Semi-Reconfigurable Processors for Fast Image Analysis

Ben Kelly

University of Guelph

March 31, 2011

Summary

- Realtime image analysis
- The IMAP architecture
- IMAP-CE and IMAPCAR
- IMAPCAR2

Realtime image analysis - what is it?

- Transformation and analysis of images in real time
  - Typically, this means 30fps or 60fps
  - At 30fps you have about 33ms to process each frame
- Subset of *realtime image processing*
- Significantly more difficult than transformation only

Realtime image analysis - what is it good for?

- Obstacle detection and avoidance
- Lane following
- Threat detection
- Object identification
- In short: machine vision

Realtime image analysis - the problem

- 30fps image analysis is not cheap
- Embedded processors don't have the power
- GPPs are too expensive and too power-hungry
- ASICs are too inflexible

GPP	General Purpose Processor
ASIC	Application Specific Integrated Circuit

IMAP - the Integrated Memory Array Processor

- Described by Fukushima et al. in 1995
- Designed to quickly and cheaply perform image processing tasks
- 8-bit SIMD RISC architecture
- Intended to act as a coprocessor to a separate CPU

RISC	Reduced Instruction Set Computer
SIMD	Single Instruction Multiple Data

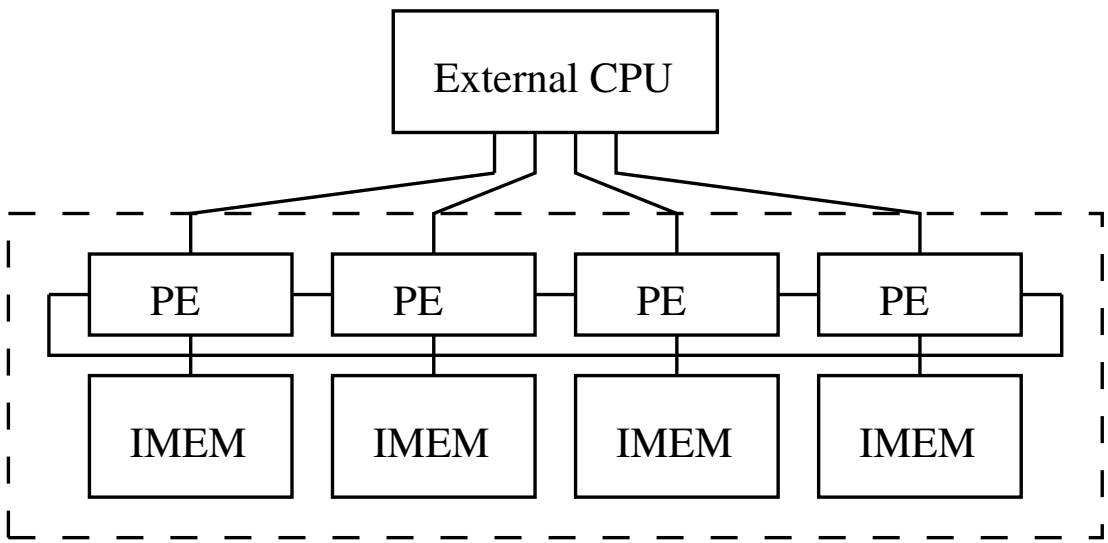
IMAP - Internal Components

- 64 8-bit SIMD PEs
- 2KB of IMEM per PE
- Simple ring network connecting PEs
- Tree network connecting external CPU to PEs

PE	Processing Element - a SIMD miniprocessor
IMEM	Internal Memory

IMAP - Internal Design

- Each PE can only directly access its own registers and IMEM
- Ring network lets PEs transfer data to registers of adjacent PEs



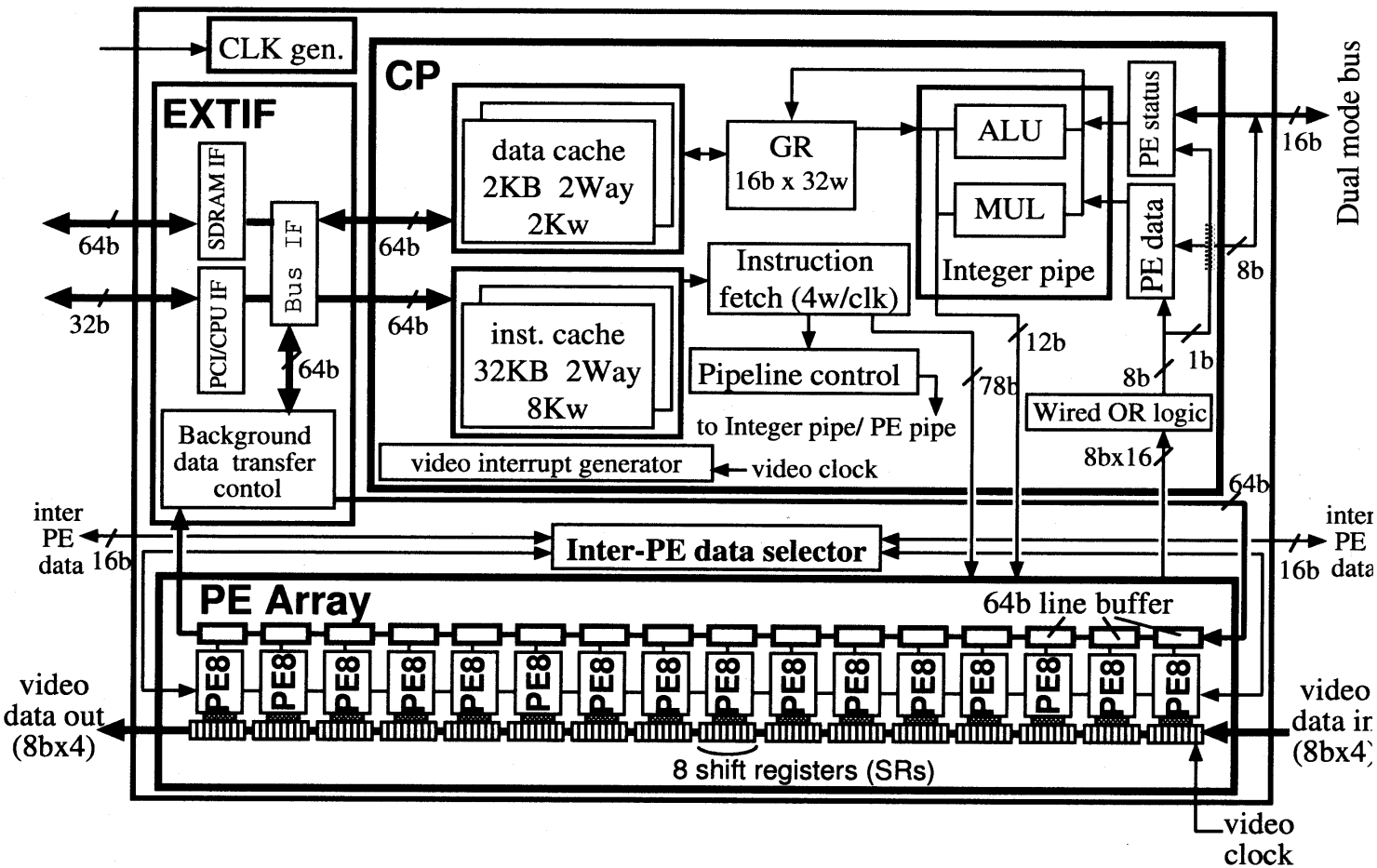
IMAP - Programming Model

- *One-Dimensional C* (1DC)
- C programming language with data-parallel extensions
  - Description of data structures spread across IMEM
  - SIMD processing of these structures
  - Collection of results
- Main code runs on the CPU; 1DC compiler automatically dispatches parallel operations to the PEs

IMAP-CE

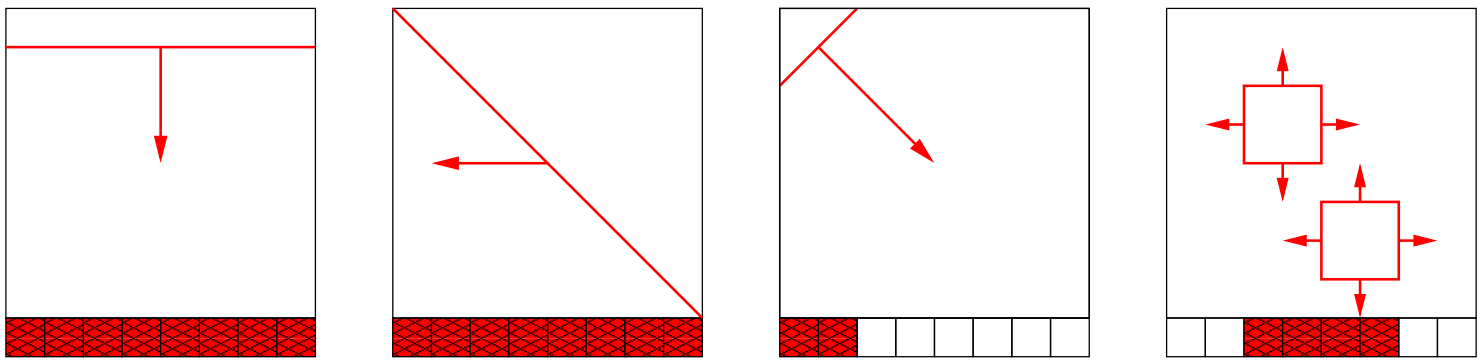
- Prototype IMAP implementation, developed by Kyo et al.
- CPU is now integrated onto the chip as the *Central Processor* (CP)
- 128 PEs, 2KB of IMEM each
  - Designed to hold an entire 512x512 image in PE IMEM

IMAP-CE - Internal Design



IMAP-CE - Usage

- Builtin support for four types of image access
  (a) Row-wise
  (b) Row-systolic
  (c) Slant-systolic
  (d) Autonomous



IMAPCAR

- Refinement of IMAP-CE, designed for use in automobiles
- ROI and DMA upgrades, including ROI scaling
- Video bus width tripled; IMAPCAR can handle two 768p or three 512p video streams simultaneously
- Program and data memory protected by ECC and parity checks respectively

ROI	Region of Interest
DMA	Direct Memory Access
ECC	Error Correction Code

IMAPCAR In Practice

- Benchmarks:
  - 3x faster than IMAP-CE
  - Comparable power requirements

IMAPCAR - Weaknesses

- Initial stages of image analysis are all SIMD
- Once regions of interest are identified, they must be analyzed
- Analysis is intrinsically MIMD
- Problem: IMAPCAR has no MIMD support!
- ROI analysis ends up happening in serial on the CP, with the PEs idle

IMAPCAR - Possible Solutions

- Use multiple IMAPCAR chips
  - Cost and power draw increase proportionally
  - Extra IMAPCARs are idle when performing SIMD operations
  - All PEs are idle when performing MIMD operations
- Add more CPs to the IMAPCAR
  - Greatly increases complexity
  - Still "wastes" the PEs

IMAPCAR2

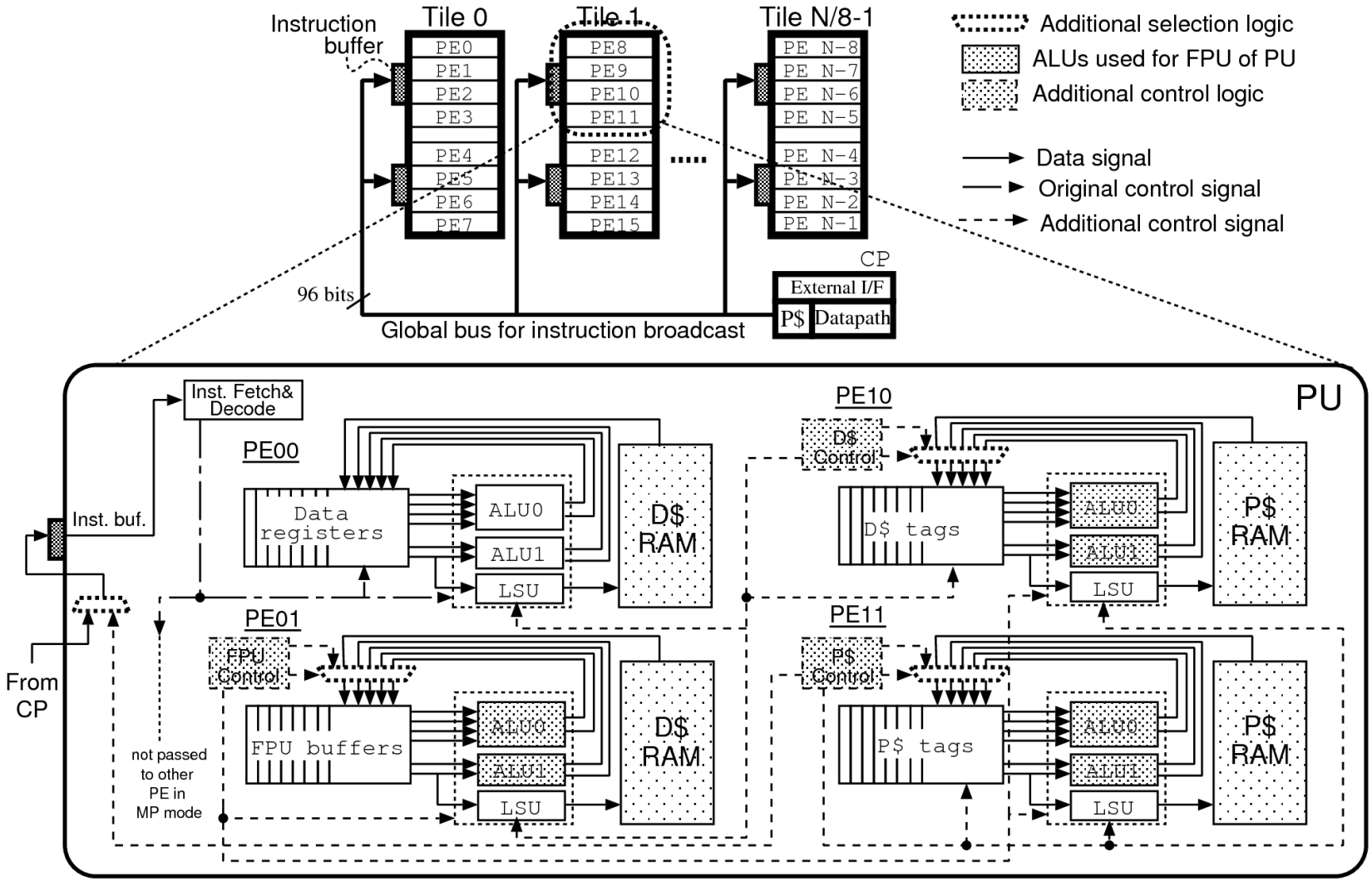
- Successor to IMAPCAR, intended to address MIMD issue
- Minor upgrades:
  - PEs and CP now use the same datapath and instruction set
  - IMEM amount doubled
  - Tiling capability
  - 16-bit addressing and instruction width

IMAPCAR2 - MIMD Support

- PEs are grouped into sets of 4
- Each group is augmented with hardware that lets them combine to function as an additional CP
  - IMEM becomes data and instruction caches
  - Extra ALUs become FPU components
  - PE 0 handles registers and instruction dispatch
- Total hardware overhead is around 20%

ALU	Arithmetic/Logic Unit
FPU	Floating Point Unit

IMAPCAR2 - Internal Design



IMAPCAR2 - Programming Model

- Like earlier IMAPs, uses 1DC
- Additional extensions for controlling PUs
- C API for PU usage is pthreads-compatible, ie, shared-memory
  - Synchronization via shared structures - mutexes, semaphores, barriers
  - Communication by reading and writing known areas of memory

IMAPCAR2 - Weaknesses

- Pthreads-alike programming model implies shared memory
- However, the IMAPCAR2 has no cache coherency
  - Cache$\leftrightarrow$RAM transfers must be explicitly invoked
- Trying to use IMAPCAR2 as a shared-memory system will not work
- Unlike IMAPCAR's deficiencies, this can be fixed entirely in software by providing a message-passing API

Conclusions

- IMAP is an old but still highly effective architecture for image processing
- IMAPCAR2 shows promise as a modern refinement of that design
- However, additional software support is needed to fully realize its potential

References I

[1] Y. Fujita et al. “A 64 parallel integrated memory array processor and a 30 GIPS real-time vision system”. In: *Proceedings of the Computer Architectures for Machine Perception*. CAMP ’95. Washington, DC, USA: IEEE Computer Society, 1995, pp. 242–. ISBN: 0-8186-7134-3. URL: http://portal.acm.org/citation.cfm?id=526253.791749.

References II

[2] Yoshihiro Fujita et al. “A 10 GIPS SIMD Processor for PC-based Real-Time Vision Applications — Architecture, Algorithm Implementation and Language Support”. In: *Proceedings of the 1997 Computer Architectures for Machine Perception (CAMP '97)*. CAMP '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 22–. ISBN: 0-8186-7987-5. URL: http://portal.acm.org/citation.cfm?id=522770.791783.

References III

[3] S. Kyo et al. "A 51.2 GOPS scalable video recognition processor for intelligent cruise control based on a linear array of 128 4-way VLIW processing elements". In: *IEEE Journal of Solid-State Circuits* 38.11 (2003), pp. 1992–2000. DOI: 10.1109/JSSC.2003.818128.

References IV

[4] Shorin Kyo and Shin'Ichiro Okazaki. "IMAPCAR: A 100 GOPS In-Vehicle Vision Processor Based on 128 Ring Connected Four-Way VLIW Processing Elements". In: *J. Signal Process. Syst.* 62 (1 2011), pp. 5–16. ISSN: 1939-8018. DOI: http://dx.doi.org/10.1007/s11265-008-0297-0. URL: http://dx.doi.org/10.1007/s11265-008-0297-0.

References V

[5] Shorin Kyo and Shin'ichiro Okazaki. "In-vehicle vision processors for driver assistance systems". In: *ASP-DAC '08: Proceedings of the 2008 Asia and South Pacific Design Automation Conference*. Seoul, Korea: IEEE Computer Society Press, 2008, pp. 383–388. ISBN: 978-1-4244-1922-7.

References VI

[6] Shorin Kyo, Shin'ichiro Okazaki, and Tamio Arai. "An Integrated Memory Array Processor Architecture for Embedded Image Recognition Systems". In: *ISCA '05: Proceedings of the 32nd annual international symposium on Computer Architecture*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 134–145. ISBN: 0-7695-2270-X. DOI: http://dx.doi.org/10.1109/ISCA.2005.11.

References VII

[7] Shorin Kyo, Shin'ichiro Okazaki, and Tamio Arai. "An Integrated Memory Array Processor for Embedded Image Recognition Systems". In: *IEEE Trans. Comput.* 56.5 (2007), pp. 622–634. ISSN: 0018-9340. DOI: http://dx.doi.org/10.1109/TC.2007.1010.

References VIII

[8] Shorin Kyo et al. “A low-cost mixed-mode parallel processor architecture for embedded systems”. In: *ICS ’07: Proceedings of the 21st annual international conference on Supercomputing*. Seattle, Washington: ACM, 2007, pp. 253–262. ISBN: 978-1-59593-768-1. DOI: http://doi.acm.org/10.1145/1274971.1275006.

[9] *IMAPCAR2: A Dynamic SIMD/MIMD Mode Switching Processor for Embedded Systems*. HOTCHIPS 21 Symposium on High Performance Chips, 2009.

References IX

[10] Kazuyuki Sakurai, Shorin Kyo, and Shin'ichiro Okazaki. "Overtaking Vehicle Detection Method and Its Implementation Using IMAPCAR Highly Parallel Image Processor". In: *IEICE - Trans. Inf. Syst.* E91-D.7 (2008), pp. 1899–1905. ISSN: 0916-8532. DOI: http://dx.doi.org/10.1093/ietisy/e91-d.7.1899.