53 GOPS Programmable Vision Processor for Processing, Coding-Decoding and Synthesizing of Images

Ulrich Ramacher, Wolfgang Raab, Nico Bruels, Ulrich Hachmann, Christian Sauer, Alex Schackow, Jörg Gliese, Jens Harnisch, Mathias Richter, Elisabeth Sicheneder
Corporate Research, Infineon Technologies AG

Rene Schueffny
University of Technology Dresden, Germany

Uwe Schulze, Hendrik Feldkaemper, Christian Luetkemeyer
University of Technology RWTH Aachen, Germany

Horst Suesse, Sven Altmann
Fraunhofergesellschaft IIS-EAS Dresden, Germany

The Electronic Eye Project

Mass market applications
- car vision systems
- advanced videoconferencing
- dialogue modems

Deficiencies of today‘s Vision Systems
- no lack of applications
- lack of real-time image architectures and robust algorithms
- lack of cheap high-performance accelerators

Design of a vision system for a large variety of indoor/outdoor applications
- conventional and „neural“ algorithms
- modular and scaleable system architecture
- CMOS image sensor + Vision Instruction Processor
- C/C++ development environment

Architectural strategy

Programming language

- C with extensions for vector and matrix data types
- specific instructions for processing, coding-decoding and graphics

High-performance vision processing

- programmable SIMD processing array
- caches adapted to new data types

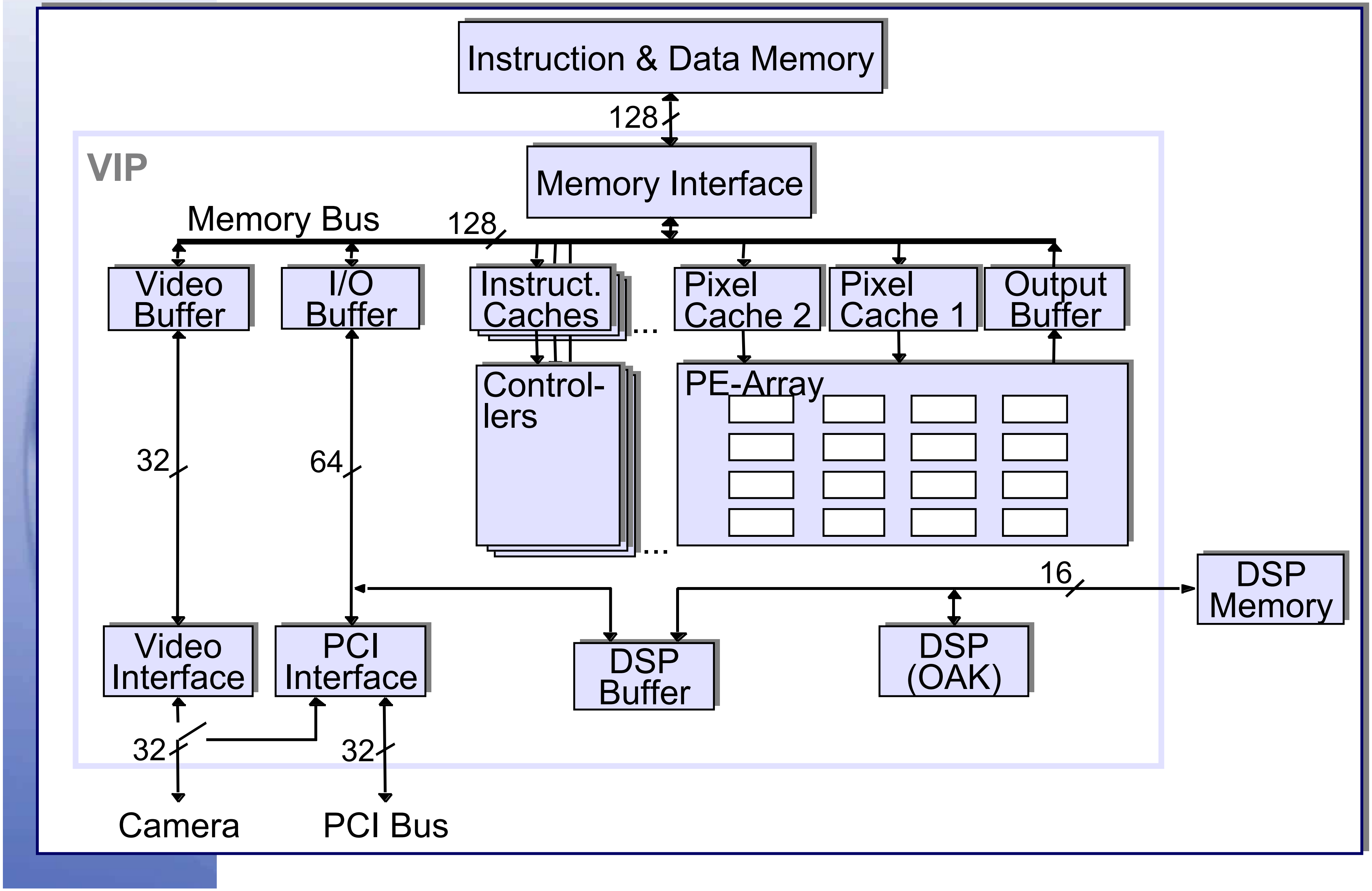
Low-Energy High-performance Control

- C-core + distributed controllers
- µ-programs

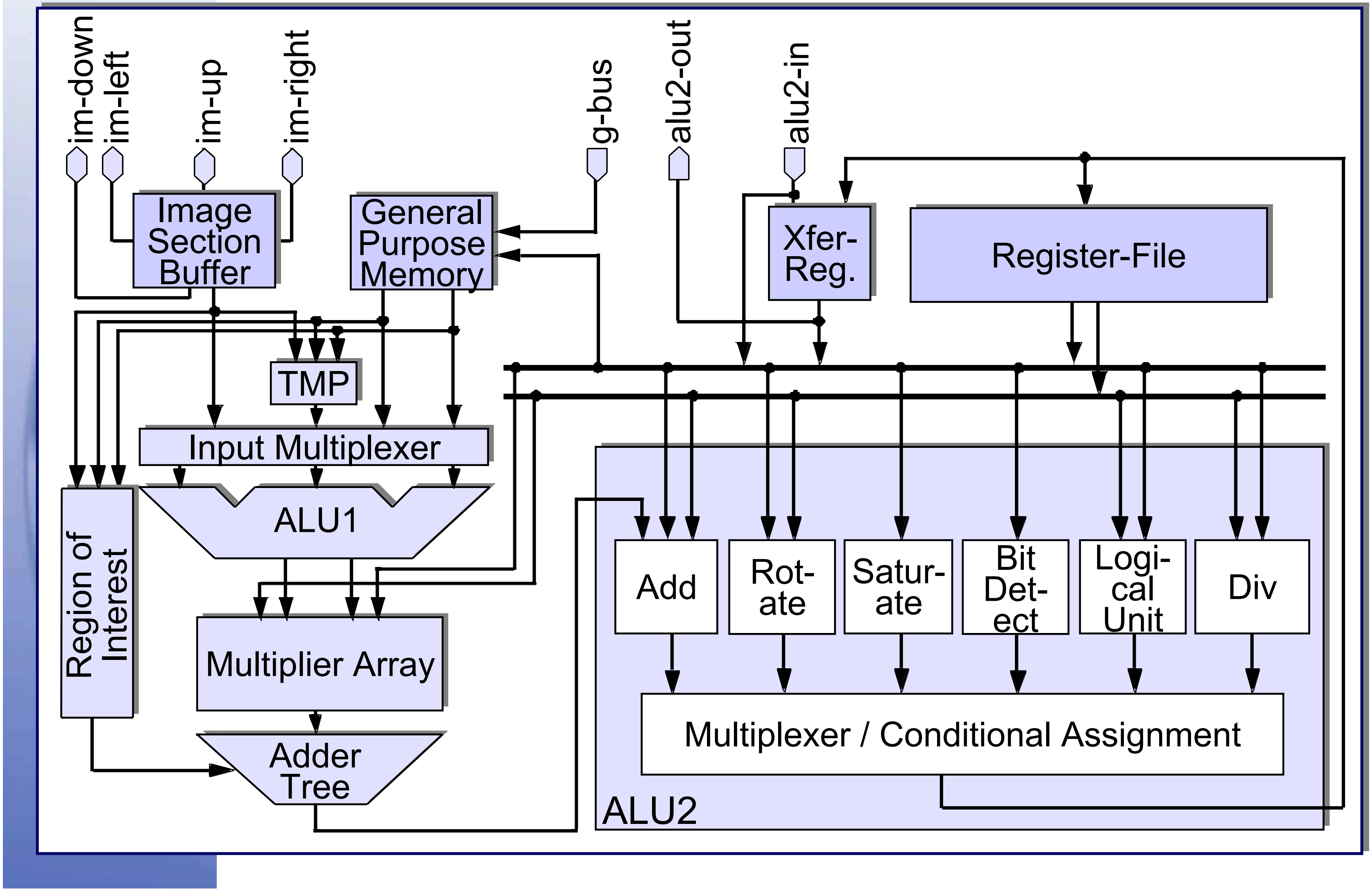
Interfaces

- PCI
- 2 digital cameras

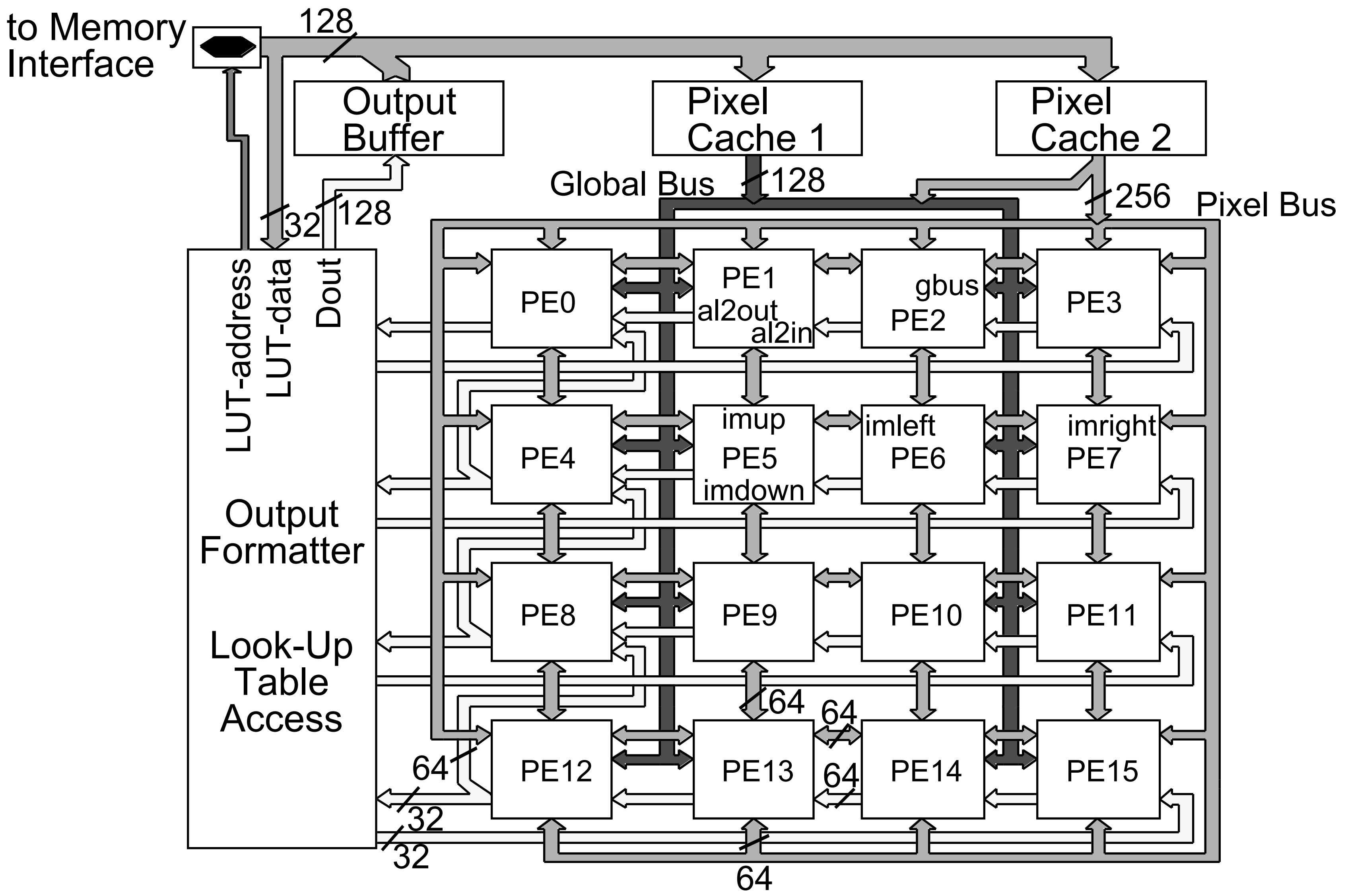
Architecture of VIP128



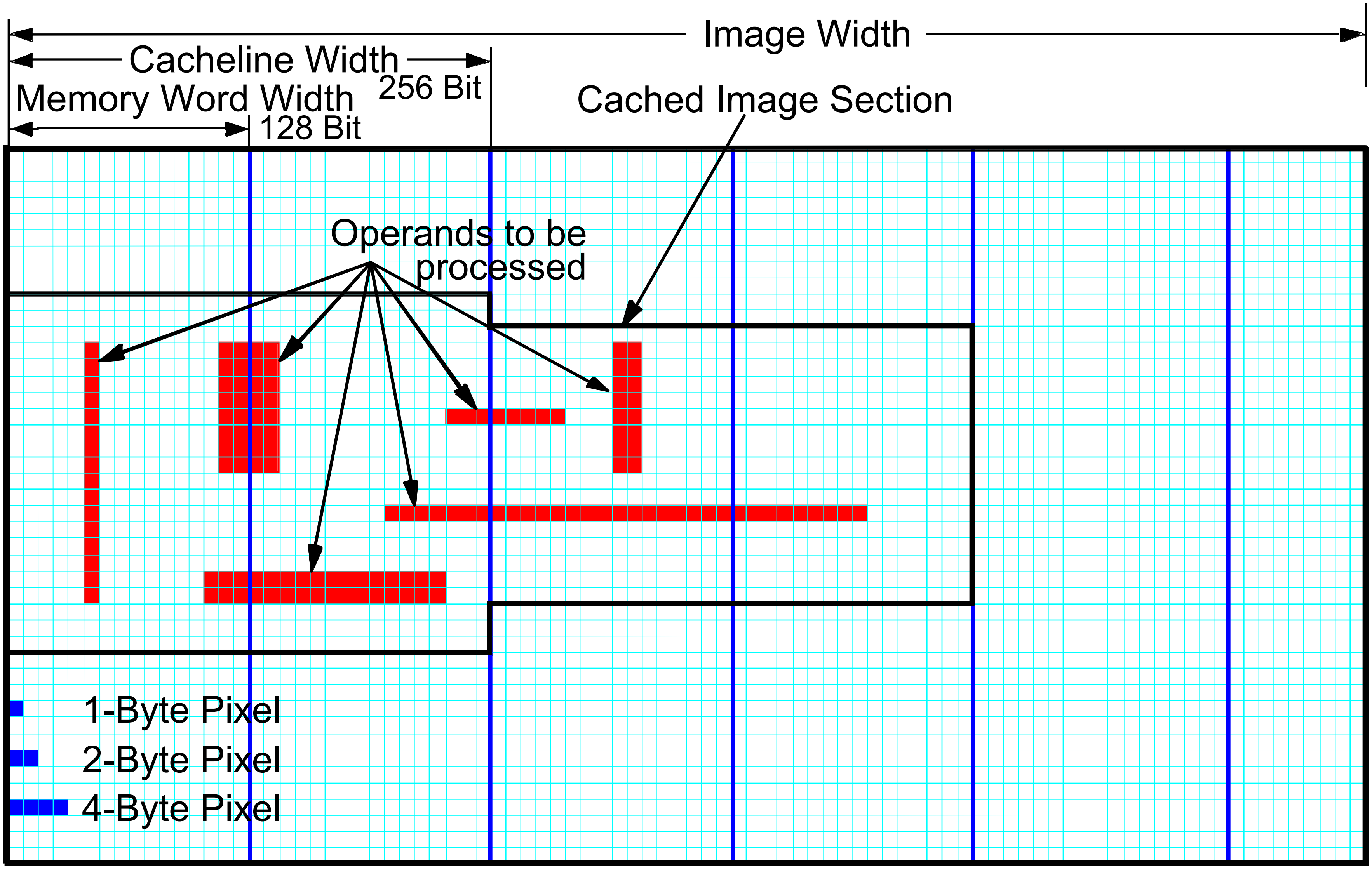
Processing Element



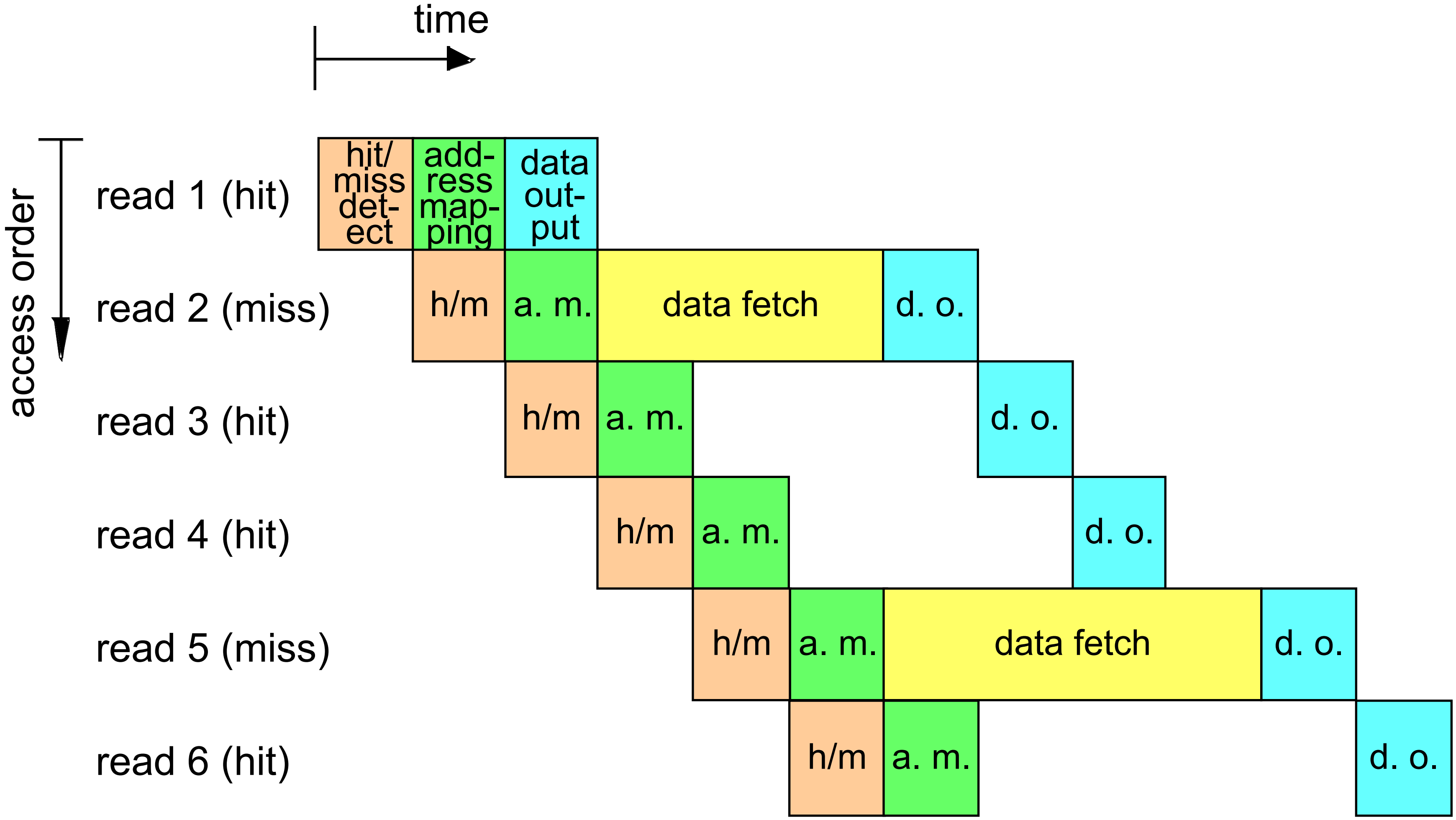
Processor Array



Data Formats of the Pixel Caches



Pipeline Operation of the Pixel Cache



time

access order

read 1 (hit) hit/ miss det- ect add- ress map- ping data out- put

read 2 (miss) h/m a. m. data fetch d. o.

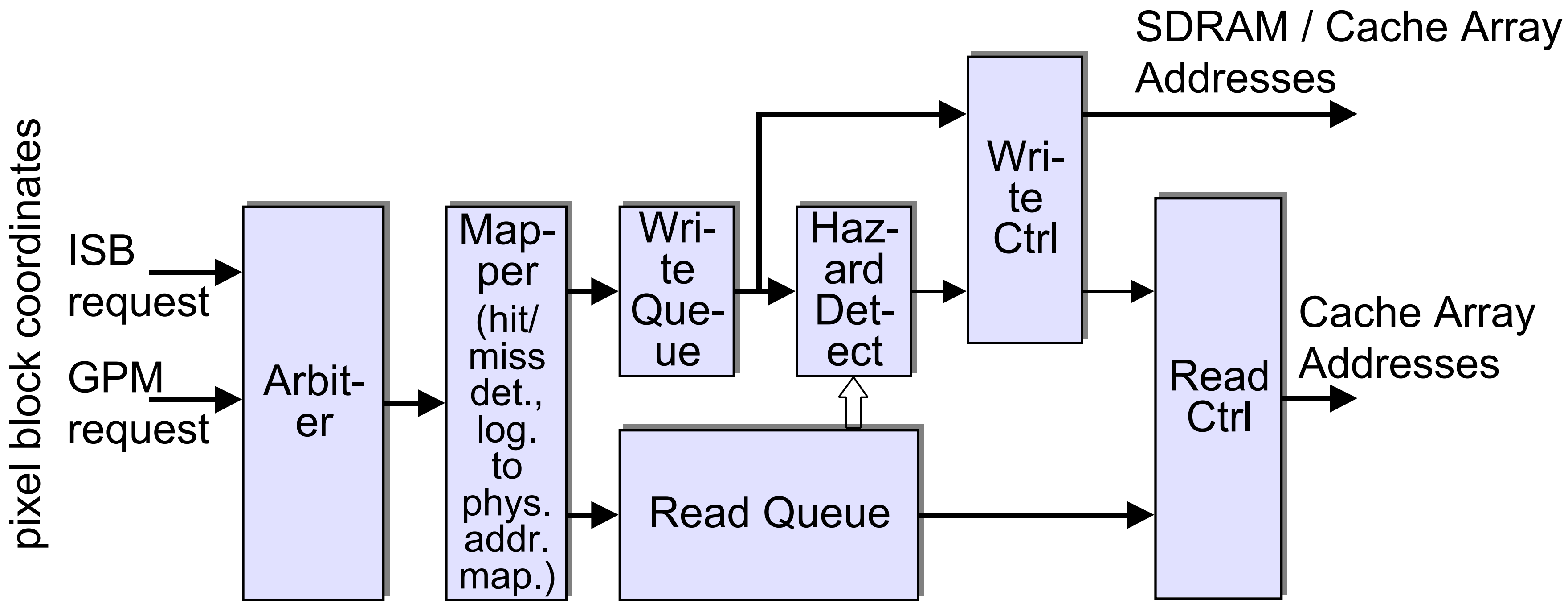
read 3 (hit) h/m a. m. d. o.

read 4 (hit) h/m a. m. d. o.

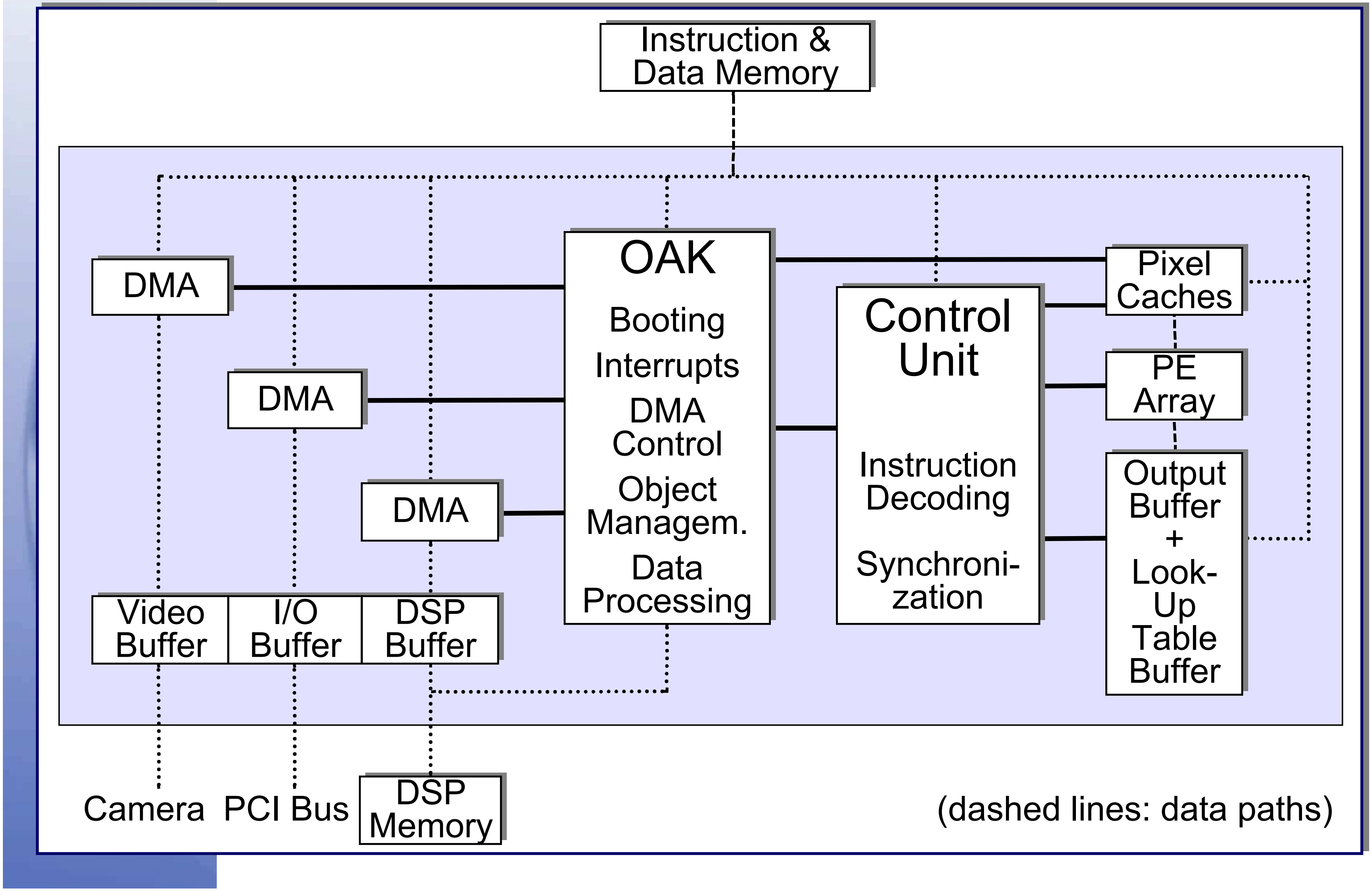
read 5 (miss) h/m a. m. data fetch d. o.

read 6 (hit) h/m a. m. d. o.

Controller of the Pixel Cache

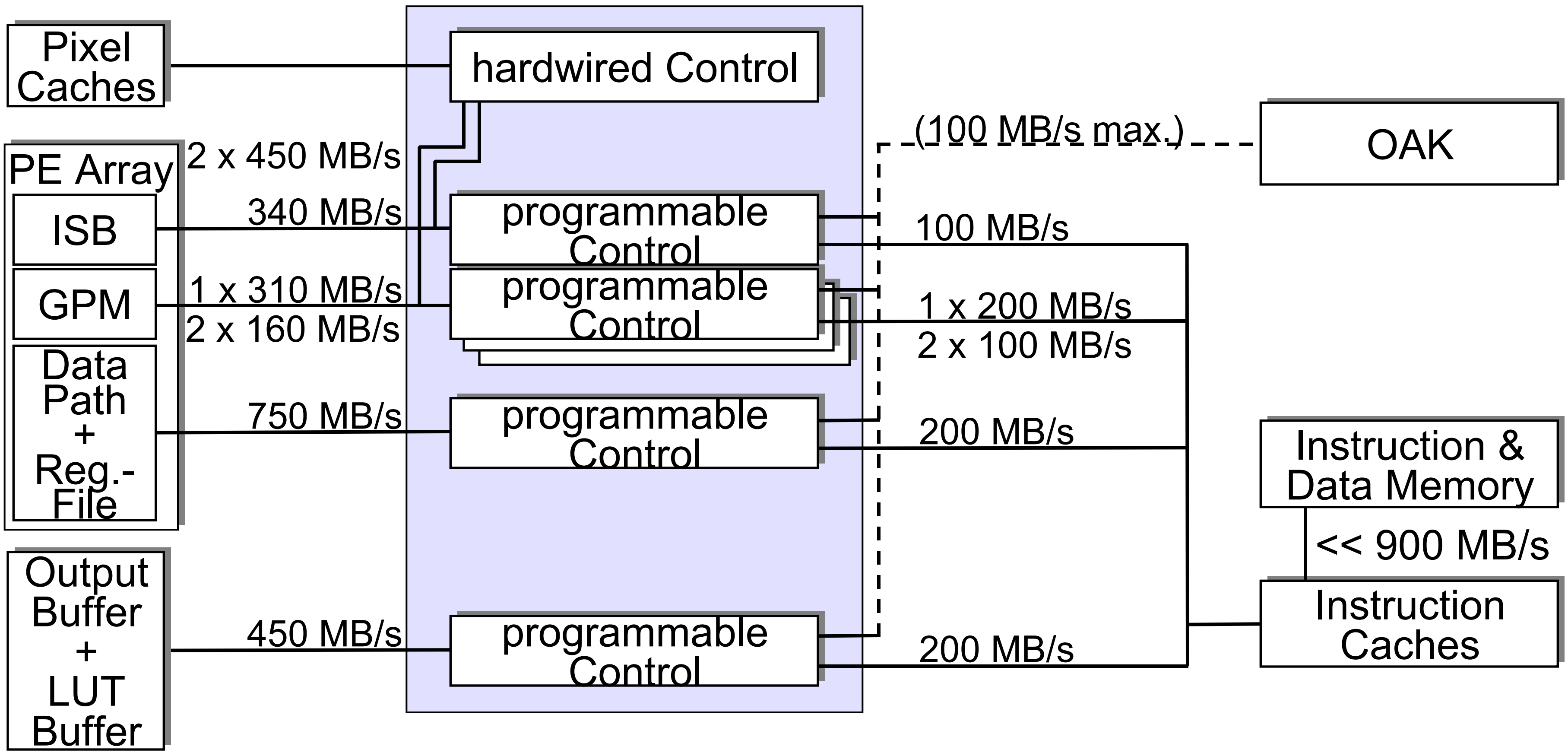


Control Paths of the VIP128



(dashed lines: data paths)

Distributed Controllers



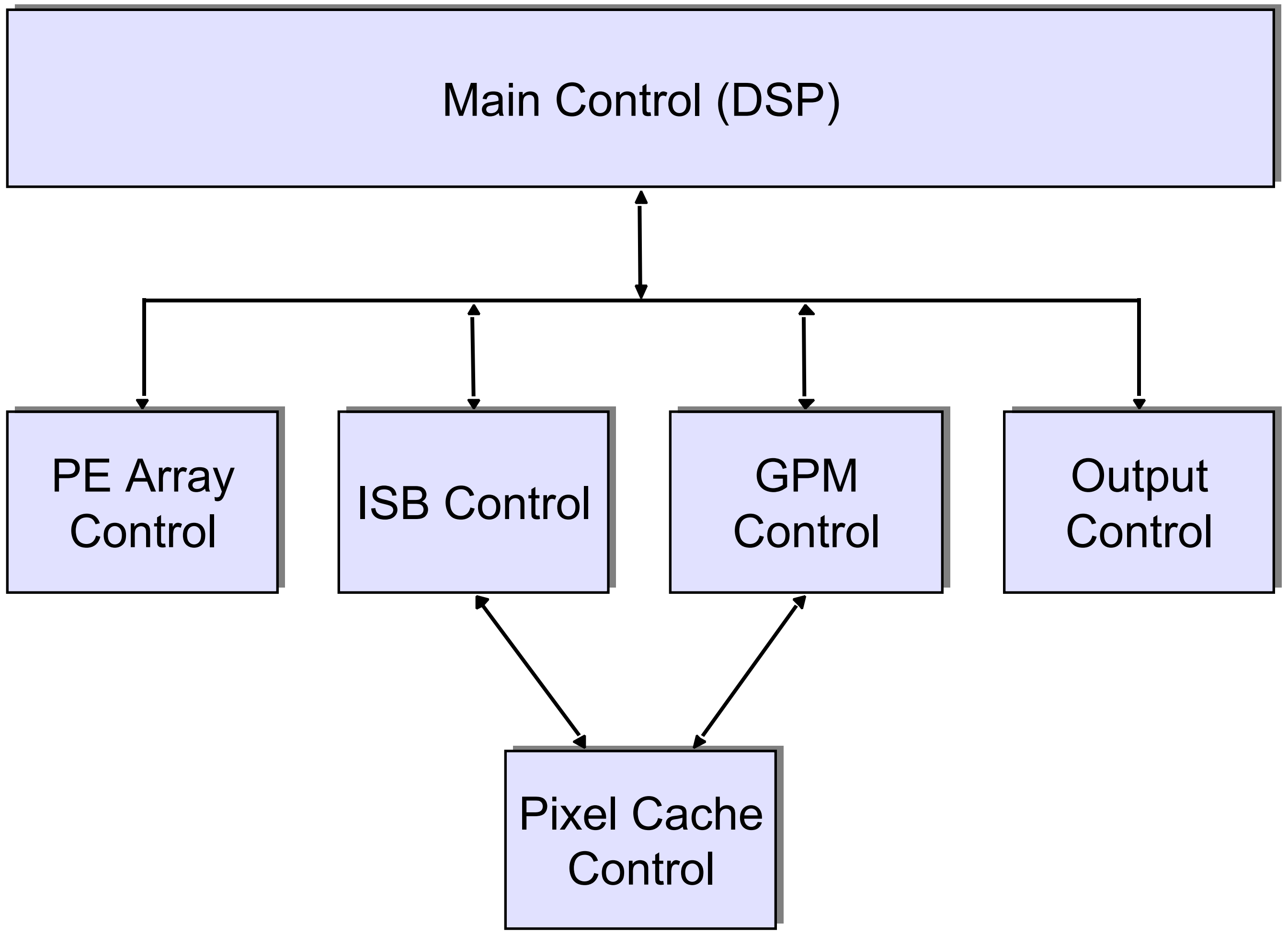
Σ 3070 MB/s

Bandwidth reduction by

Σ 900 MB/s

- * operation coding (instructions)
- * program loops + instruction caches

Control Hierarchy



C program

micro-programs

hardwired

µProgrammed Controller

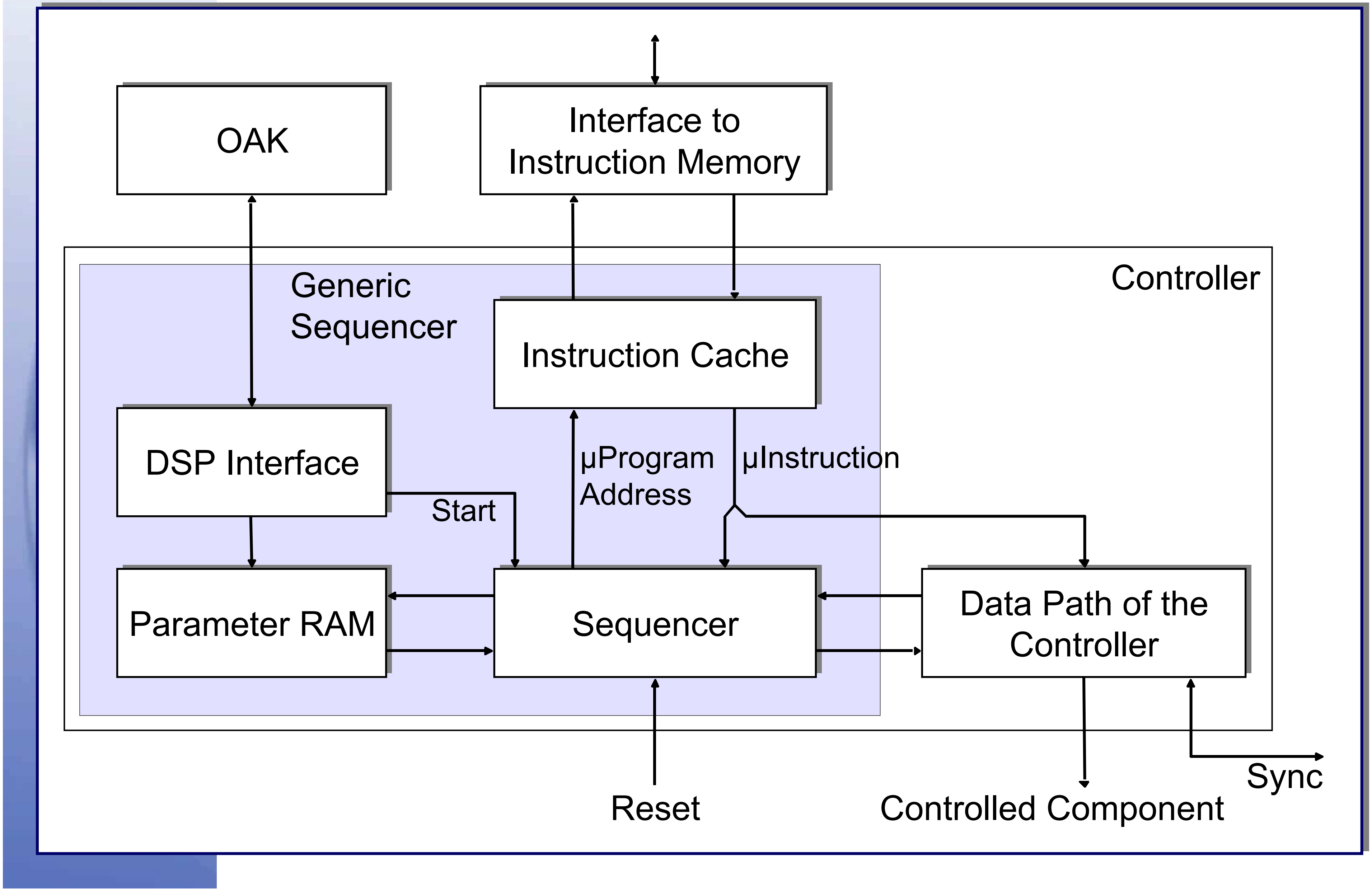


Photo of the VIP128

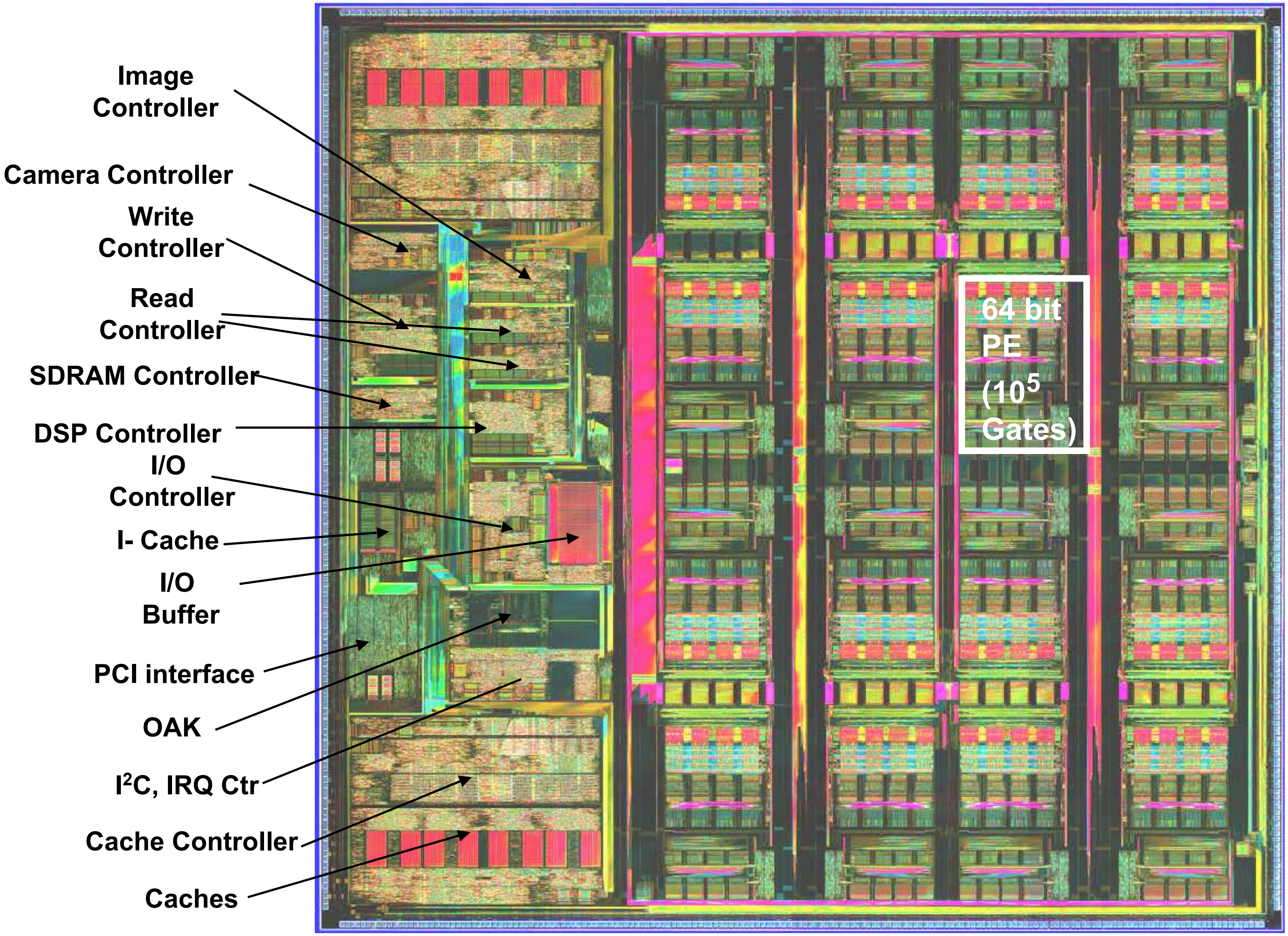


Image Controller

Camera Controller

Write Controller

Read Controller

SDRAM Controller

DSP Controller

I/O Controller

I- Cache

I/O Buffer

PCI interface

OAK

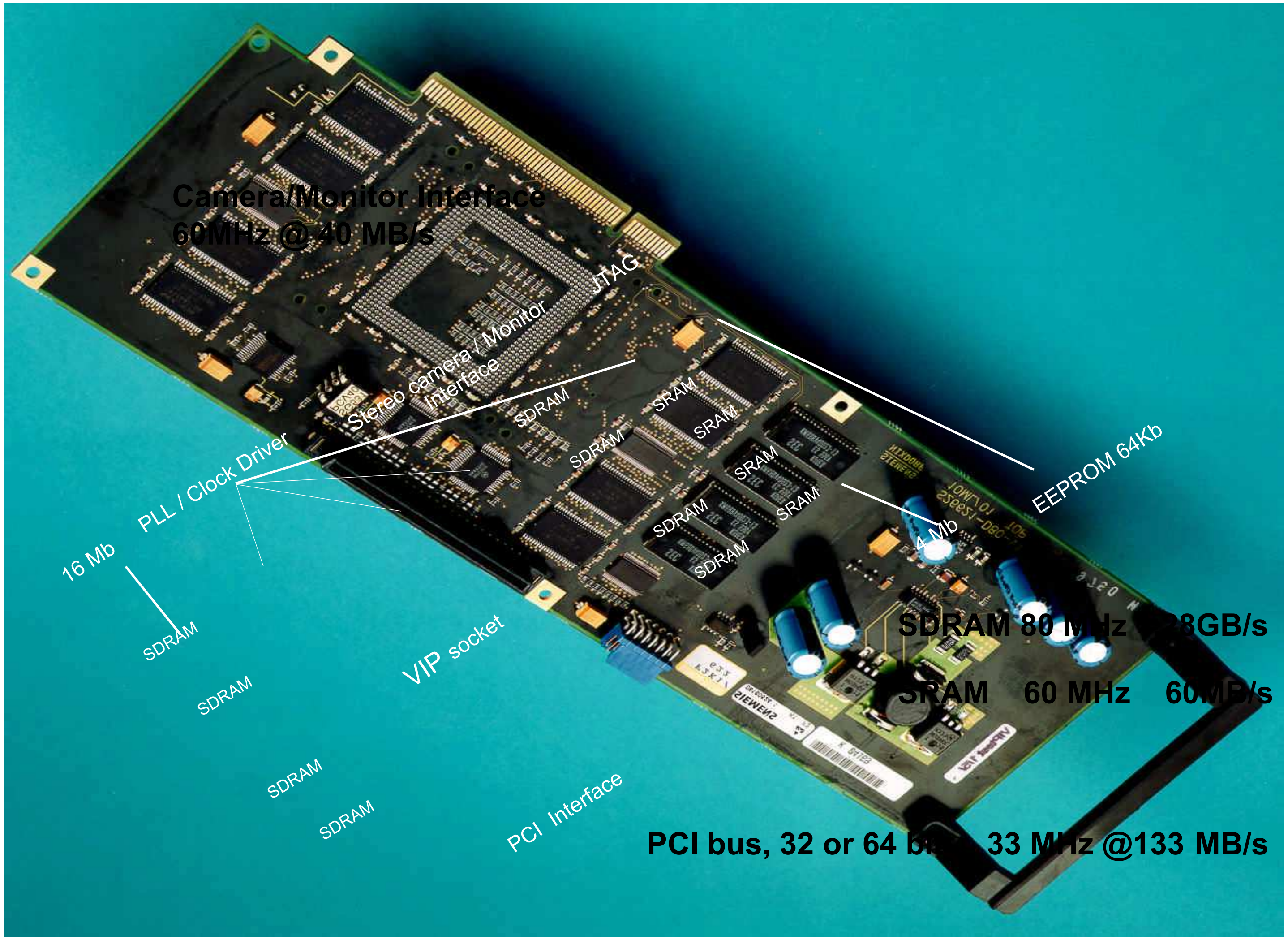
I^{2}C, IRQ Ctr

Cache Controller

Caches

64 bit PE (10^5 Gates)

PC Board for VIP 128



Characteristic Data

Technology	0.35 µm CMOS
Clock frequency	100 MHz
Total power consumption	8 Watts at 3.3 V
Total area	22 x 23 mm^2
Number of transistors	12 Mio. (2 Mio. for memory incl.)
Number of PEs	16
PE transistors / area	450 000 / 12 mm^2
On-chip memory	37 KByte
Number of data caches	2 (2 KByte each)
SDRAM / SRAM bandwidth	10 Gbit/s at 80 MHz / 267 Mbit/s at 50 MHz
PCI bandw. (32 bit / 64 bit)	1 Gbit/s / 2 Gbit/s at 33 MHz
Data types and width	Scalar, vector, matrix / 1, 2, 4 Byte
Peak performance	12.8 x 10^9 Macc/s (8x16 bit) 6.4 x 10^9 Macc/s (8x32 bit, 16x16 bit) 3.2 x 10^9 Macc/s (16x32 bit) 1.6 x 10^9 Macc/s (32x32 bit)
Number of PE instructions	147 for ALU2, 57 for ALU1
DSP & controller performance	350 MIPS (produce 24 x 10^9 control bits / s)

User Interface

Objectives:

- **special data types and high level operations, adapted for image processing**
- **object oriented interface**
- **encapsulation of all hardware details without loss of performance**
- **easy porting of available applications from C++ to VPL**
- **reasonable costs for development of VPL environment**

User Interface

Solution:

- compiler frontend based on ELI compiler construction kit
- VPL syntax: ANSI C plus extensions, VPL is a subset of C++
- native high level data types: scalar, vector, matrix, lookup table
- reference classes for handling cutouts of images
- classes for camera control and communication
- numerics of data types: block floating point with 1, 2 or 4 byte mantissa

VPL Example Program

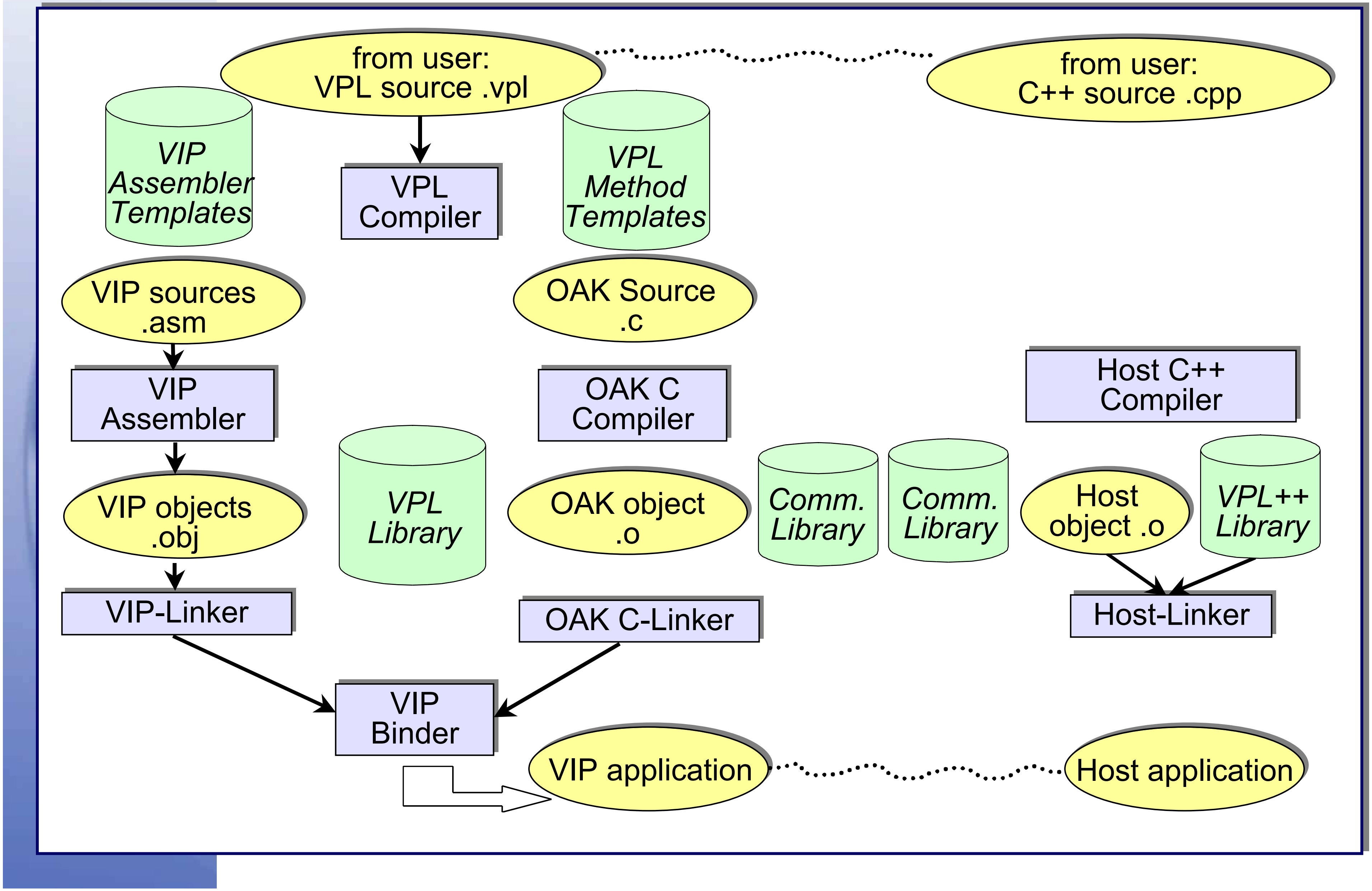
```
int main(void)
{
    VPLMatrix1U source; VPLMatrix2SE kernel; VPLMatrix4SE target;
    VPLConvScanParam convParams;

    // allocation of image memory, setting of exponent
    source.create(128, 128);
    kernel.create(9, 9, -10);
    target.create(128, 128, -8);

    // next could follow initialization of source, kernel
    // and convolution parameters, but has been skipped here

    // convolution of source with kernel
    target.conv(source, kernel, &convParams);
    return 0;
}
```

VPL Environment (Automatic Flow)



Prototype Applications

- Face Recognition:
  - downsampling, gabor transform, normalization and matching on PE array, other parts on OAK
- Overtake Checker:
  - gaussian filtering, gradient, thinning, edge linking, distance transformation and dilatation on PE array, other parts on OAK
- MPEG2 Encoder (in development):
  - DCT, motion vector estimation and quantization on PE array, other parts on OAK
- MPEG2 Decoder (in development):
  - dequantization, IDCT and adding motion vectors on PE array, other parts on OAK
- Computer Graphics:
  - Flat and Gouraud Shading of OpenGL on PE array, other parts on host

VIP System: Benchmarks

Application	Description	Performance

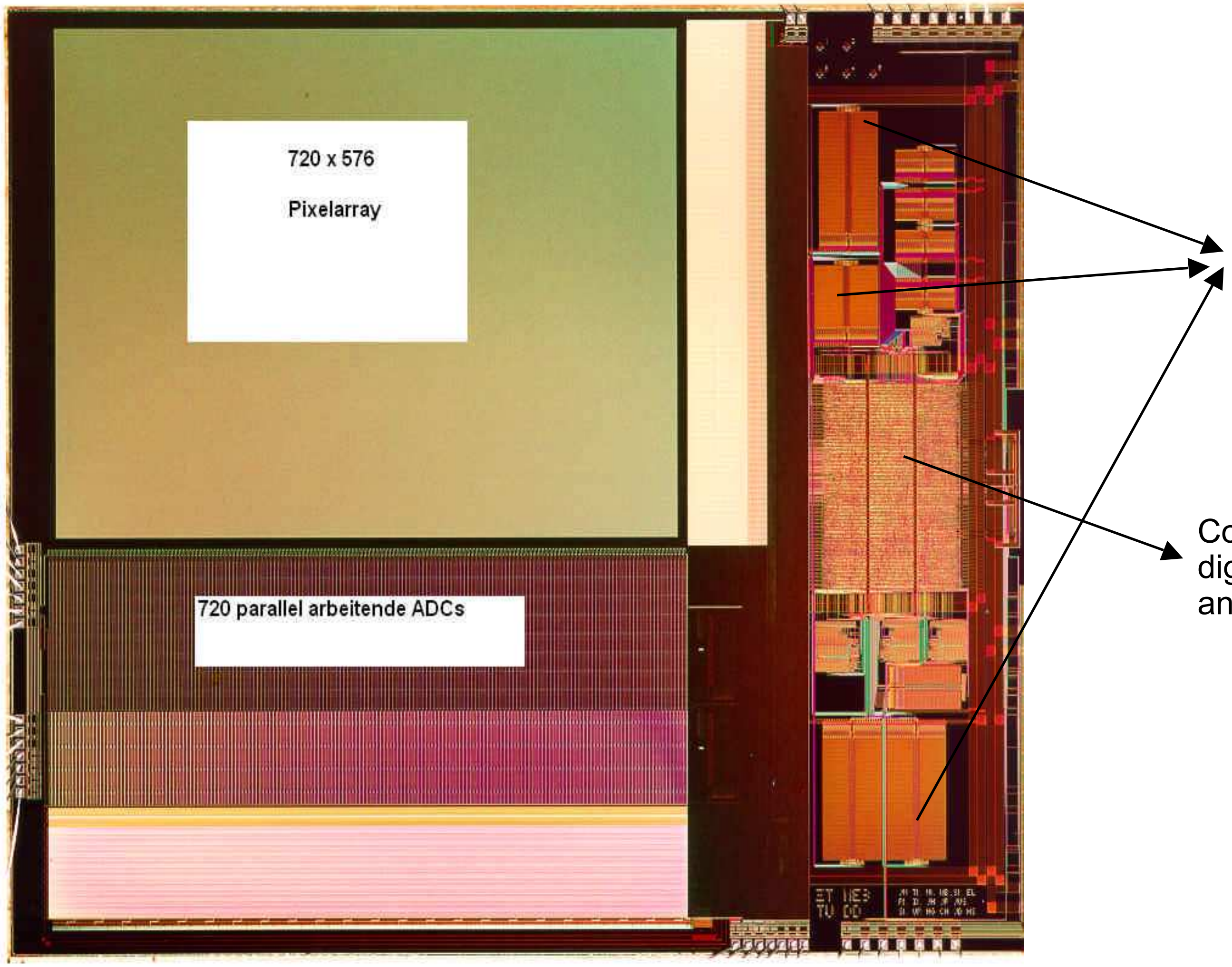
Convolution	7 x 7 inseparable kernel;
	512 x 512 image;
	16-bit data (pixel and kernel)
		4.1 ms @ 100 MHz

Block Motion Estimation	16 x 16 reference block;
	720 x 576 pixels/frame;
	search range 16 x 16 pixels; full search;
	minimum absolute differences or minimum square differences metric;
	8-bit data
		11.3 ms / frame @ 100 MHz

DCT/IDCT	8 x 8 blocks; 720 x 576 pixels/frame; 16-bit data	2.5 ms / frame @ 100 MHz

Gabor Filter	17 x 17 complex kernel (16-bit);
	4 orientations;128 x 128 image (8-bit);
	computation of Gabor energies
		4.3 ms @ 100 MHz

720x576 CMOS Video Camera



SRAMs

Controller ,
digital signal path
and BTU.656 formatter

Product : Intelligent Vision System

0.10 µm :	VIP128	< 30 mm^2
	CMOS-Camera	< 23 mm^2
	DRAM 1 MB	< 1.2 mm^2
	SRAM 256KB	< 4mm^2

		< 60 mm^2

Flash 256KB
< 0.5 mm^2

Total power consumption: 270 mW
200 MHz 100 BOPS 3 µW/MOPS