Introduction to PlayStation®2 Architecture

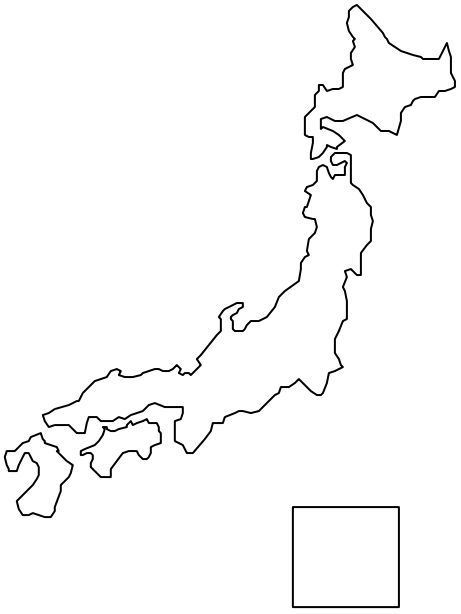
James Russell
Software Engineer

SCEE
Technology Group

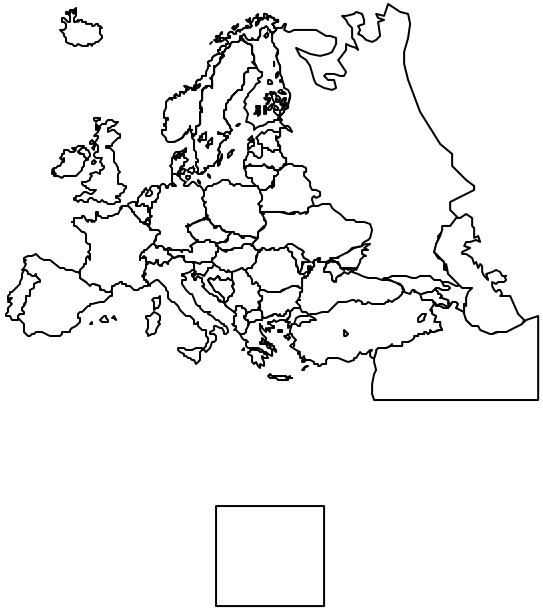
In this presentation

- Company overview
- PlayStation 2 architecture overview
- PS2 Game Development
- Differences between PS2 and PC.

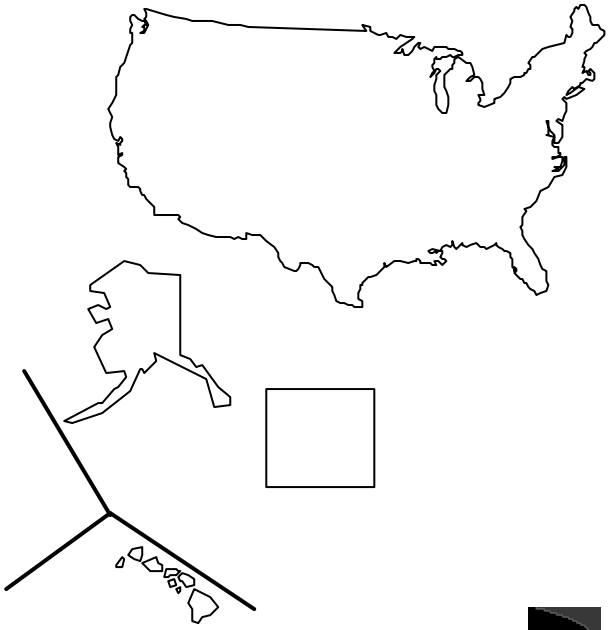
1) Sony Computer Entertainment Overview



Japan



SCE Europe
(includes Aus,
NZ, Mid East,
Southern Africa)



America

echnology Group

Sales

- 40 million sold world-wide since launch
  - Since March 2000 in Japan
  - Since Nov 2000 in Europe/US
- New markets: Middle East, India, Korea, China
- Long term aim: 100 million within 5 years of launch
- Production facilities can produce 2M/month.

Design considerations

- Over 5 years, we'll make 100,000,000 PS2s
- Design is very important
  - Must be inexpensive (or should become that way)
  - Technology must be ahead of the curve
- Need high performance, low price.

How to achieve this?

- Processor yield
  - High CPU clock speed means lower yields
- Solution?
  - Low CPU clock speed, but high parallelism
- Nothing readily available
  - SCE designs custom chips.

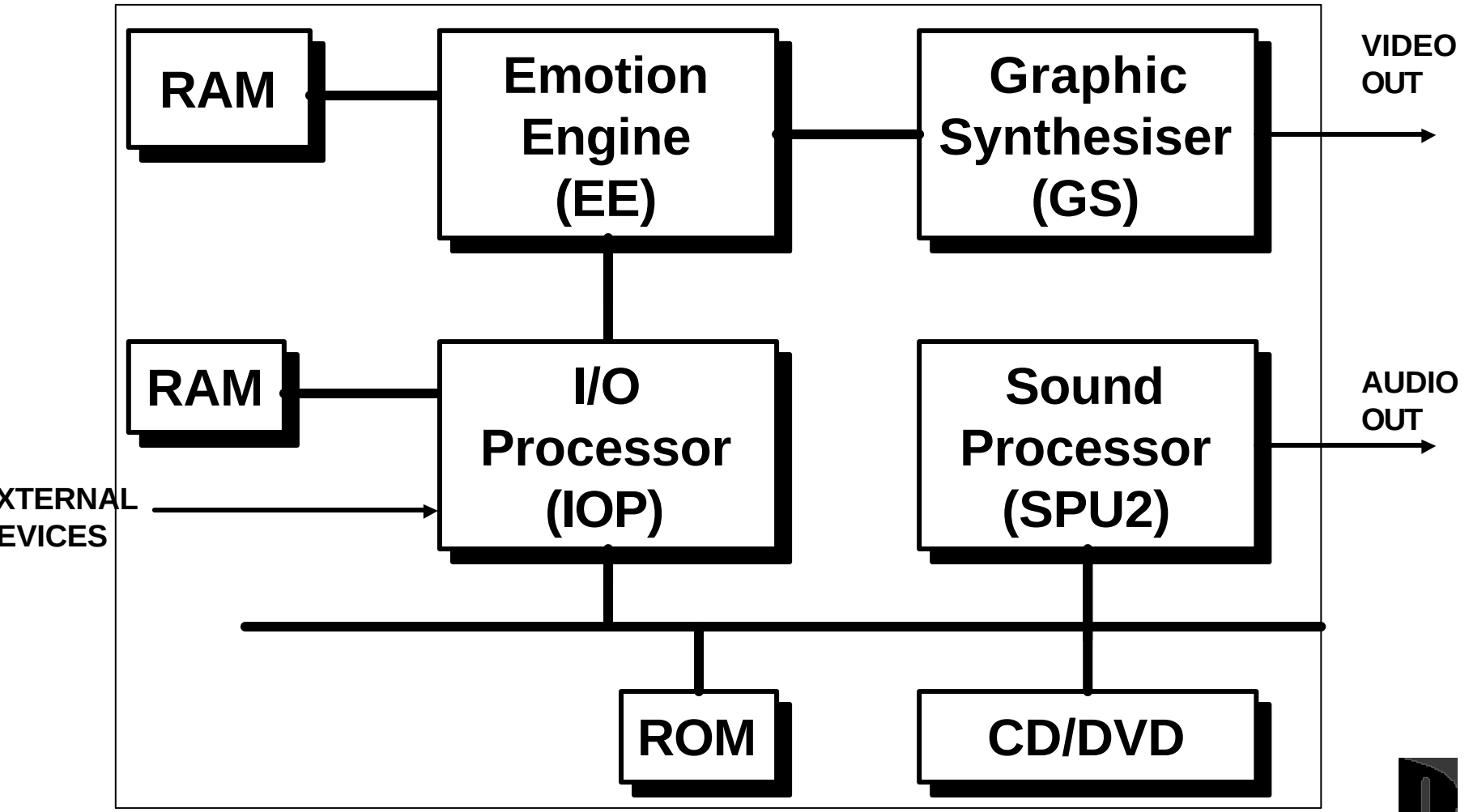
2) Technical Aspects of PlayStation 2

- 128-bit CPU core “Emotion Engine”
  - + 2 independent Vector Units
  - + Image Processing Unit (for MPEG)
- GS - “Graphics Synthesizer” GPU
- SPU2 - Sound Processing Unit
- I/O Processor (CD/DVD, USB, i.Link).

“Emotion Engine” - Specifications

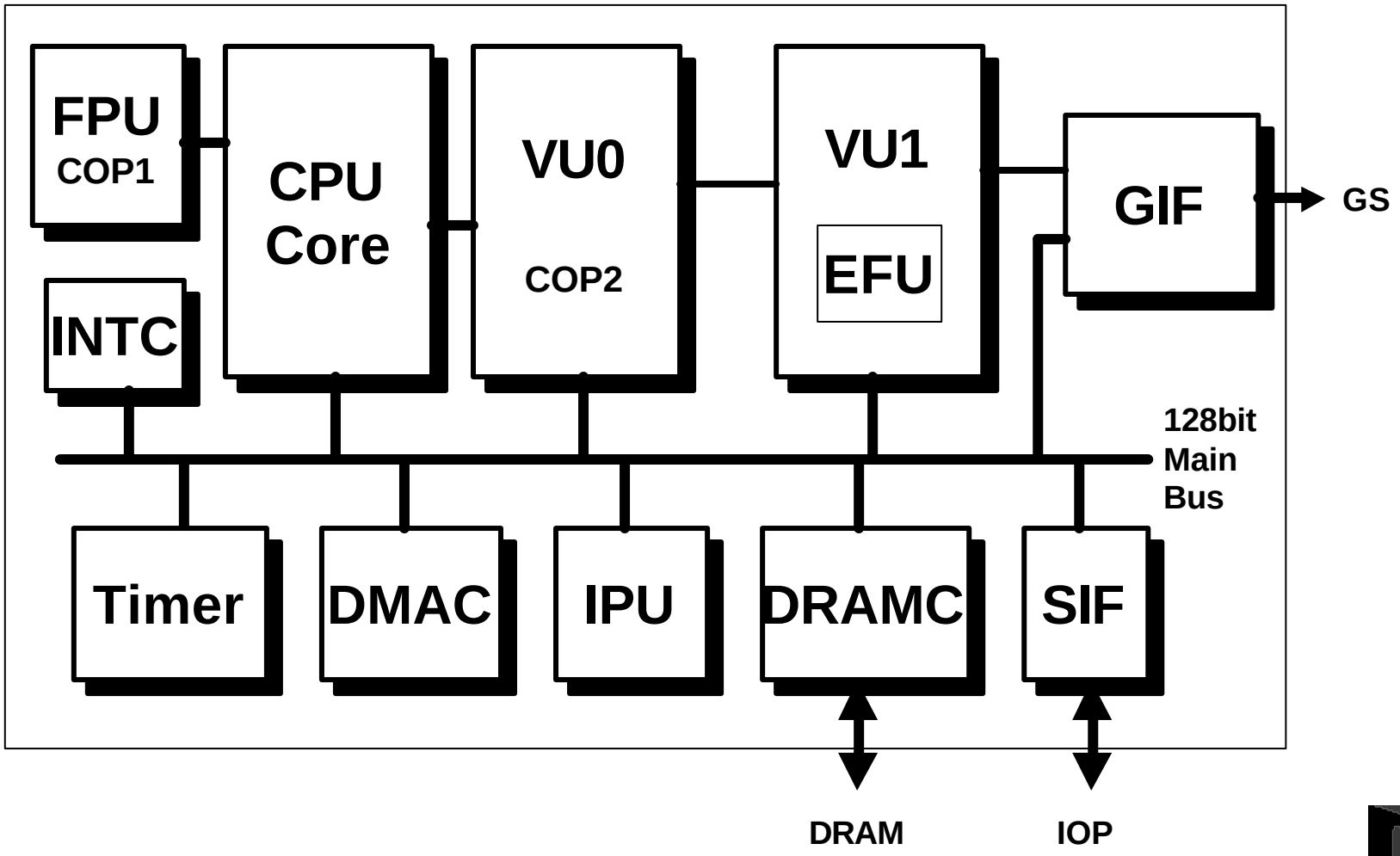
↗ CPU Core	128 bit CPU
↗ System Clock	300MHz
↗ Bus Bandwidth	3.2GB/sec
↗ Main Memory	32MB (Direct Rambus)
↗ Floating Point Calculation	6.2 GFLOPS
↗ 3D Geometry Performance	66 Million polygons/sec.

System Architecture



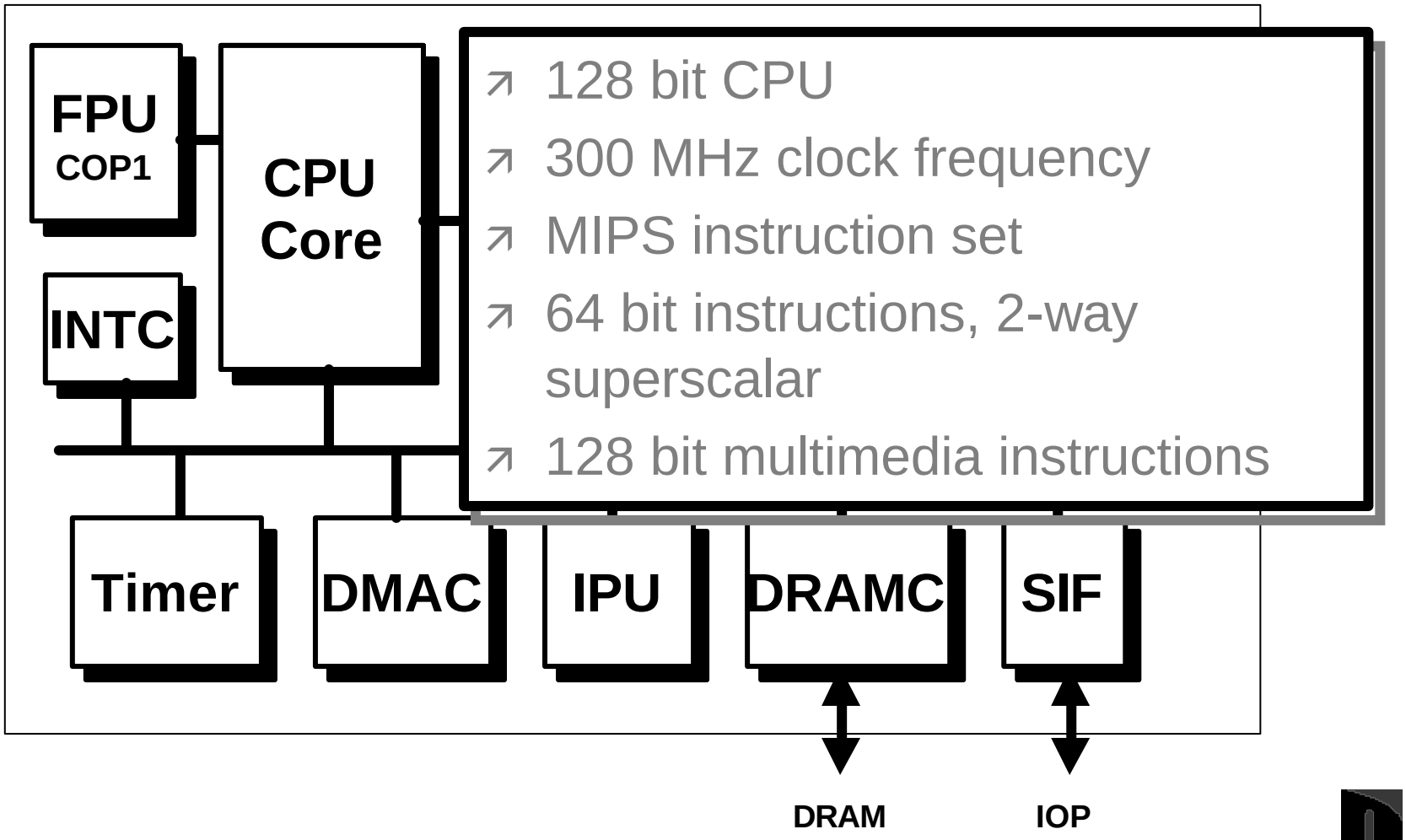
Emotion Engine architecture

Overview



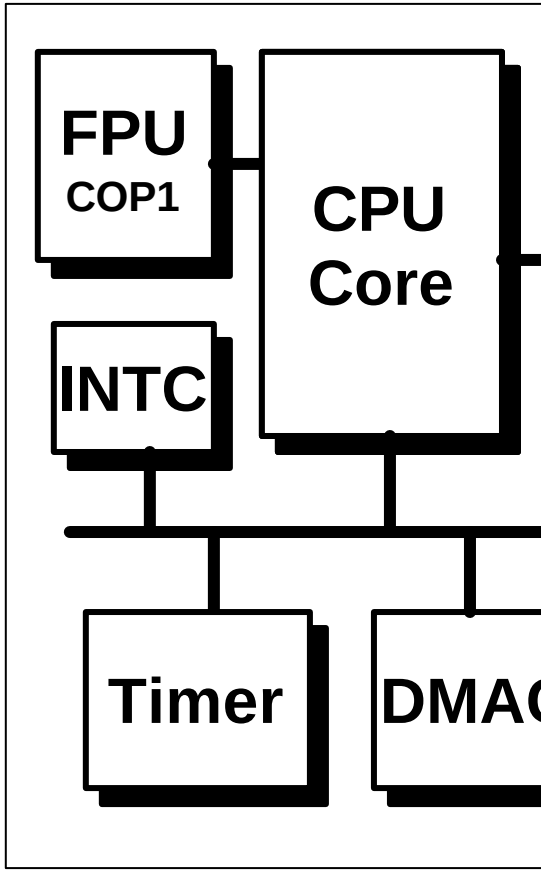
Emotion Engine architecture

CPU Core

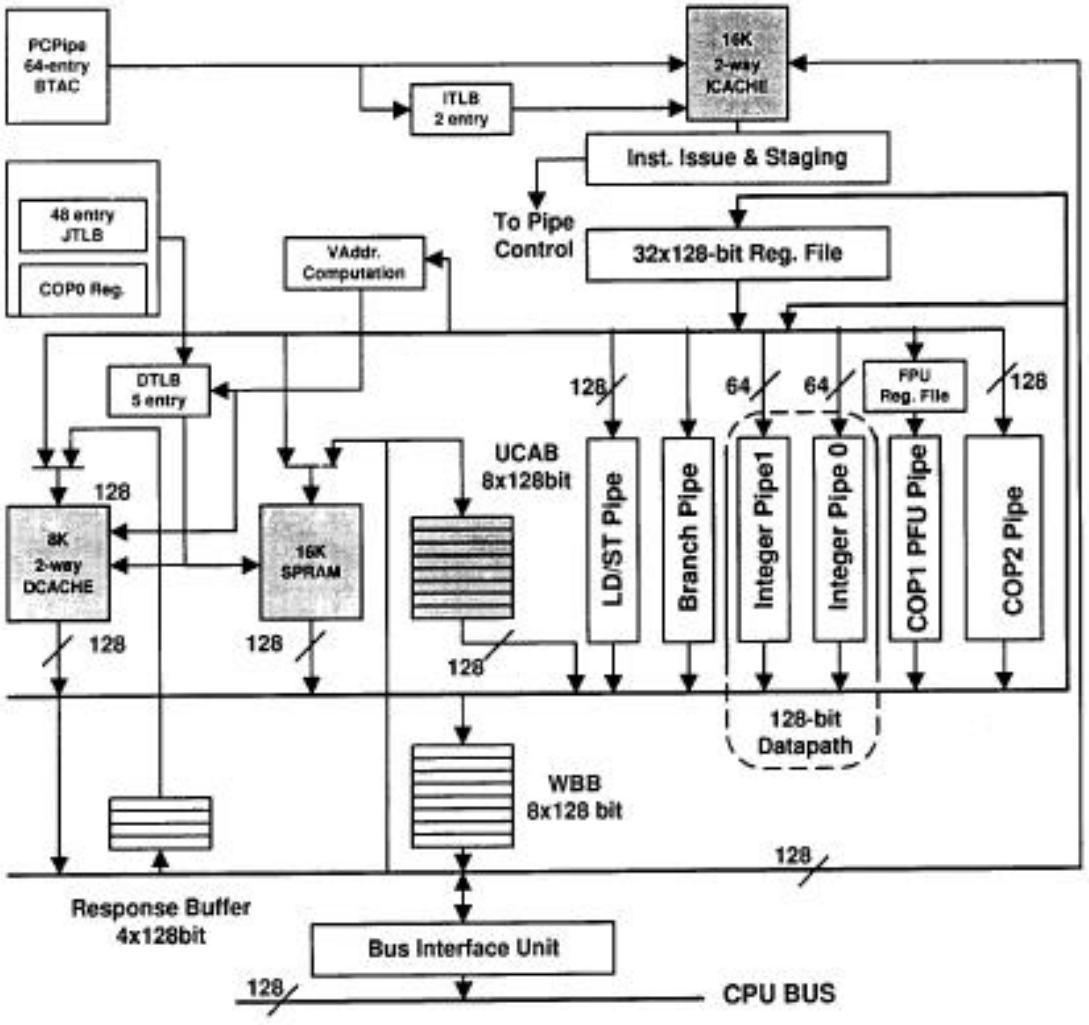


- 128 bit CPU
- 300 MHz clock frequency
- MIPS instruction set
- 64 bit instructions, 2-way superscalar
- 128 bit multimedia instructions

Emotion Engine architecture

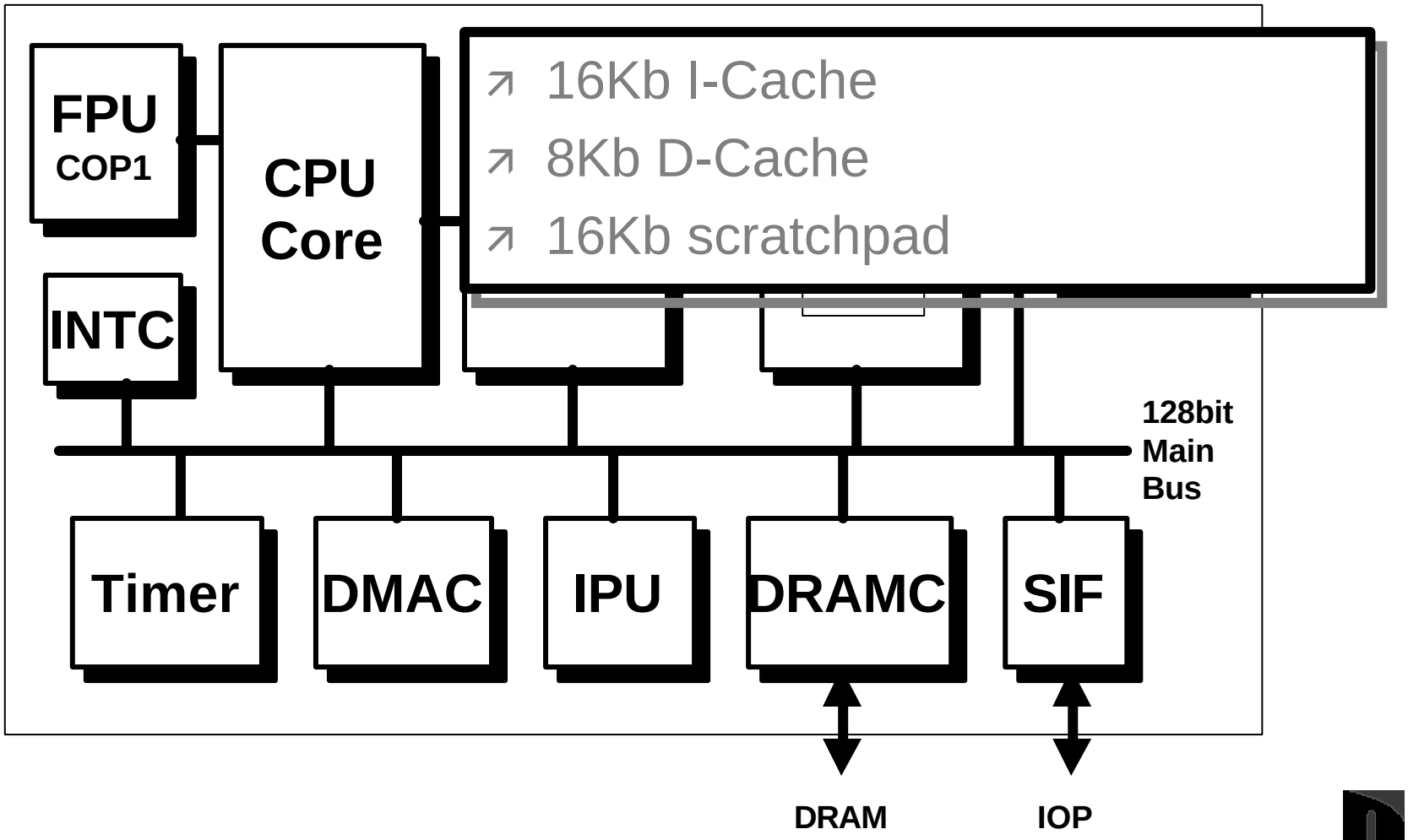


CPU Core



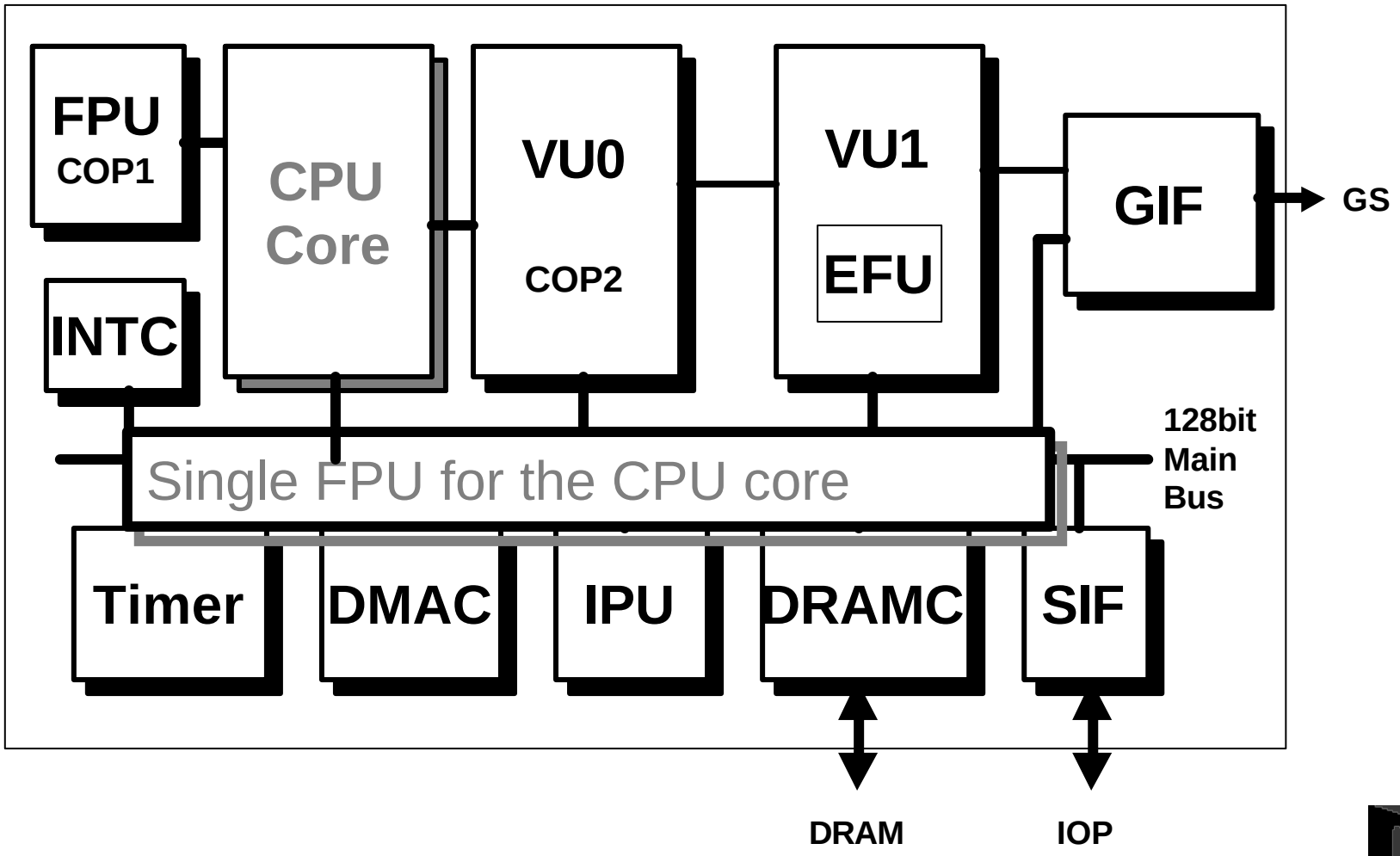
Emotion Engine architecture

CPU Core



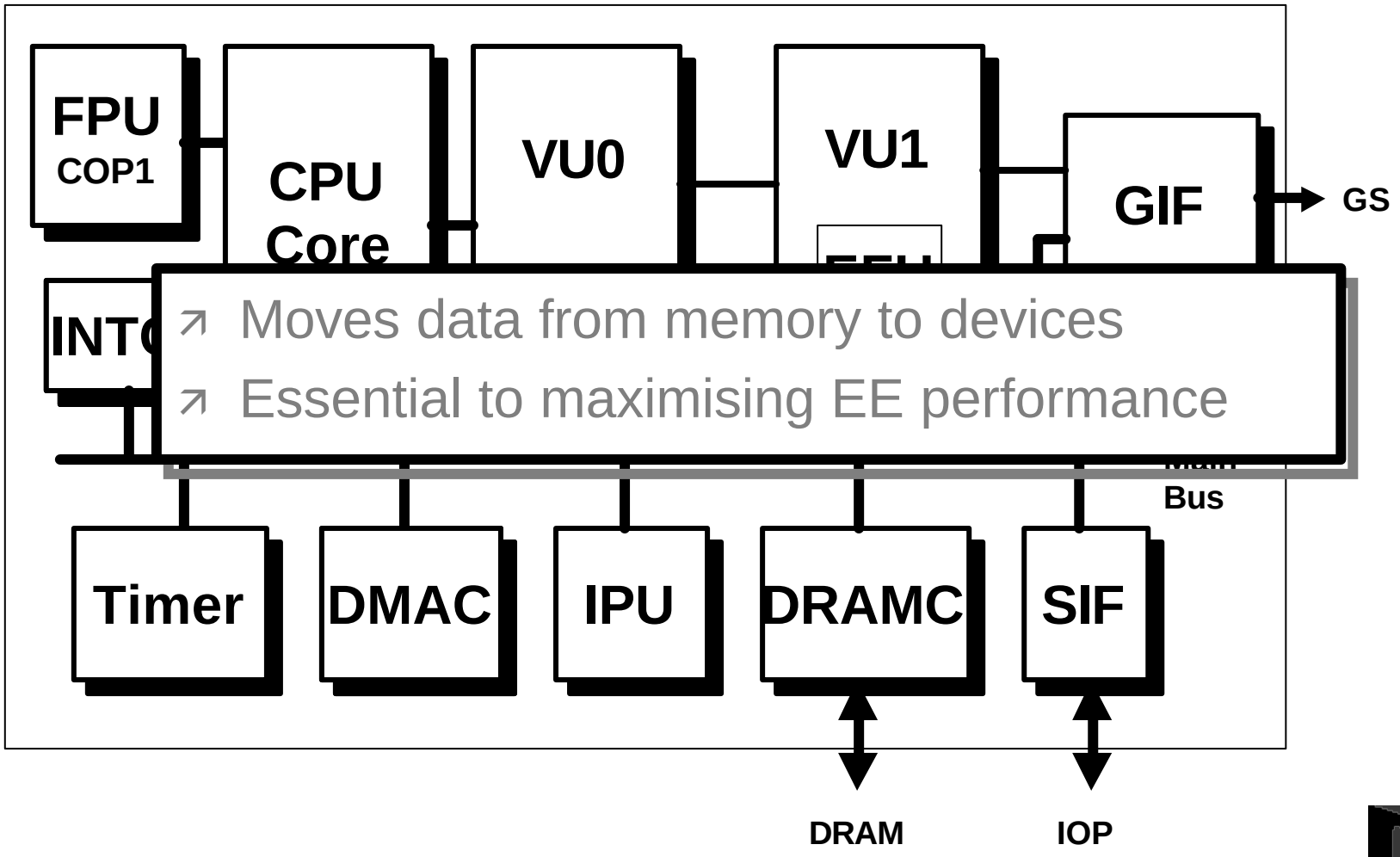
Emotion Engine architecture

Floating Point Unit (FPU)



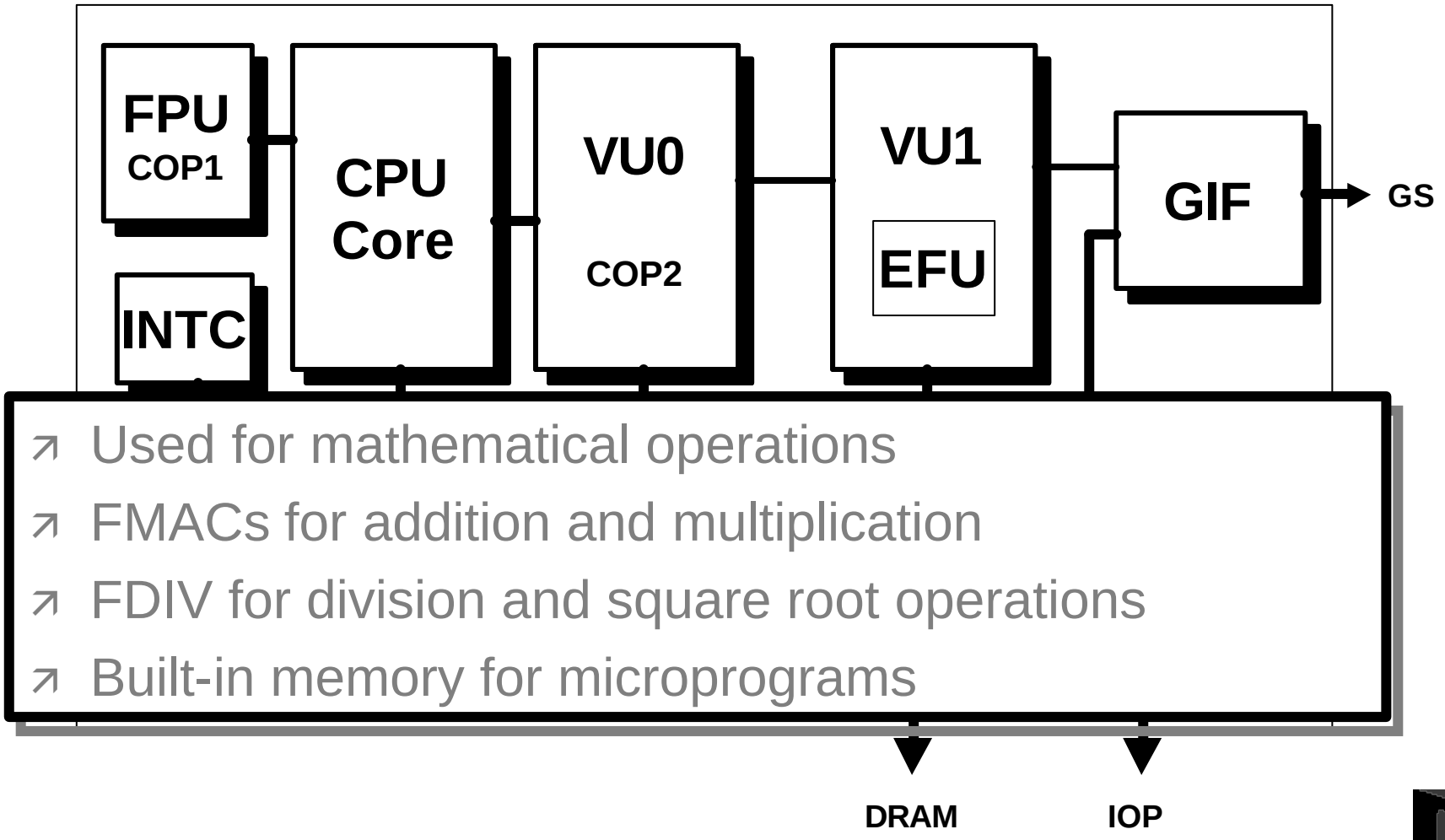
Emotion Engine architecture

DMA Controller (DMAC)



Emotion Engine architecture

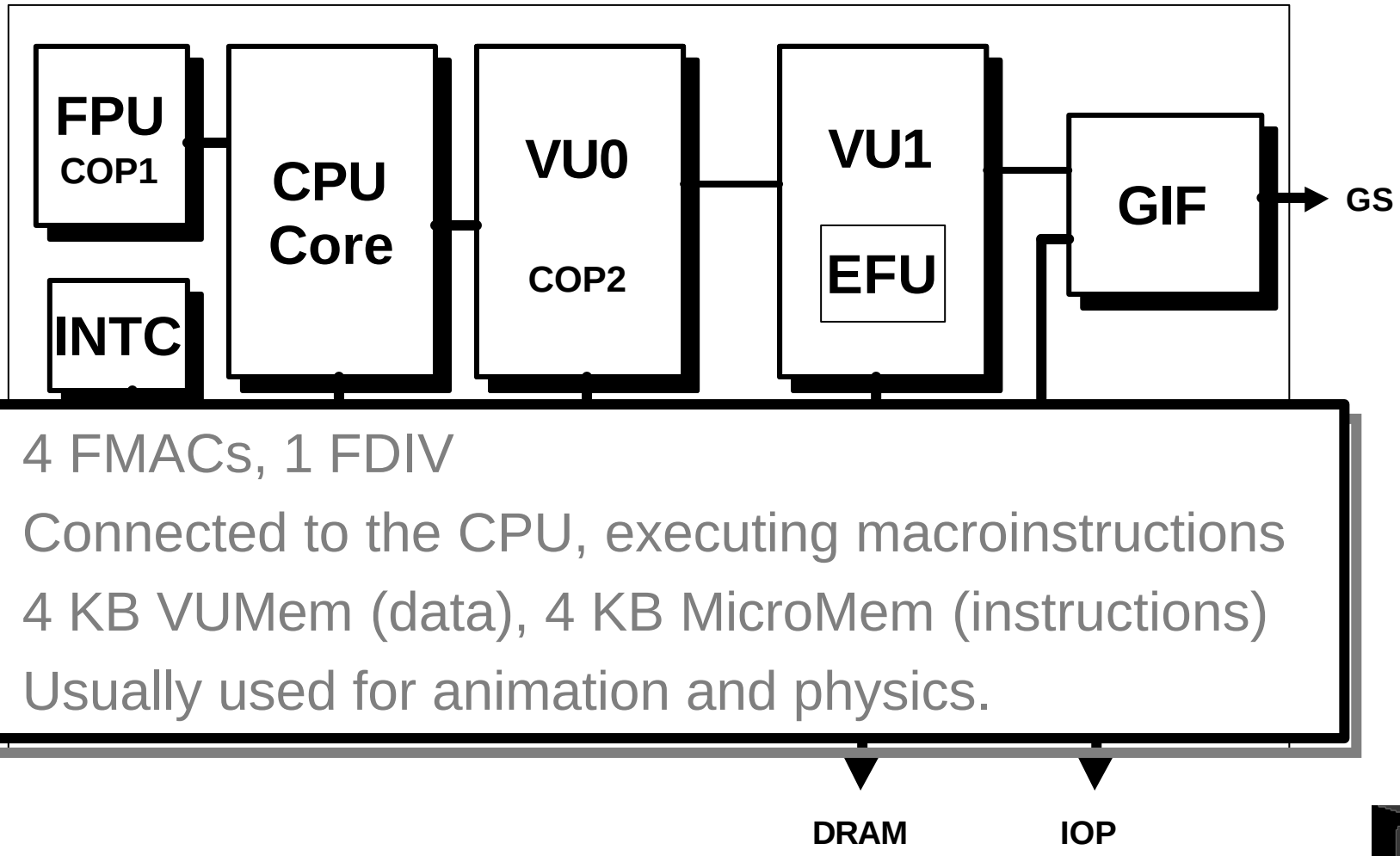
Vector Processing Units (VU0 & VU1)



- Used for mathematical operations
- FMACs for addition and multiplication
- FDIV for division and square root operations
- Built-in memory for microprograms

Emotion Engine architecture

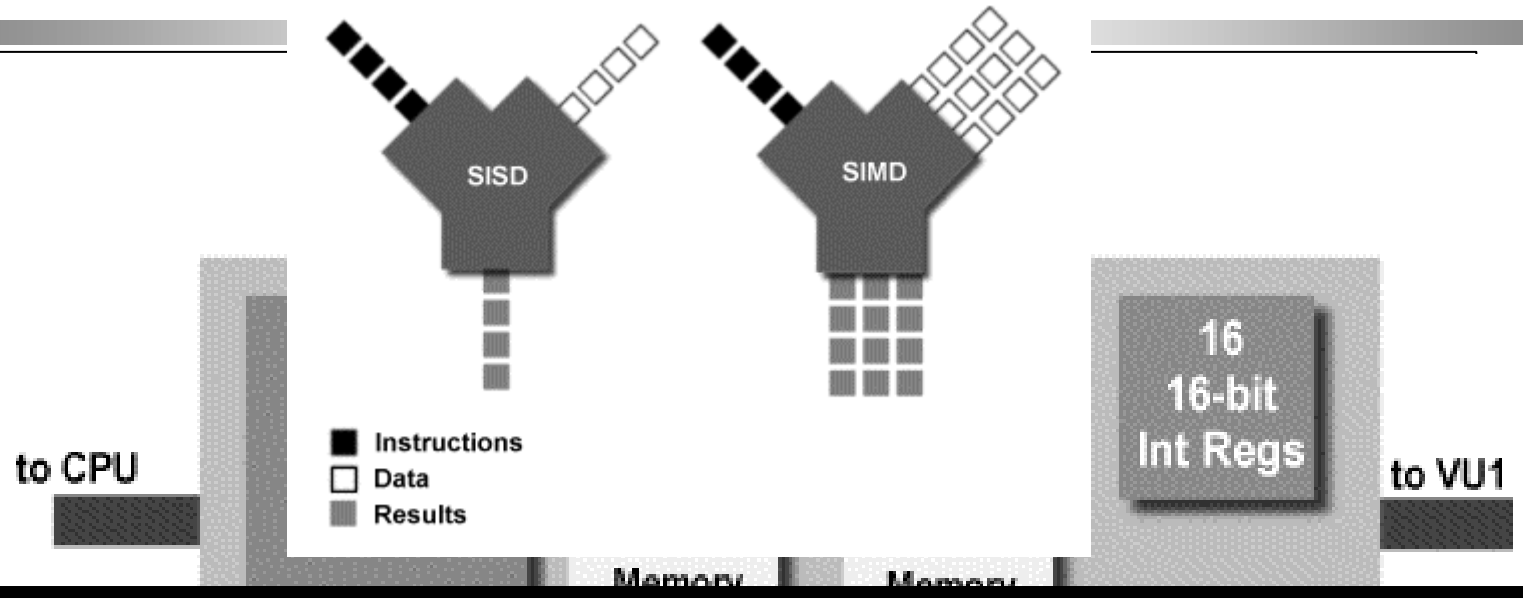
Vector Unit 0 (VU0)



- 4 FMACs, 1 FDIV
- Connected to the CPU, executing macroinstructions
- 4 KB VUMem (data), 4 KB MicroMem (instructions)
- Usually used for animation and physics.

Emotion Engine architecture

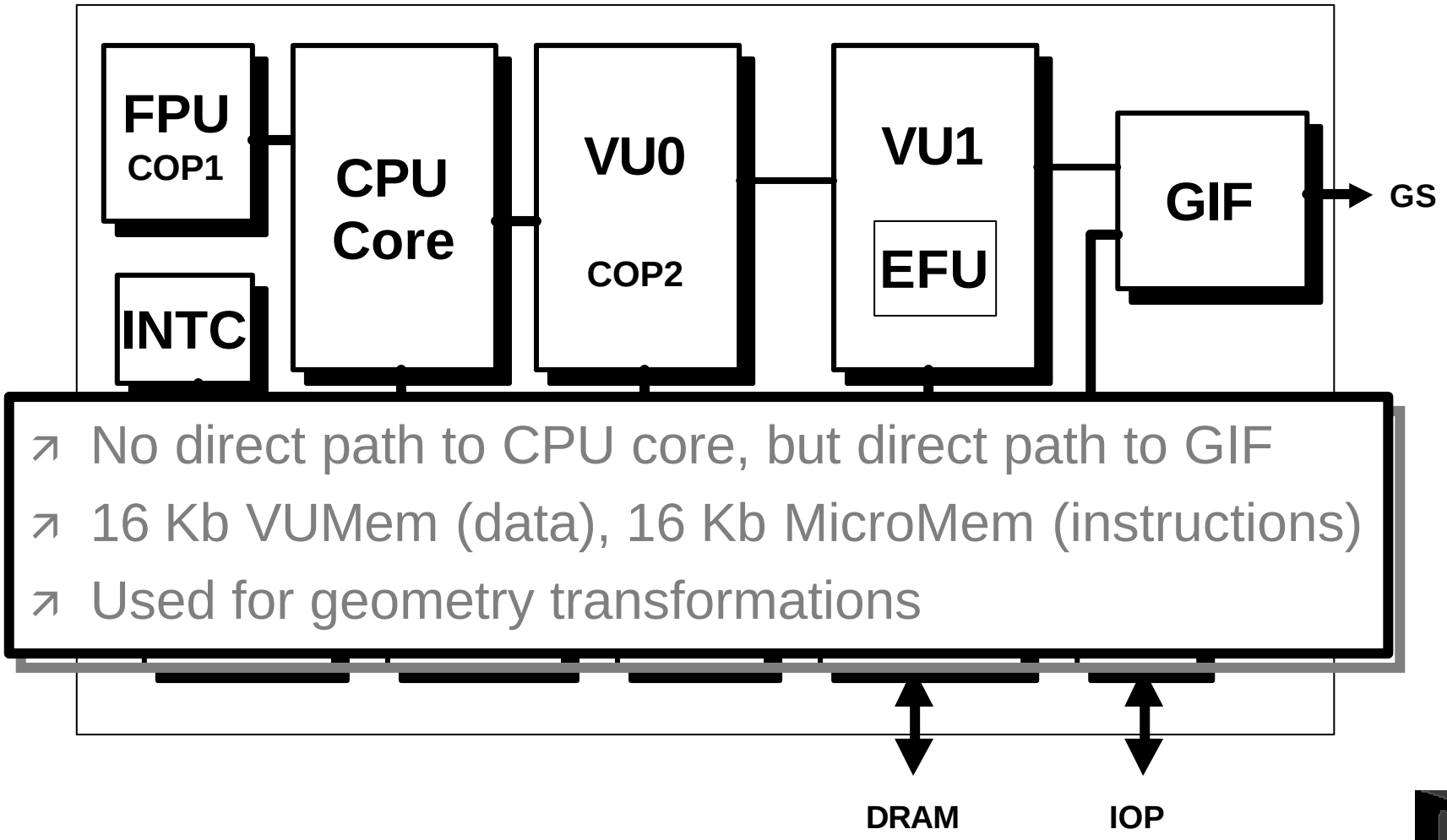
Vector Unit 0 (VU0)



- The VU0 is a 128 bit SIMD/VLIW design.
- A SIMD processor operates on vectors of data. As an example, when a SIMD instruction adds 64 bit numbers, the 64 data streams are sent to 64 ALUs to perform 64 sums in a single clock cycle

Emotion Engine architecture

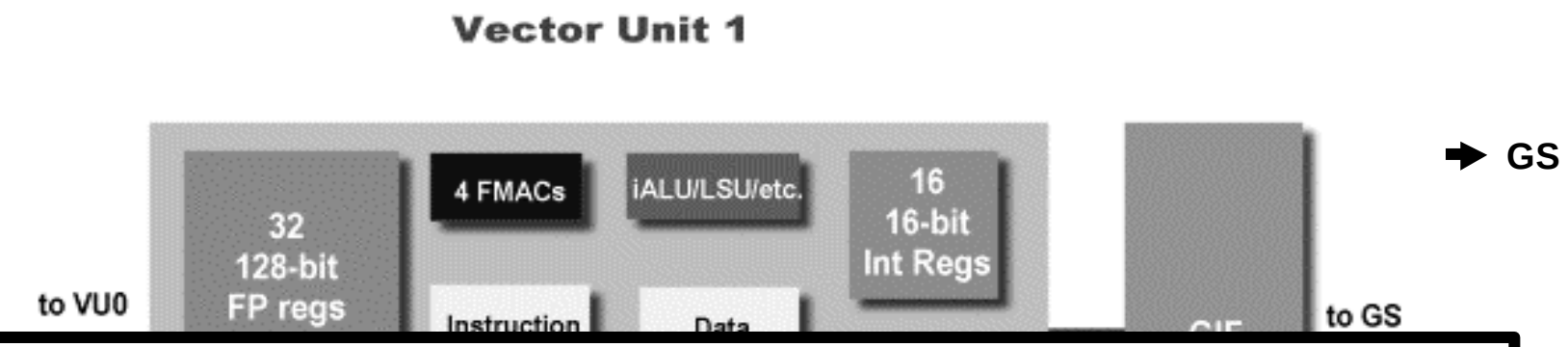
Vector Unit 1 (VU1)



- No direct path to CPU core, but direct path to GIF
- 16 Kb VUMem (data), 16 Kb MicroMem (instructions)
- Used for geometry transformations

Emotion Engine architecture

Vector Unit 1 (VU1)

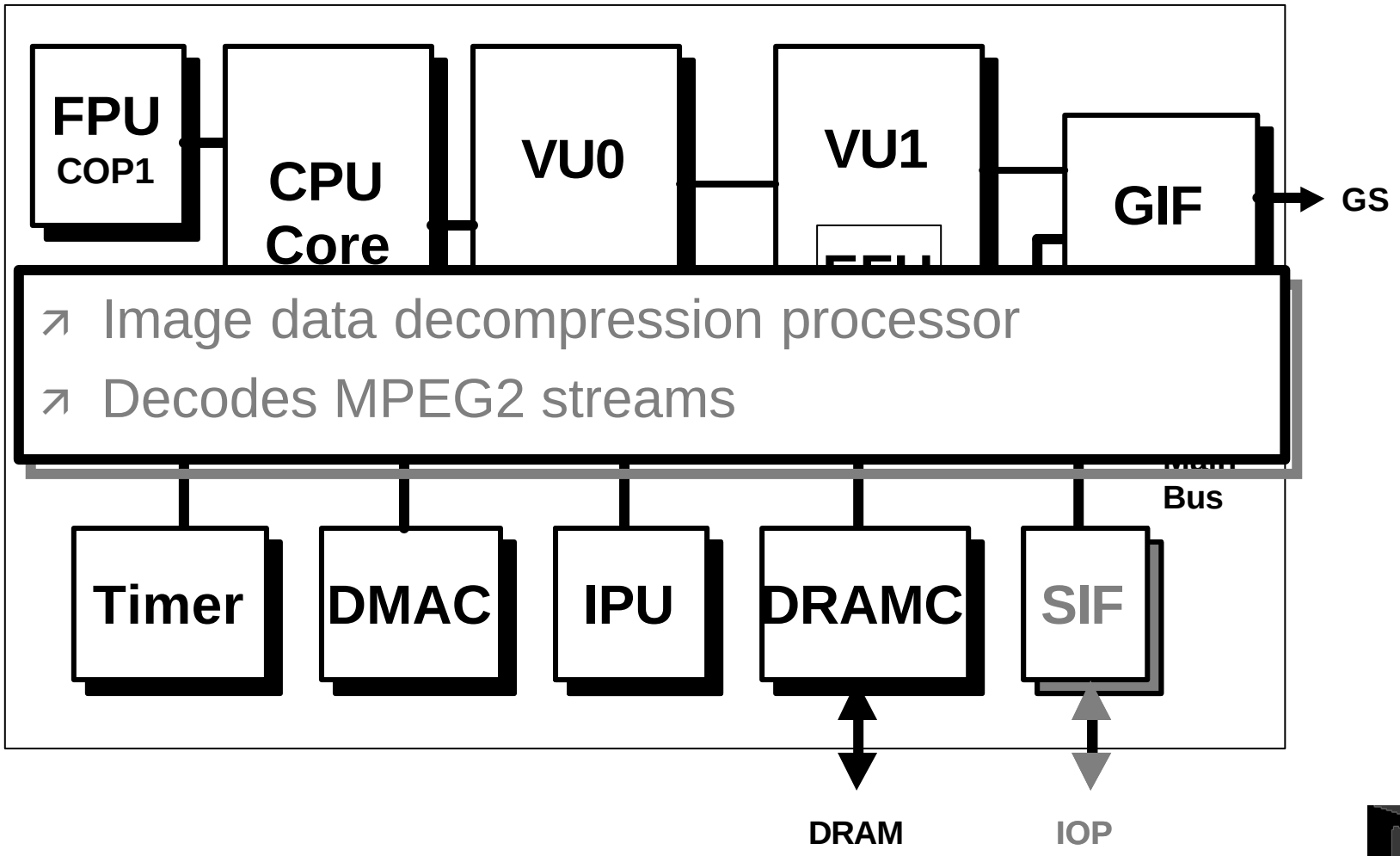


- VU1 has 16K instruction memory and data memory while VU0 has only 8K/8k of memory. This larger amount is because VU1 is a geometry processor and therefore is required to handle much more data than VU0

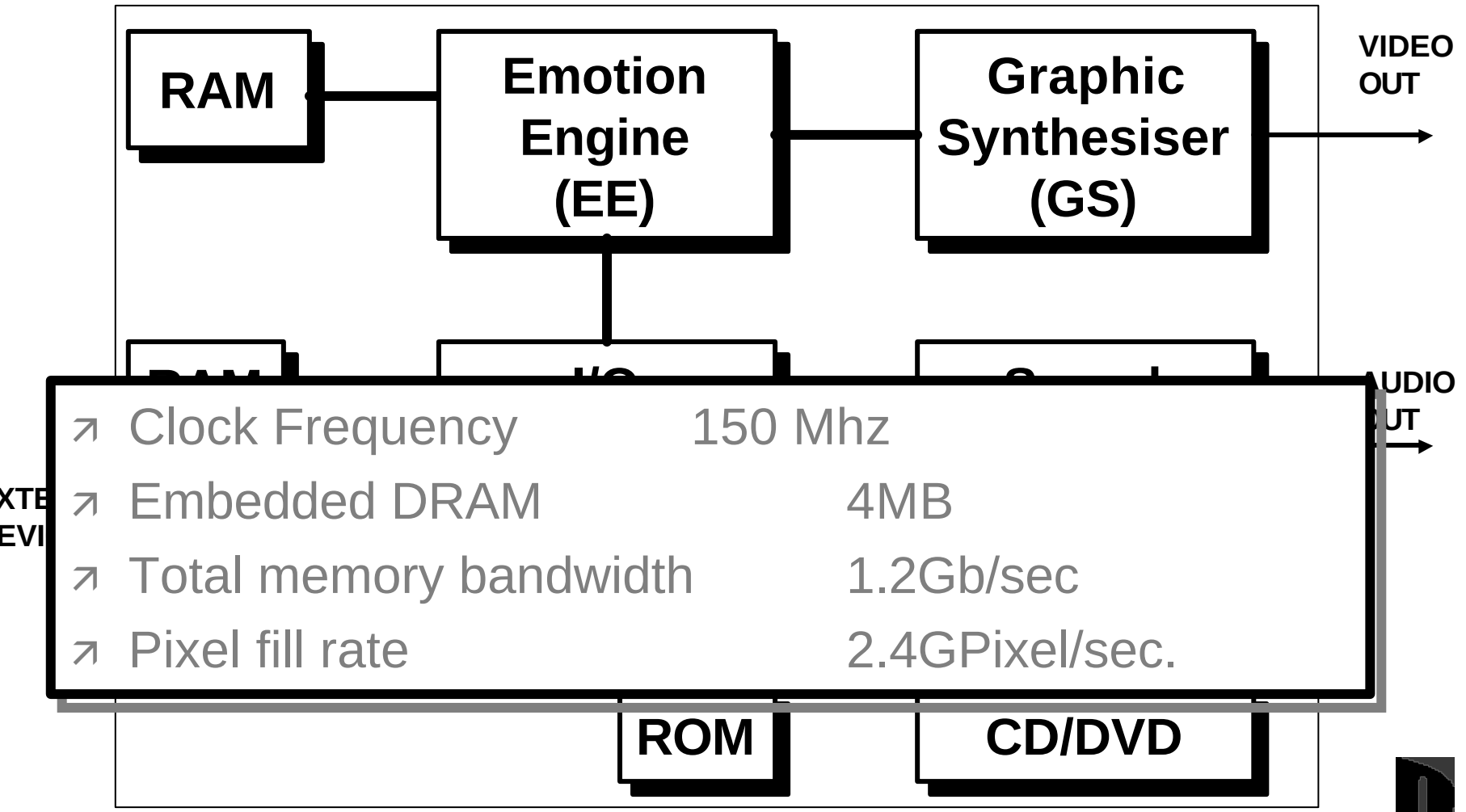
used to perform more basic calculations for geometry processing No direct path to CPU core, but direct path to GIF

Emotion Engine architecture

Image Processing Unit (IPU)



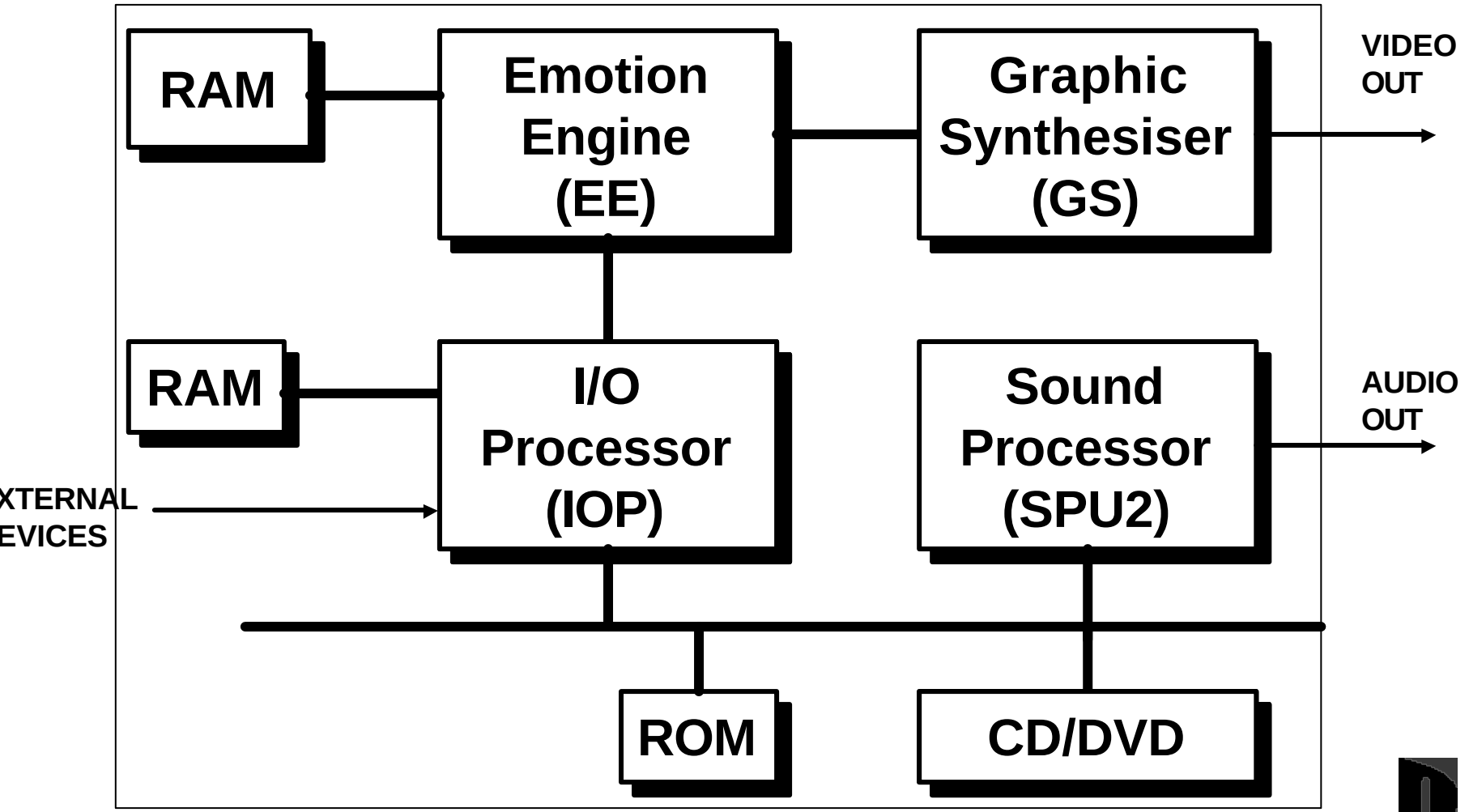
System Architecture



GS specifications

↗ Clock Frequency	150 Mhz
↗ Embedded DRAM	4MB
↗ Total memory bandwidth	1.2Gb/sec
↗ Pixel fill rate	2.4GPixel/sec.

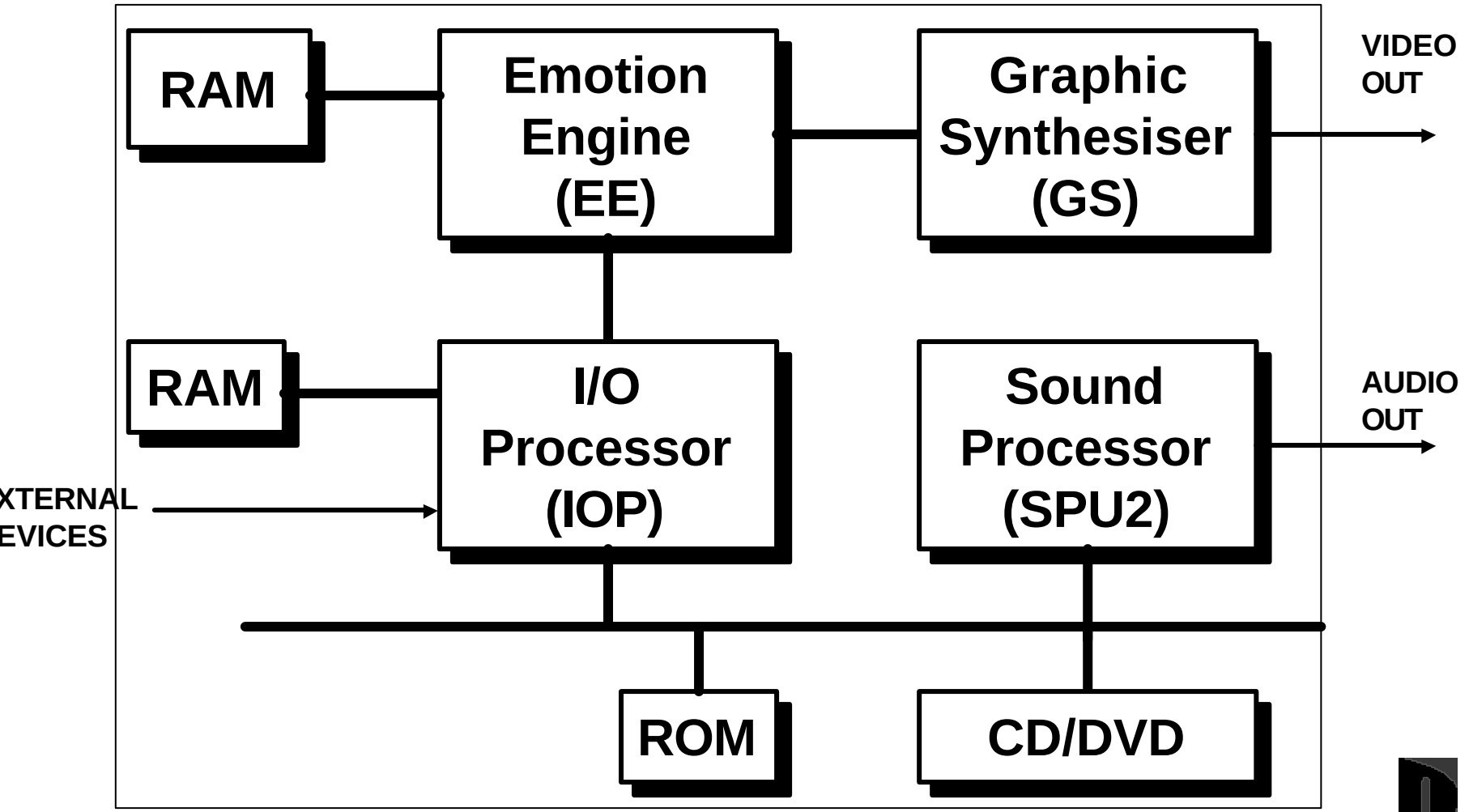
System Architecture



IOP (Input/Output Processor)

- Contains an R3000 (PlayStation CPU+)
- Used for backwards compatibility
- 2 MB of RAM
- Handles all external devices
  - Controllers
  - USB
  - SPU 2
  - CD/DVD unit
  - IEEE1394
  - Hard disc, ethernet/modem.

System Architecture



SPU 2

- 48 Channels
- 2MB sound memory
- Output to DAC or Optical digital output (Dolby 5.1)
  - Realtime DTS 5.1 is possible.

Coming Soon..

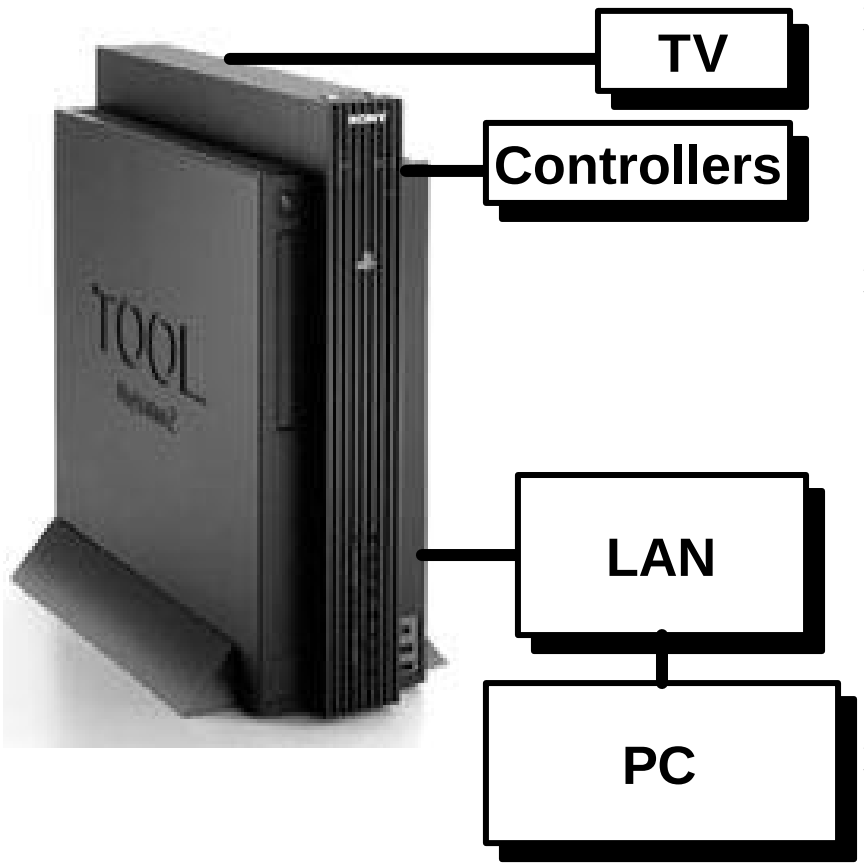
- Broadband Adaptor
  - HDD interface & 100/10 Ethernet port
  - Ethernet allows access to broadband (via ADSL/CATV/Satellite/etc)
  - HDD used by game for local storage, or downloadable content.

3) Game Development

- Programming a game on the PS2.

PS2 Development Environment

The TOOL



- TOOL = PlayStation 2 with more RAM, and network
- A separate Linux/Windows box runs the compilers and debuggers
  - Connects over the network to the TOOL.
- Use Linux-based tools (provided), or 3rd-party Windows development tools

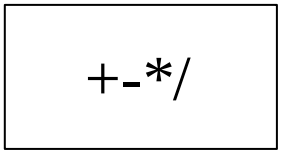
Console programming

- Halfway between embedded system and PC.
  - Small & basic OS
  - Large amount of control
- Low level coding
  - No drivers
  - Standard hardware means you can optimise for the system
  - Performance analysis has benefit.

Differences between PS2 and PC

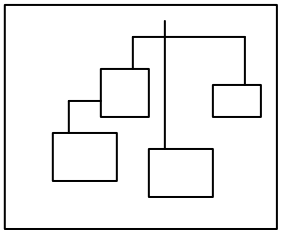
- Uses parallelism
  - Information should ‘stream’ through the system
  - But not all algorithms are parallelisable
- Random memory access hits hard
  - Data must be reorganised so that related parts sit together
- Optimisation is easier on PS2
  - Standard hardware means optimisation works on all machines.

Basic Rendering Pipeline



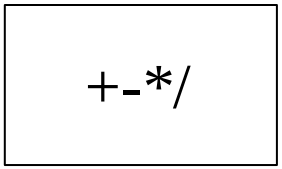
Calculate animation

CPU + coprocessor VU0



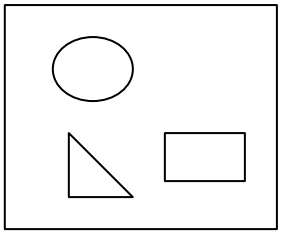
Traverse scene

List processing DMA



Transform to 2D

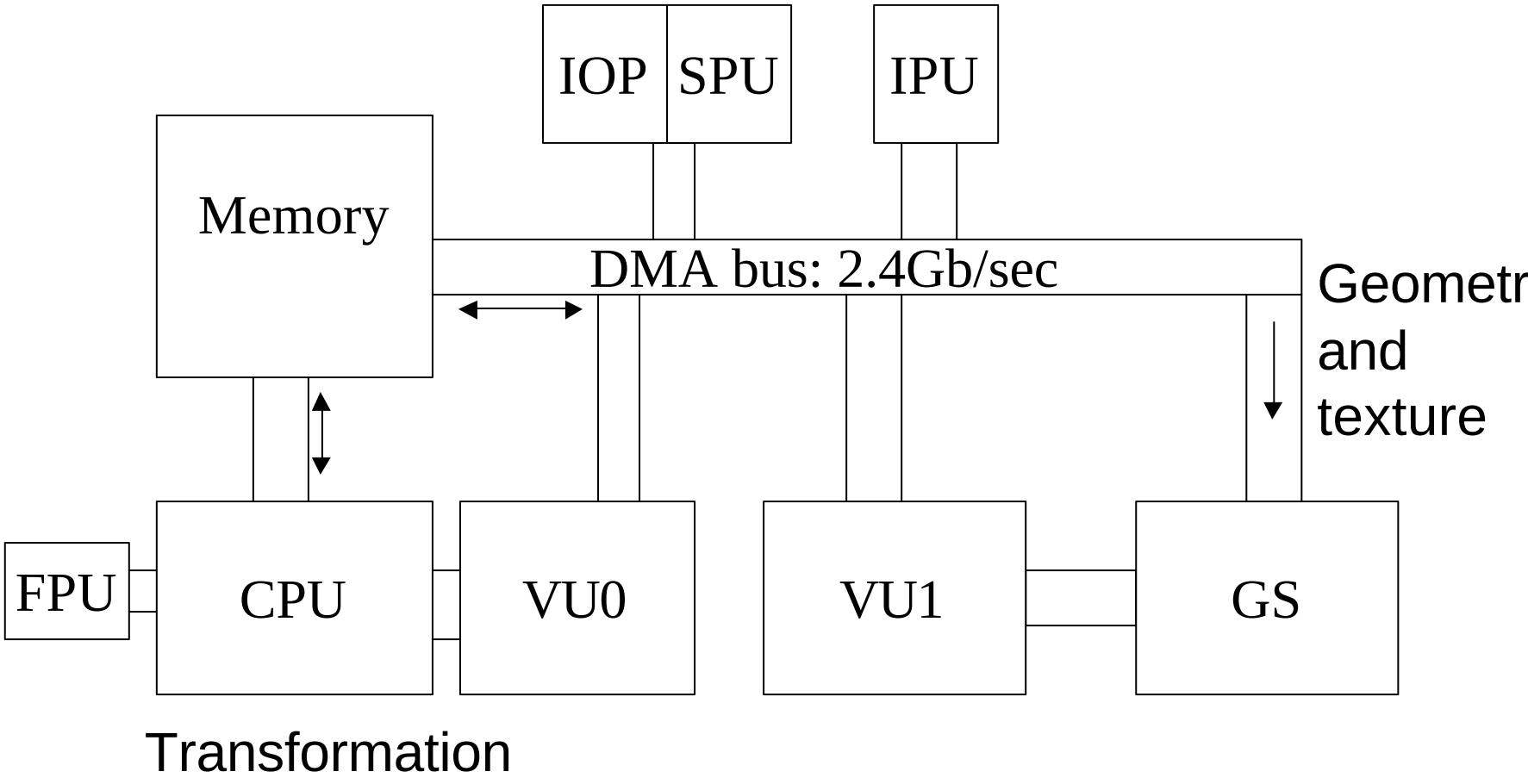
VU1



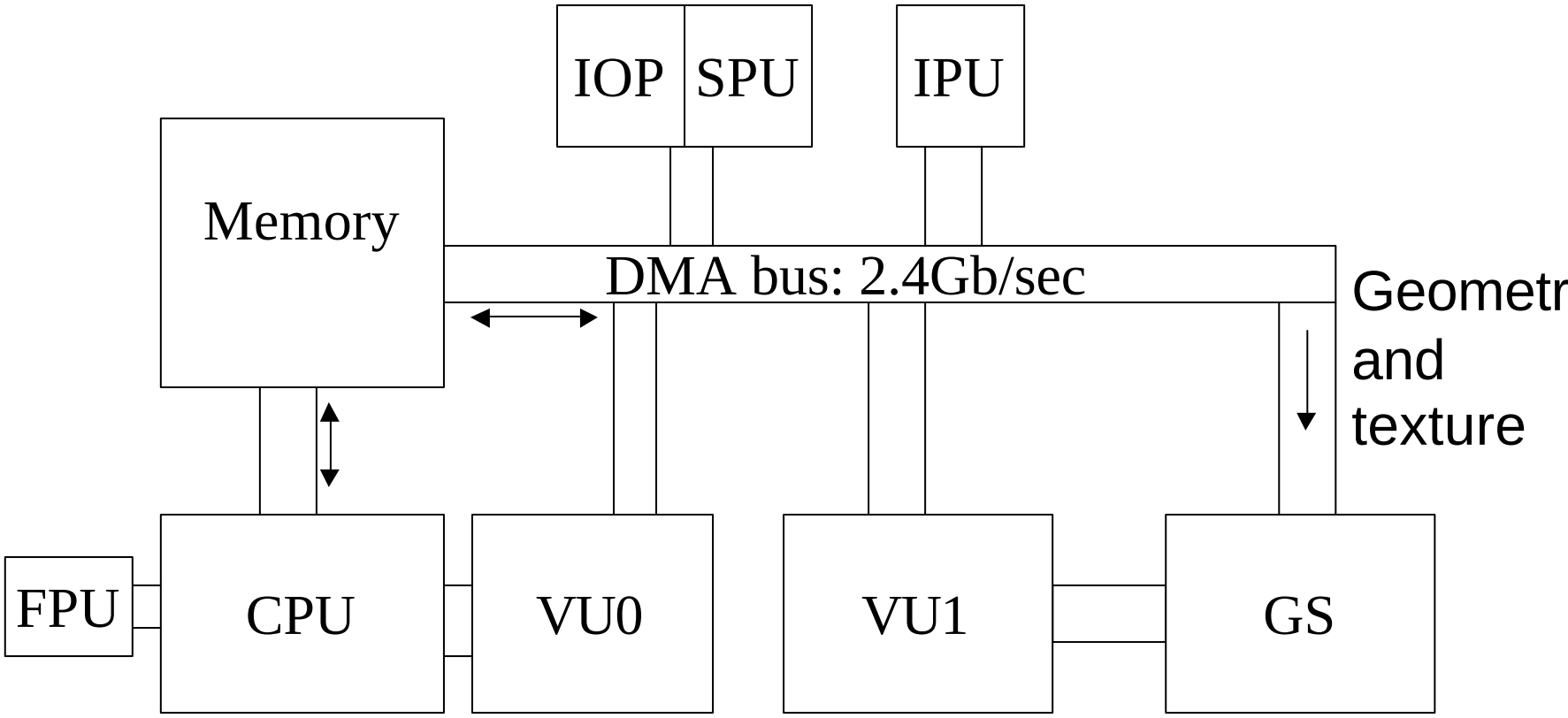
Rasterisation

GS

1st Attempt At A PC Port (max 0.5 million polys)

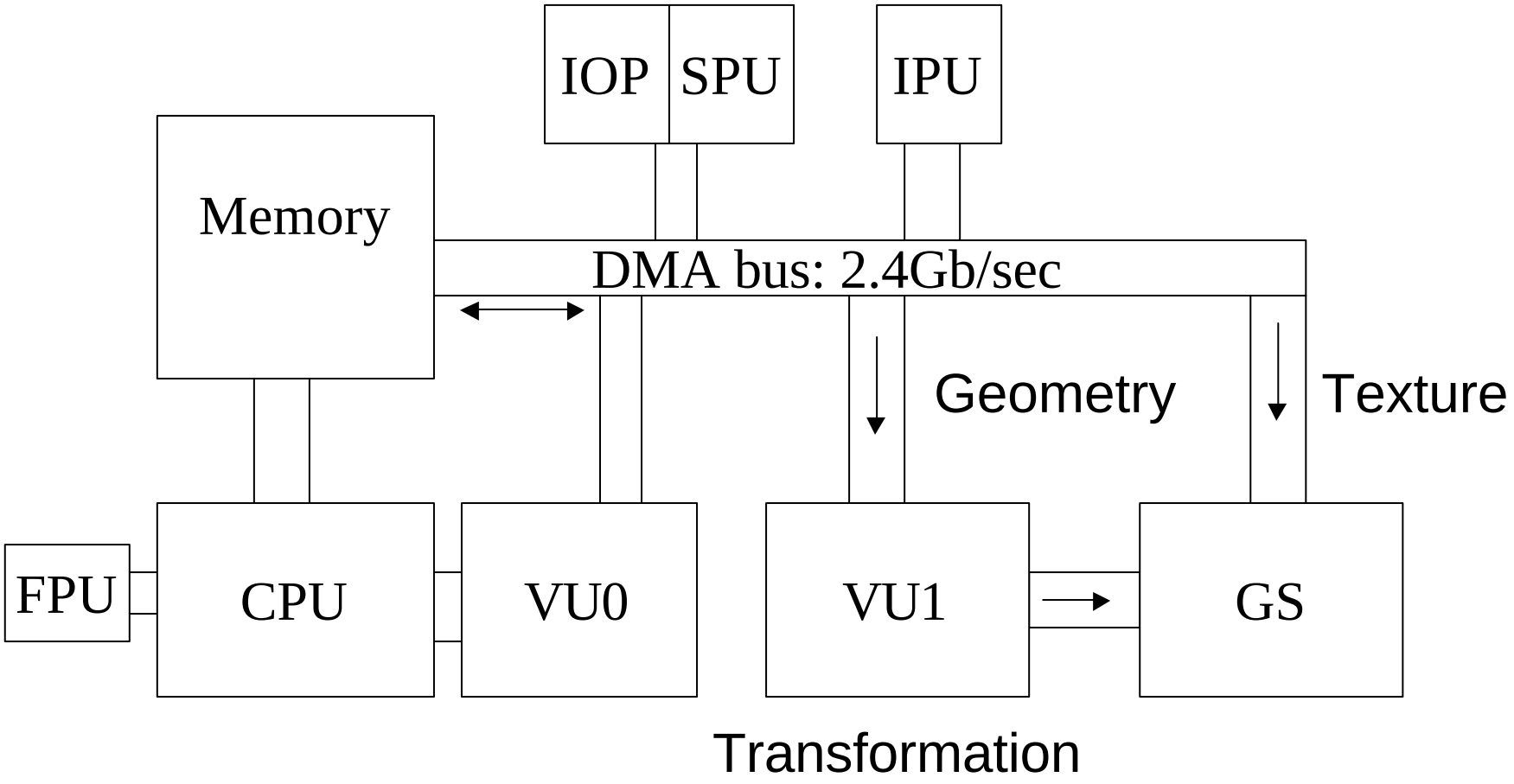


2nd Attempt At A PC Port (max 1.5 million polys)



Transformation
in parallel with CPU

Complete Game (lighting, animation) (typical 5-10 million polys)



How To Improve PS2 Performance

- By not treating the PS2 as a PC
- Think parallel – think ‘assembly line’
- Code for small Instruction and Data Cache

Summary

- PS2 is a state-of-the-art machine
- Achieves high performance and low cost through high parallelism
- But it requires a different way of programming
- Question Time!