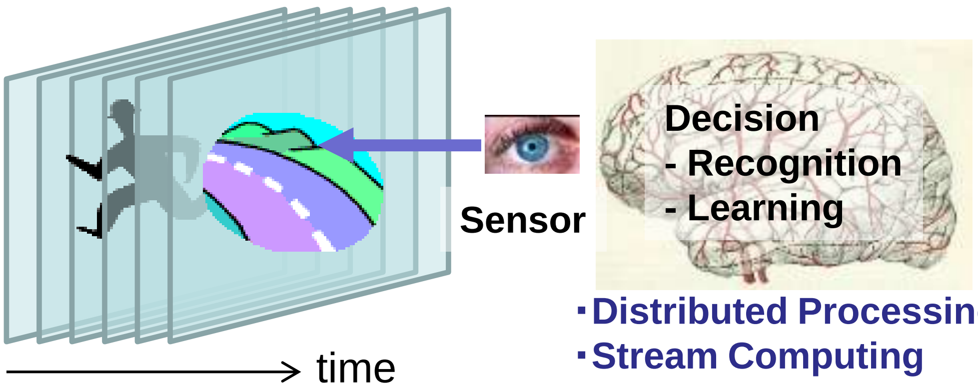


Energy-Efficient Computing goes to Heterogeneous & Manycore

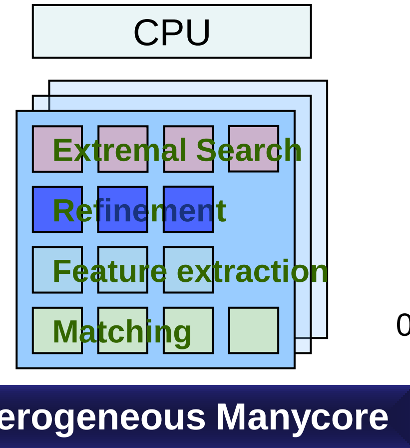
SMYLEvideo: Application Domain Specific Heterogeneous Manycore for Computer Vision

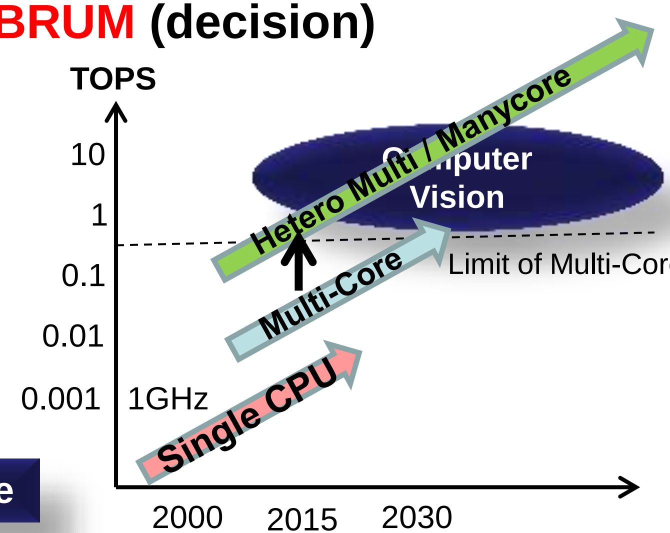
"ML and 3D Object Recognition on Image Stream"

Computer Vision (CW) ＝EYE (sensor) + CEREBRUM (decision)



- Distributed Processing
- Stream Computing





Why we need Hetero Manycore?

- **Conceptually:** Machine Learning (ML) ≒ Functional Configuration
- **High Perf. Requirement:** More than 1 TOPS
- **Inherent Parallelism:** More than 99% of processing
- **Several types of Proc.:** Huge cost with Hardwires Implementation
  - **Resolutions:** VGA, XGA, SGGA, FHD, 2K, 4K, etc.
  - **Algorithms:** SIFT, Optical Flow, Ransac, Viola & Jones, Model based Recognitions, etc.
  - **Others:** Multi-Medium Streams (MPEG-2, MPEG-4, H.264, etc.)

Key requirement is High Performance with Flexibility

What is Video Mining System

Camera(s)
Video(MPEG, H.264, …)

Data Parallel
~n streams, 2-D image blocks
Task Parallel
Gaussian Filter, Integral Map, Motion Vectors, Motion Compensation, Entropy Decode, iDCT, de-blocking Filter, etc.

Data Parallel
Image Stream by frames, Motion Streams (MVs)

20%

Feature Extraction（Low Level）

Video Decoder + α

Stream | Motion Stream | Audio Stream

- face
- building
- road
- speech
- music

Shot boundary Detection

- Frame
- Shot
- Scene
- Video Sequence

Keyword Detection（Mid Level）

x_2 x_3 ···x_n

80%

- Highlight
- Person
- Dog

Event Detection（High Level）

Data Parallel
Image Streams, Motion Streams, Pixels, Image blocks, frames, shots
Task Parallel; **Recognition Algorithms**
Viola Jone, SIFT, SVM, Matching, optical flow, etc.

Recognized Images, Extracted Images

Data Parallel
Search target, event detection
Task Parallel
Matching, Editing

More than 99% of application can be parallelized

Parallelisms in Algorithms for Video Mining

Application	Objective	Algorithm	Parallelism
Video Analysis	Prediction of Motion Vector	Optical Flow	Line
	Specific feature detection and extraction	SIFT[*1]	Data Partitioning
	Detection of human, and tracking	Cascaded Haar Like	Pixel Level
	Line detection for field separation	Huff	Line
	Elimination of error from continuous frames	Ransac[*2]	Sample Data
Human Search	Face detection from several angles	Vector Face Detection	Pixel Blocks
	Extraction of features on faces	Model Based Face Detection	Task Level
	Specific feature detection and extraction	SIFT	Data Partitioning
Video Editing	Segment Extraction	Graph based Segmentation	Grid Level
	Detection of Motion Vector	Block Matching	Line

*1 SIFT：Scale-Invariant Feature Transform

*2 Ransac：Random Sample Consensus

Many type of pallarelisms are inherent in algorithms for Video Mining

Goals of SMYLEvideo Manycare

- Real-Time Processing : **1TOPS~**
- Scalability : **10fps, 20fps, 30fps**
- Programmability : **Software Based Implementation**
- Flexibility : **OpenCV (Computer Vision)**

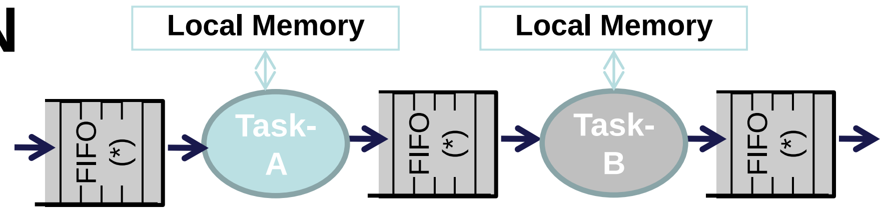
  SIFT, Optical Flow, Ransac, Viola & Jones, Model based Recognitions, SVM, etc.
- Low Power : ~**1.5W**
- Low Clock Frequency : ~**100MHz**

High Performance, Programmable, Scalable

System Level Architecture of SMYLEvideo

Distributed Processing with KPN

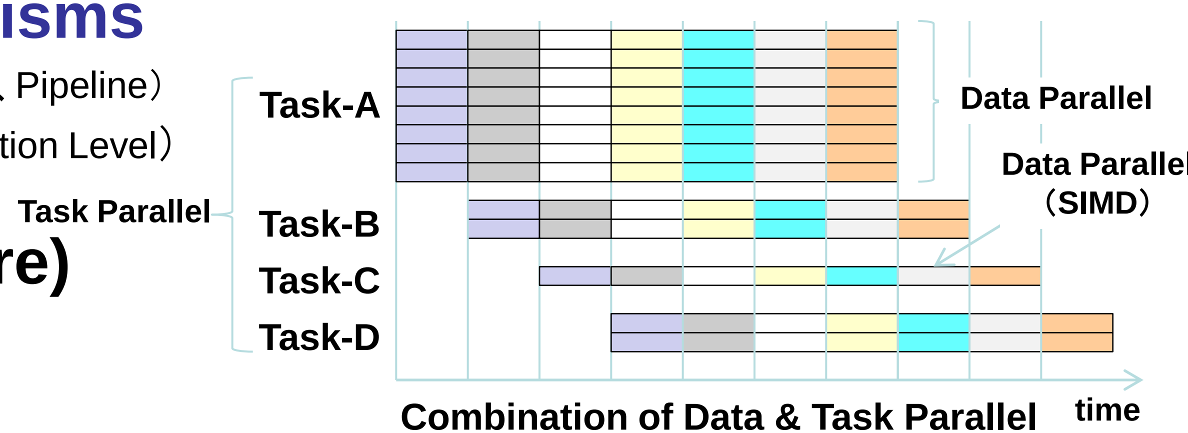
- Non-Shared Memory Processes
- <u>Zero-Overhead Message Passing Mechanism</u>



Kahn Process Network

Combination of Parallelisms

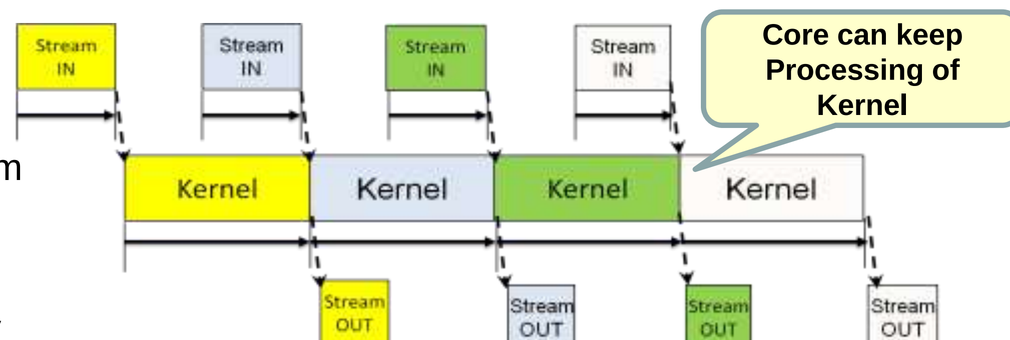
- Distributed Parallel Processing（Task、Pipeline）
- Data Parallelism（High-Level、Instruction Level）



Combination of Data & Task Parallel

Stream Processing (Core)

- Kernel
- Stream-In (Read Message)
- Stream-Out (Write Message)

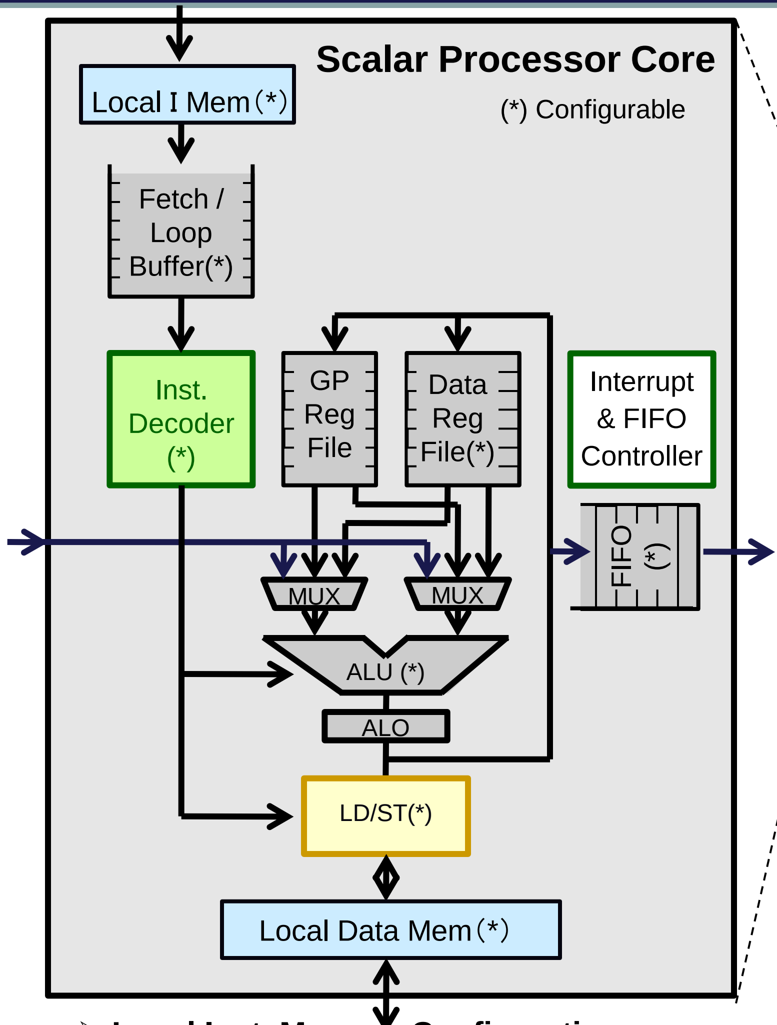


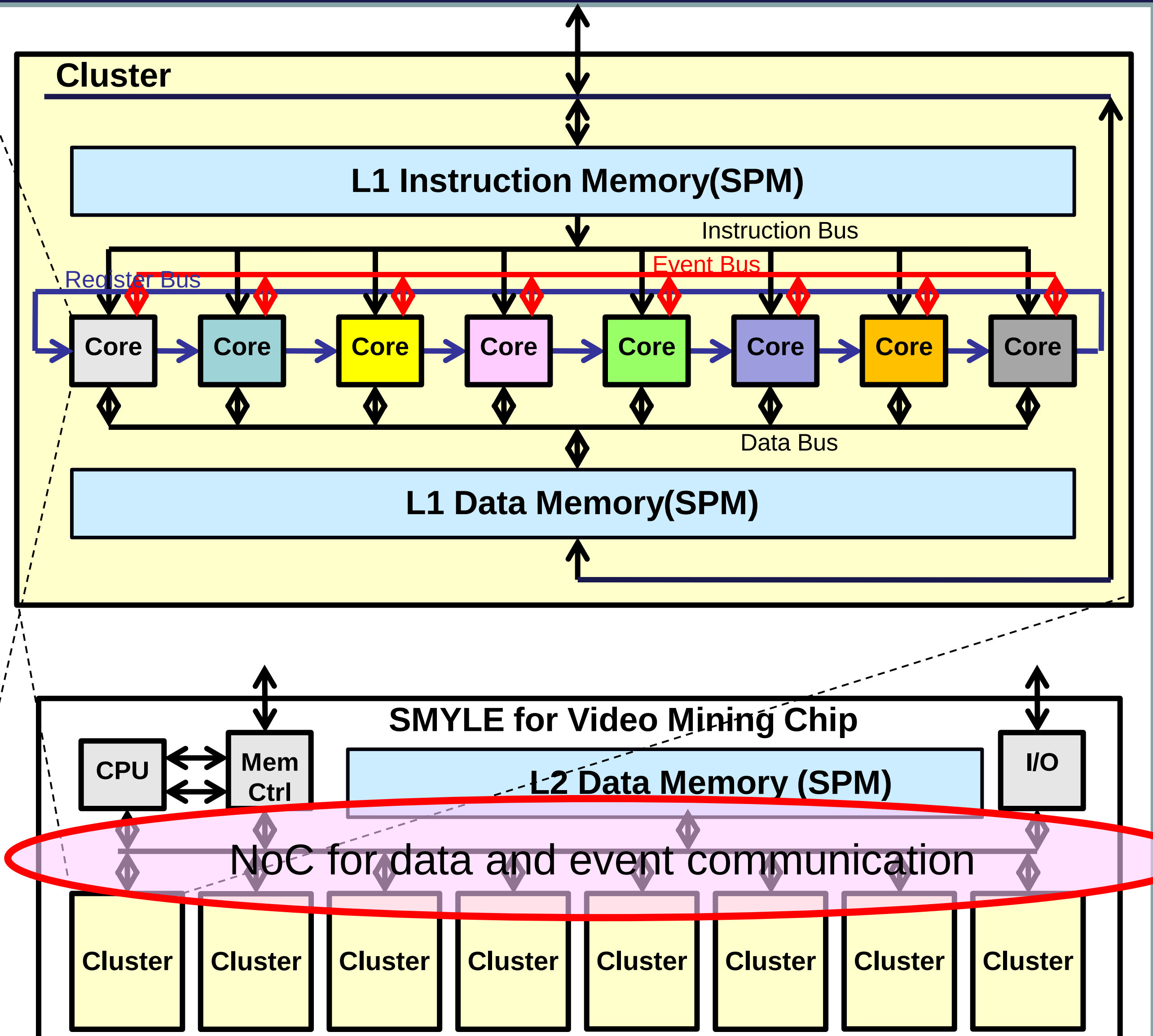
Optimization of Core

- Support Stream Processing：background Stream
- Complex Inst：Reduction of Kernel cycle
- FIFO support mechanism
- Reduction of energy for instruction / data supply

Distributed Processing, ZOMP, Task Parallel, Stream Processing , ASIP

SMYLEvideo : Basic Architecture



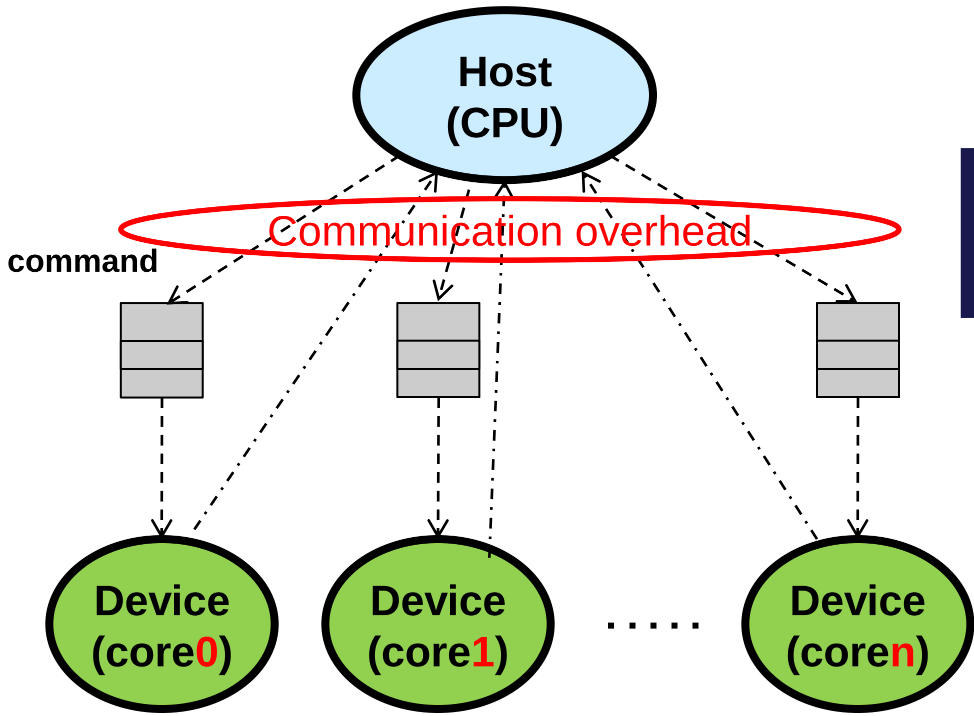


- ➢ Local Inst. Memory Configuration
- ➢ Loop Buffer Configuration
- ➢ Instruction Extention (Decoder, ALU, LD/ST)
- ➢ Local Data Memory Configuration

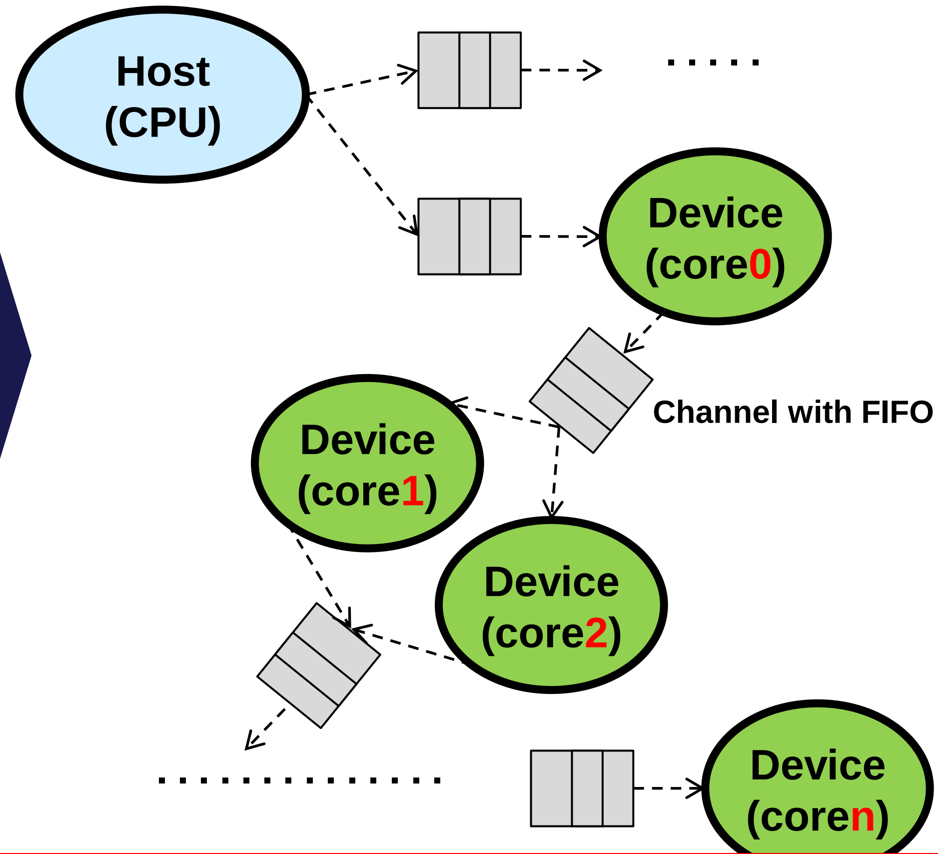
Application Domain Specific Scalable Heterogeneous Manycore

Partitioning
OpenCL vs. Distributed Processing

- **OpenCL (CPU centric)**
  - Bottleneck
    - Processing on Host
    - Increasing communication with Host
  - **Hard to express distributed processing**



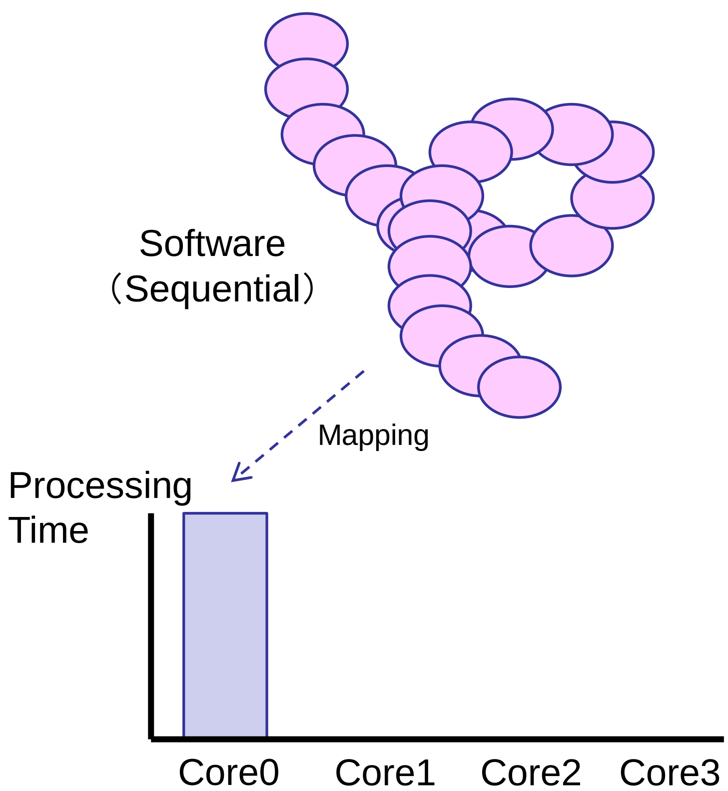
- **Distributed Processing**
  - **Scalability**
  - **Can combine with Data Parallel Processing**

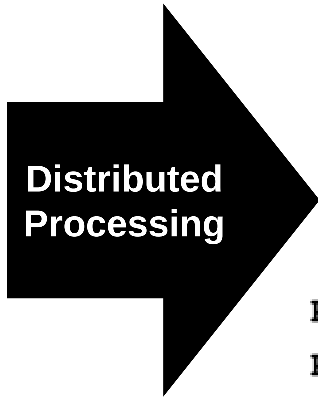


Take an approach of Distributed Processing for removing bottlenecks

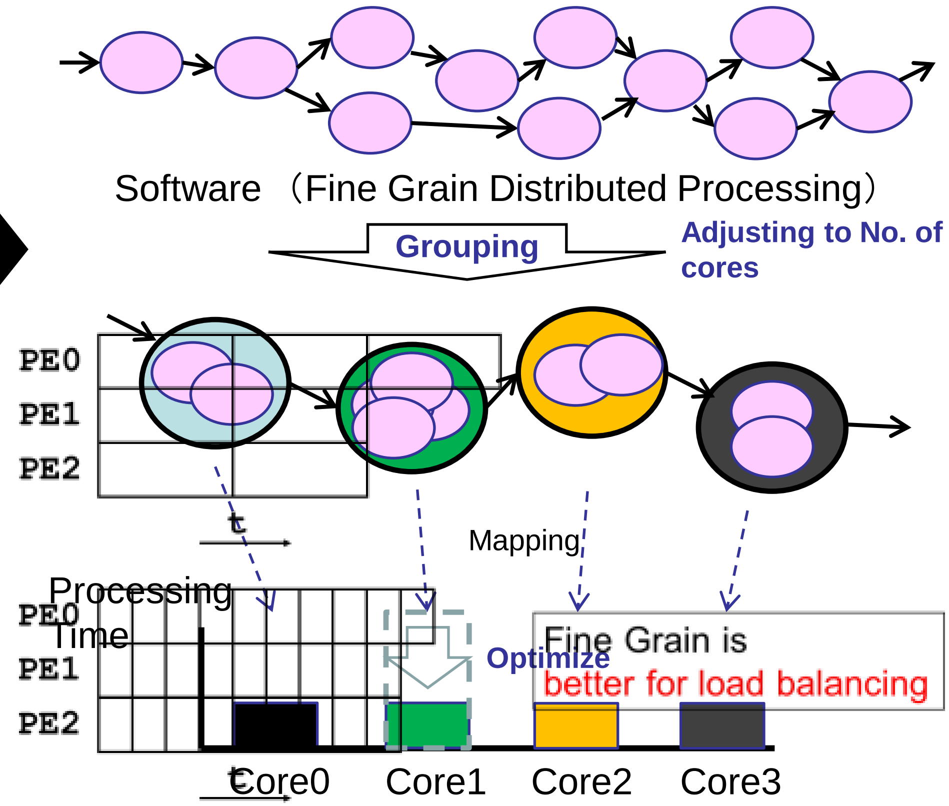
Software Partitioning

Sequential to Distributed Processing





- ☐ Profiling
- ☐ Streaming
- ☐ Partitioning
- ☐ Grouping
- ☐ Optimizing



<Issues>

- Cannot utilize **Multi-Core / Many-Core**
- **Requires High Performance** for Video Mining

<Advantages>

- Utilize Multi-Core / Many-Core resources
- **Easy to balance the load**

Investigation has done on Many Algorithms ; Viola & Jones, SVM, SIFT, etc.

Approach for Energy-Efficient Computing

- Goal : **High-Performance & Low-Power Programmable Accelerator**
  (Energy-Efficient, Low Cost, Flexible, Scalable)
- Approach : **Low Clock Frequency**

◆ Power consumption

$$P_{Total} = P_{Dynmic} + P_{Static}$$

$$= \frac{1}{2}\alpha C V^2 f \alpha + I_{Leak} V$$

C: Load capacity
V: Source voltage
f: Frequency
α: Siwitching rate
I_{Leak}: Leakage current

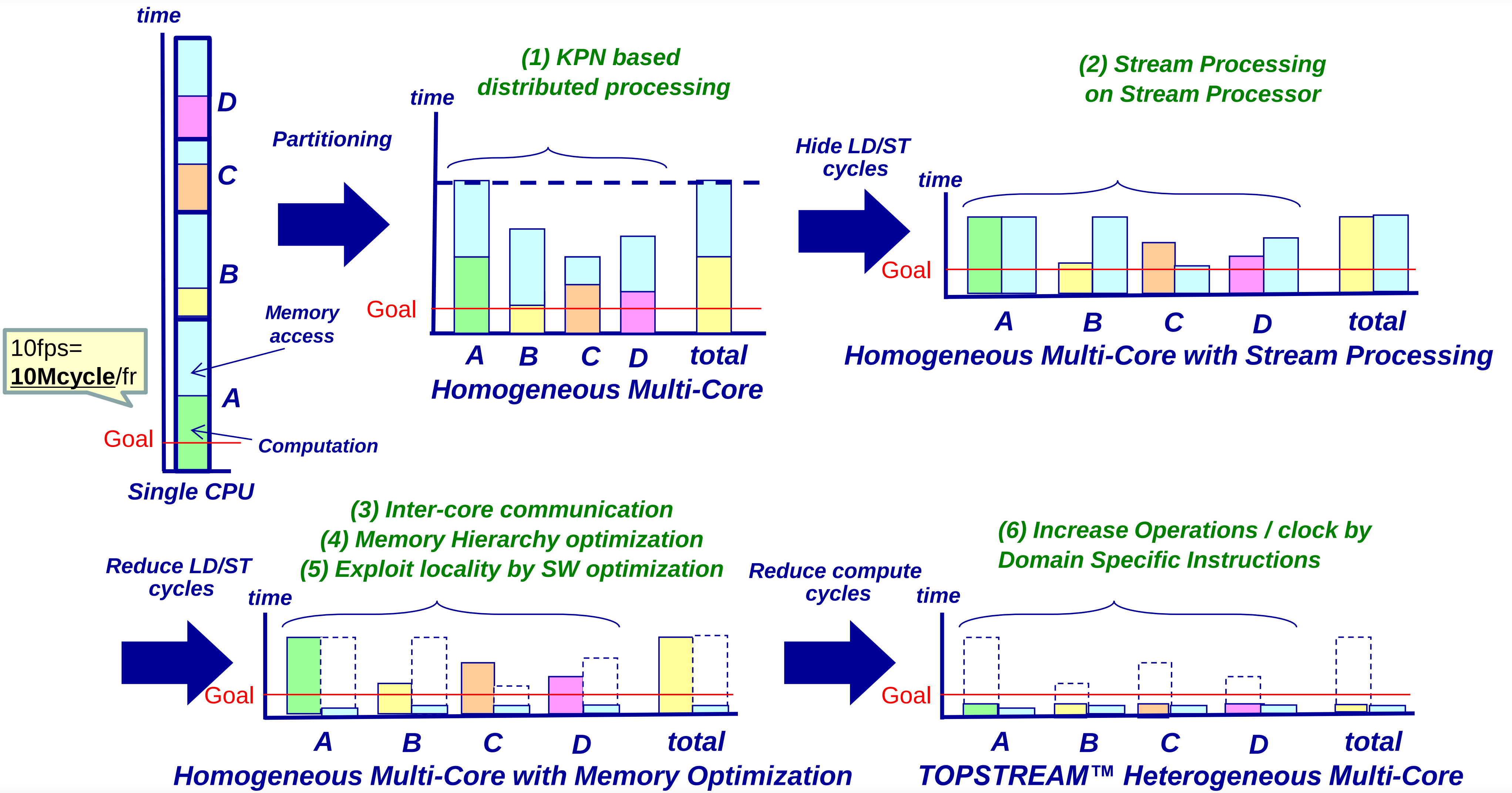
◆ Performance

$$\text{Performance} = \text{OPC} \times f$$

OPC: Operation Per Clock

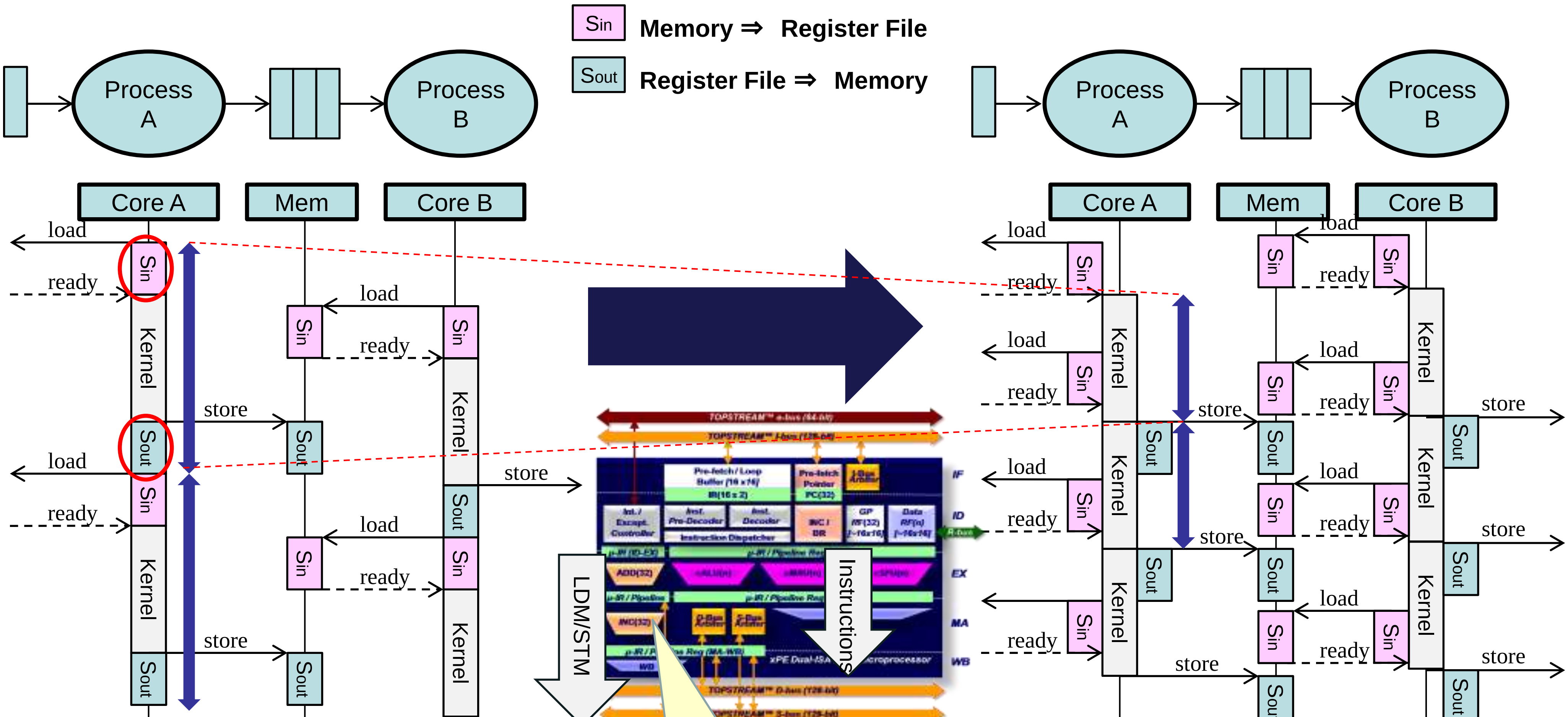
High Performance @ Low Clock Frequency drives Low Power

Approach to reduce clock frequency with Architecture-Algorithm Co-Design



Stream Processing Core

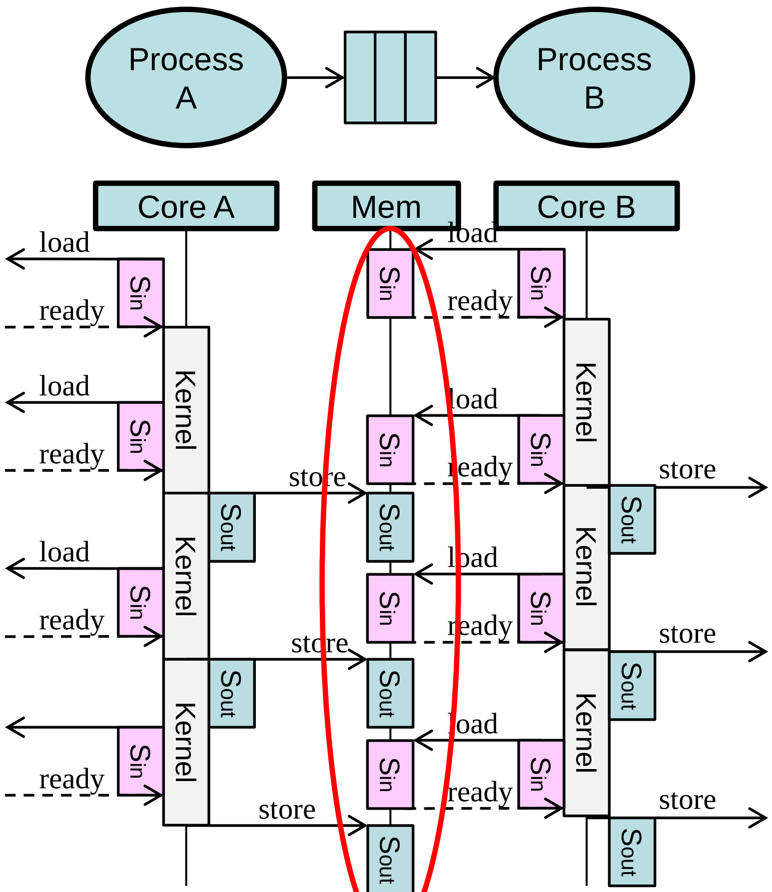
- Hide overhead of Stream-In and Stream-Out



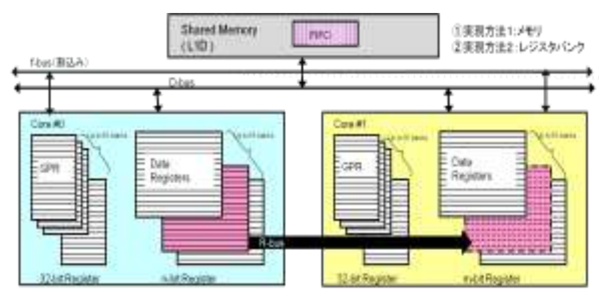
Hide cycles for inter-process communication (Stream-In and Stream-Out)

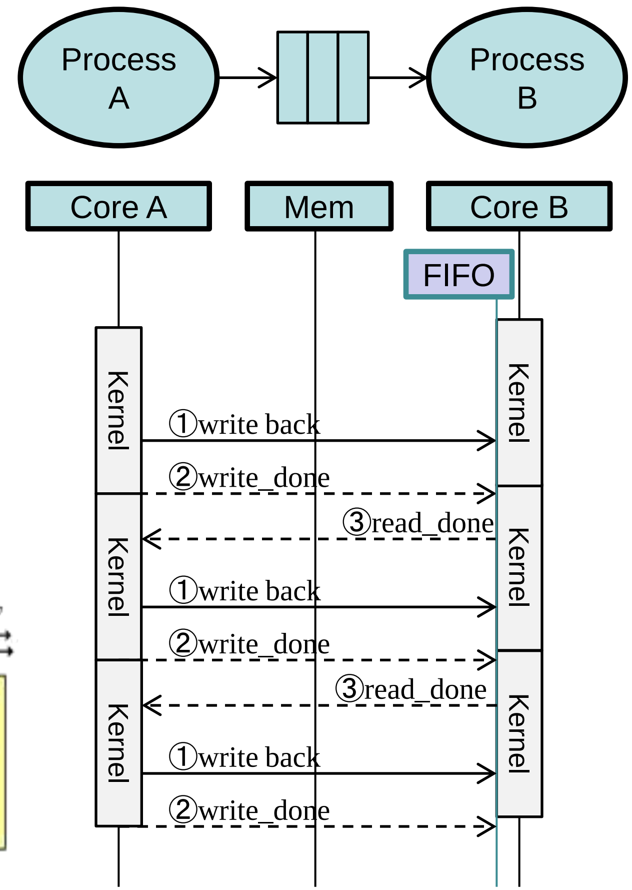
Inter-Core FIFO : Register Bank Sharing

- Reduction of Memory Access Bandwidth and its Energy



Register Bank Sharing

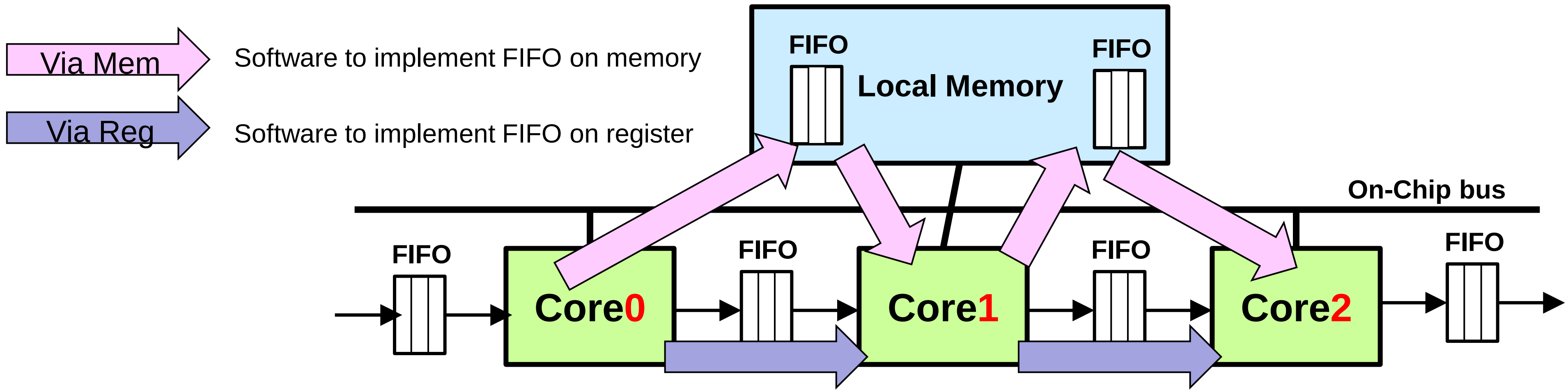


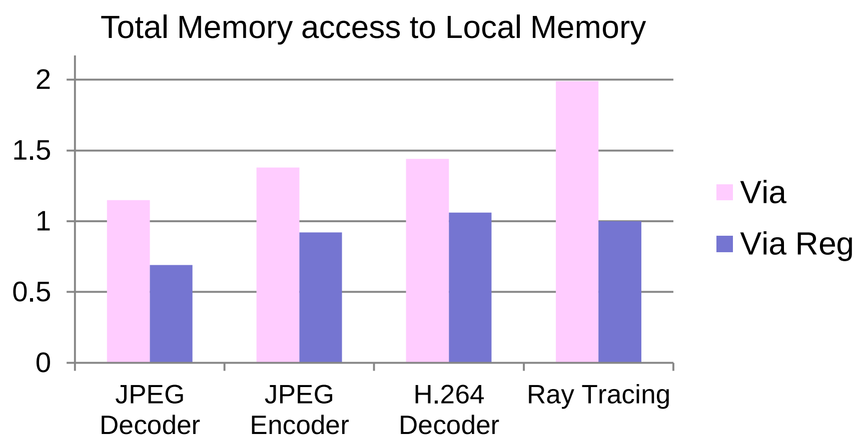


Reduce memory access time and Energy!

Reduction of memory traffic

■ **Path for Message Passing**

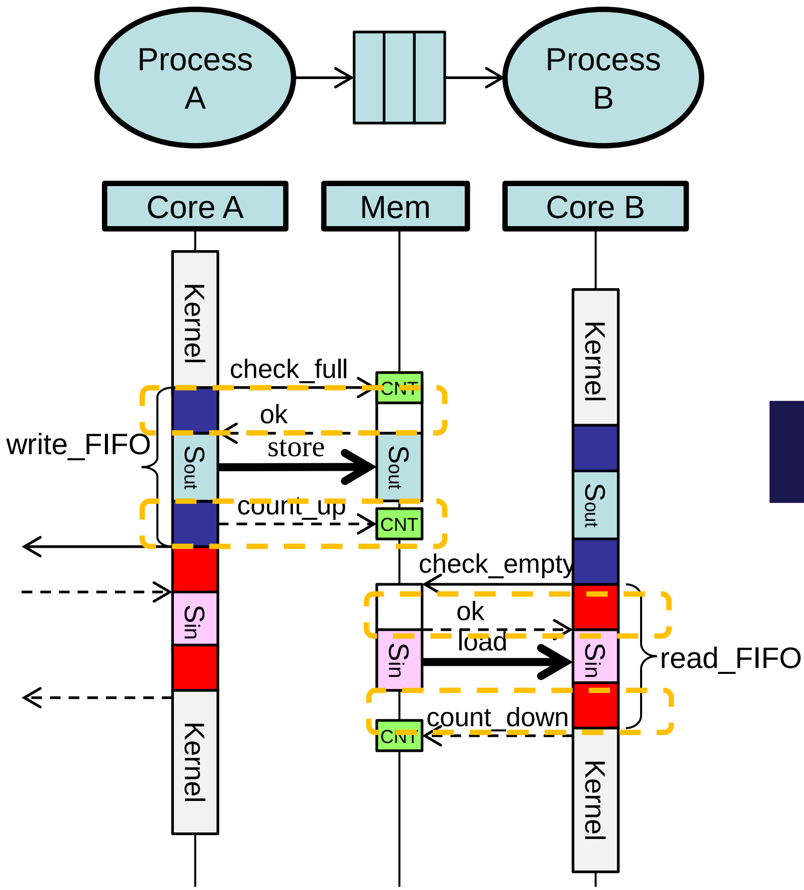




Significant Reduction of Memory Trafifc : more than 30%

ZOMP
Zero-Overhead Message Passing Mechanism

- Remove cycles and memory access for checking FIFO counts and synchronization

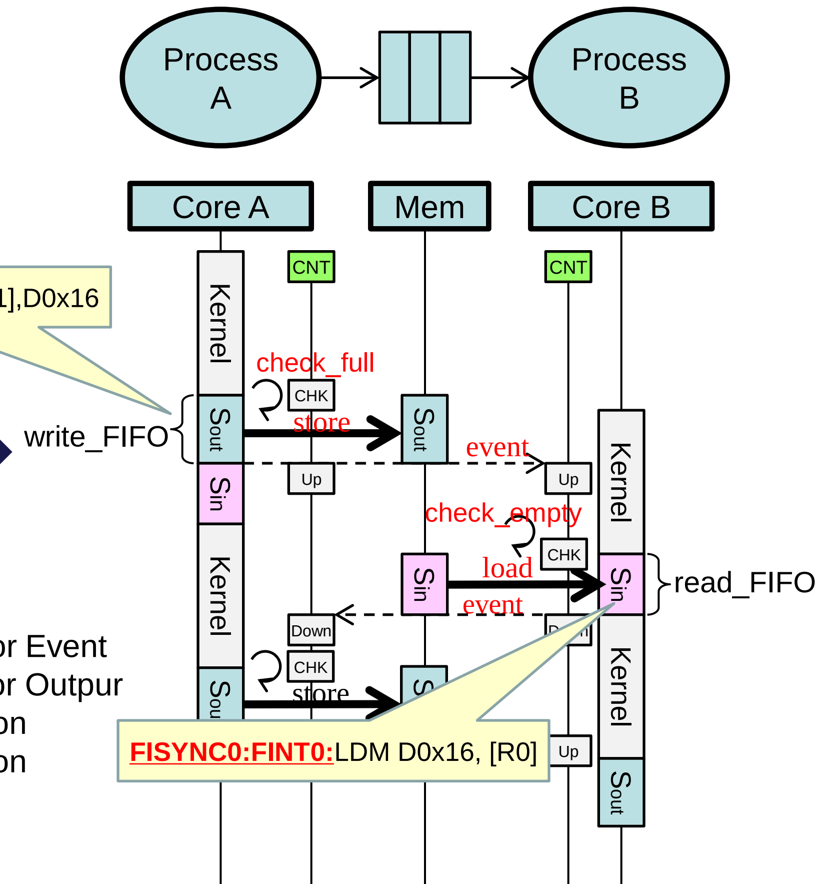


<Hardware>

- FIFO Configuration Reg
- FIFO Counter

<Sofftware>

- Prefix I nstruction
  - -FISYNCn: Check & Wait for Event
  - -FOWAITn: Check & Wait for Outpur
  - -FOINT: Event Generation
  - -FIINT: Event Generation

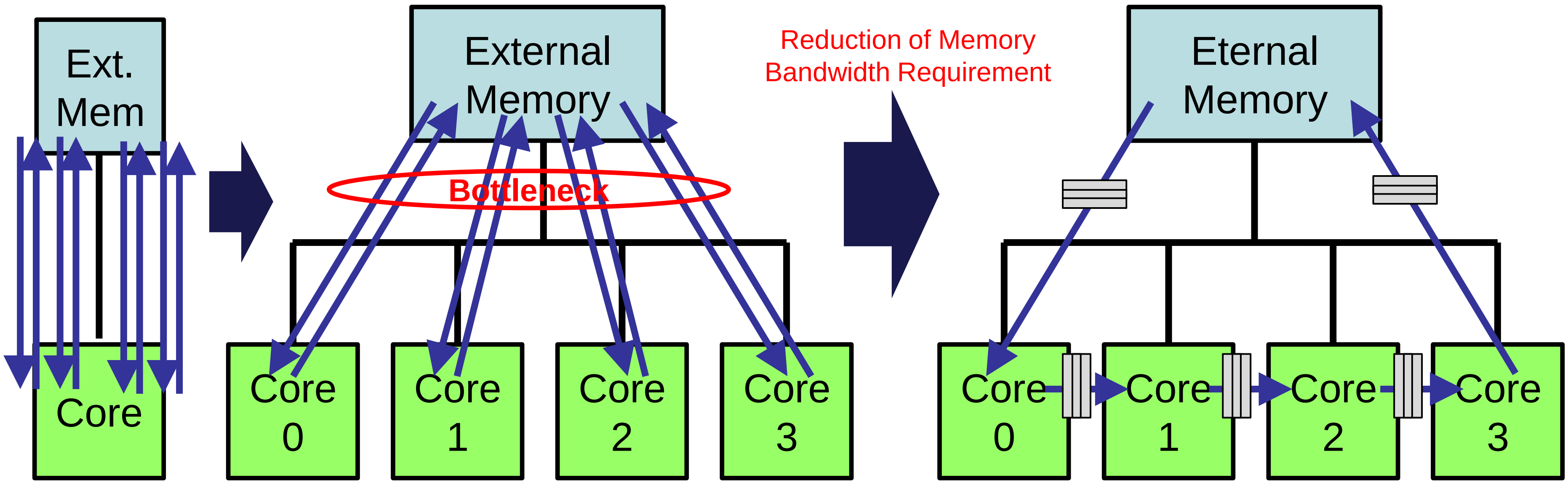


No overhead : Just Add Prefix instructions on Stream-In & Stream-Out

Memory Access reduction by Distributed Stream Processing

- **Memory Centric Processing**
  - **- Each core works data on External Memory**
  - **- Integration of processors and memories**

- **Distributed Stream Processing**
  - **Core to Core Stream passing**
    - **On-Chip memory**
    - **Register Sharing**



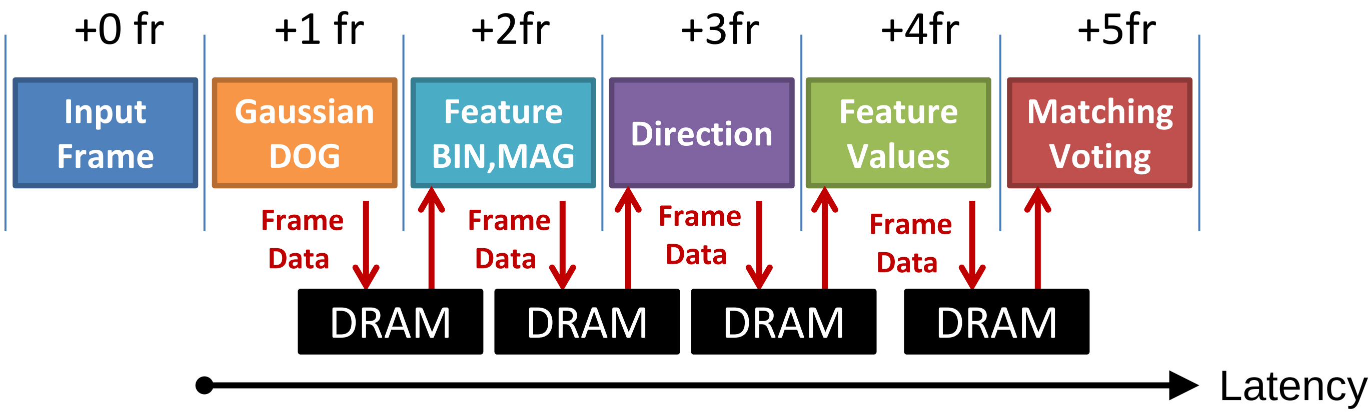
Frame based processing **Block based processing**

Increase scalability by reducing memory bandwidth requirement

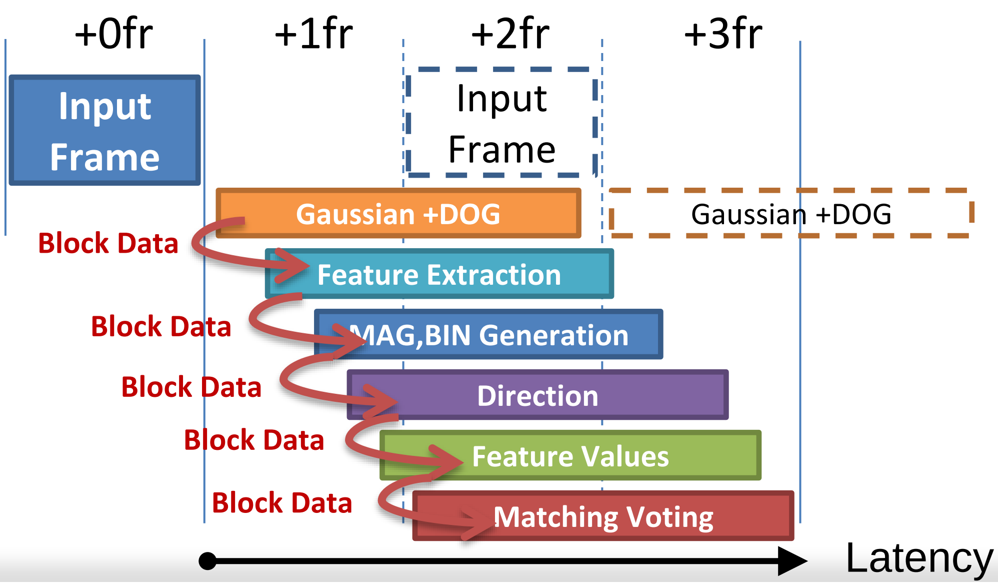
Frame based vs. Block based Processing

Ex). SIFT

Frame based processing



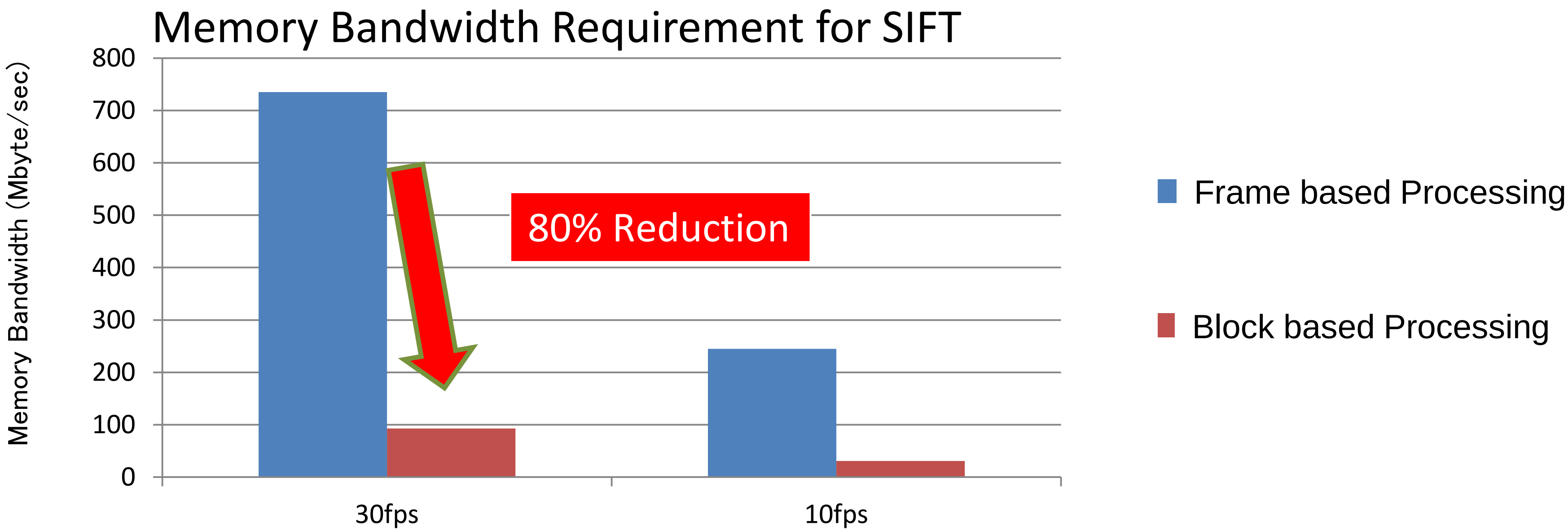
Block based processing



Smaller latency with Smaller Data

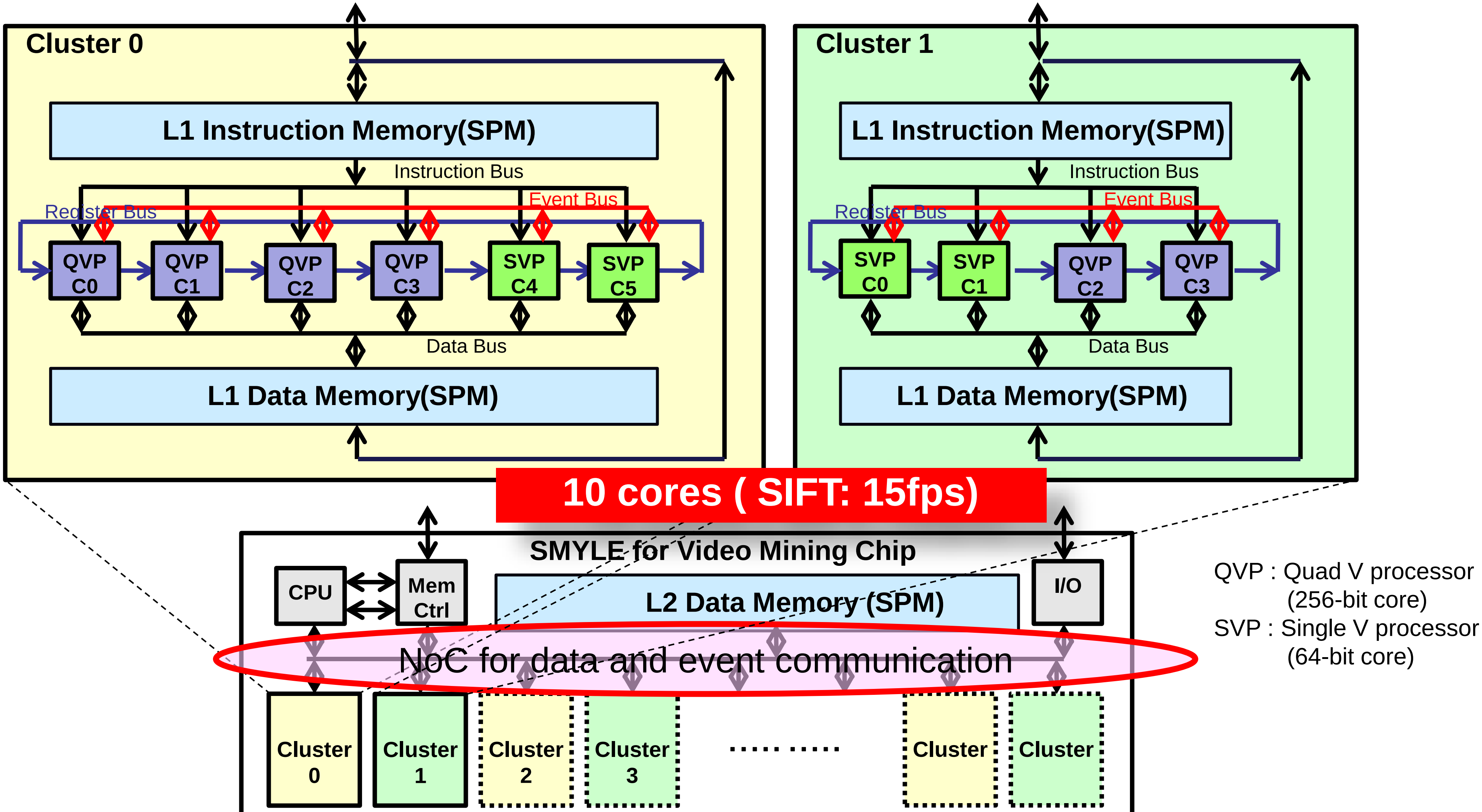
Frame based vs. Block based Processing

	Frame based Processing	Block based Processing
Global Memory Usage	22Mbytes	3.1Mbytes
Cluster Local Memory Usage	0.15Mbytes	0.8Mbytes
Recognition Latency	167mSec(5.1frame)	100mSec(3frame)



Memory Usage : 1/7, Memory Bandwidth Requirement : 1/5

SMYLEvideo Configuration



Scalable Performance and Functionality with adding Clusters

Conclusions

- **Manycore** will play a crucial role in extending the roadmap for enabling the next generation **SoCs** required for **"Video Mining"** one of Computer Vision systems.
- **Zero-Overhead Message Passing Mechanism (ZOMP)** can **efficiently increases the system performance** and **scalability** of Manycore processors.
- **Block based distributed processing** drastically **reduces memory access bandwidth** and **increases room for higher performance** on Manycore processors.
- **SMYLEvideo** provides **scalability in performance and functionality** with its **clustered architecture**.