Hardware/Software-Codesign of a Tightly-Coupled Coprocessor for Video Content Analysis

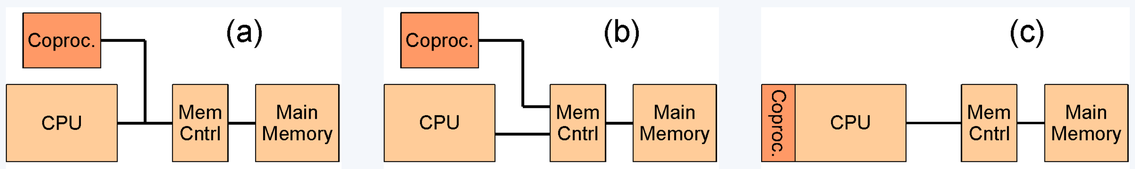
Juergen Wassner, Klaus Zahn and Ulrich Dersch

Lucerne University of Applied Sciences and Arts, School of Technology and Architecture, Switzerland

Lucerne University of Applied Sciences and Arts
HOCHSCHULE LUZERN
Technik & Architektur

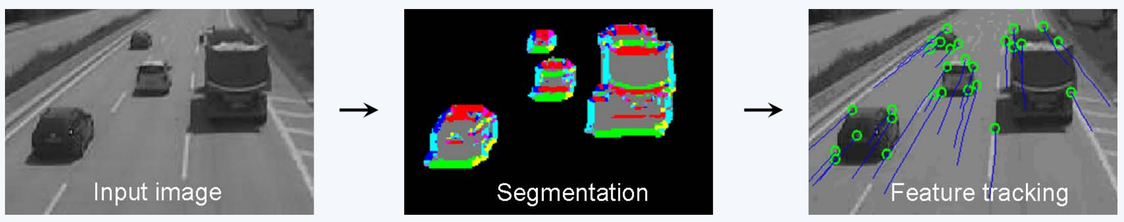
Introduction

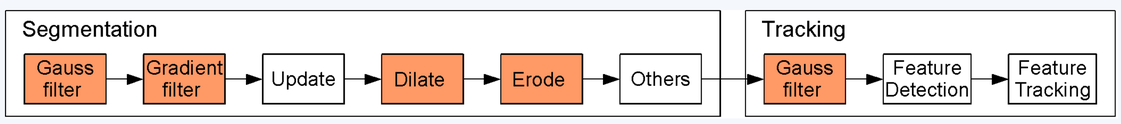
- FPGA hardware accelerators are typically used to boost performance of pixel-based video processing in embedded systems
- Three basic types of such accelerators can be discerned:
  - Coprocessors loosely-coupled to CPU via system bus (a)
  - Coprocessors loosely-coupled to CPU via memory controller (b)
  - Coprocessors tightly-coupled to CPU via specialized interface (c)



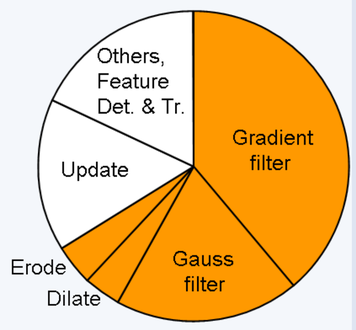
- This work proposes a method for efficient data transfer on such specialized interface and quantifies potential of tightly-coupled coprocessors (TCC)

Target Application & Platform



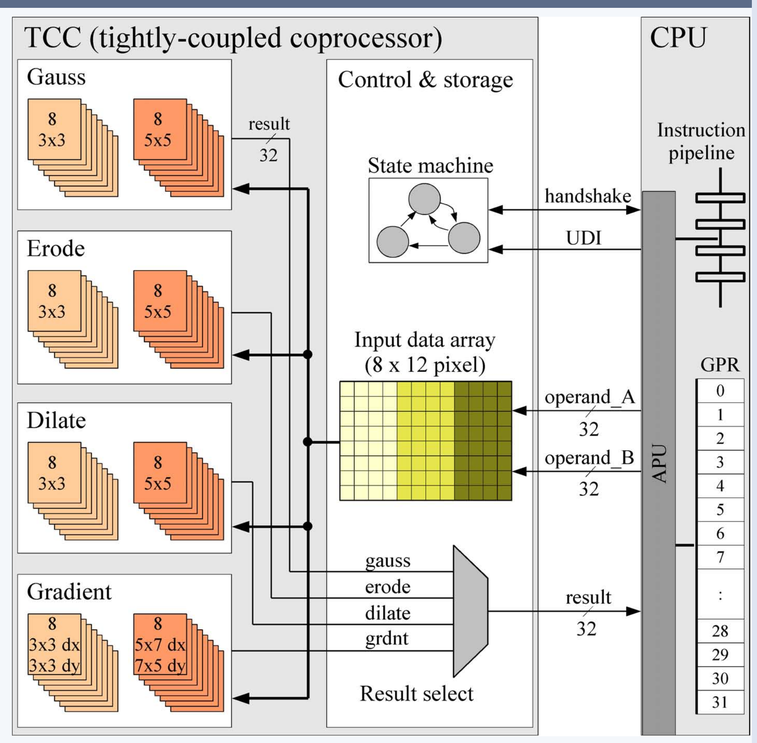


- Video Content Analysis (VCA) algorithm:
  - foreground/background segmentation by pixel-wise gradient calculation and orientation analysis
  - feature detection and tracking on foreground pixels
- Target Platform:
  - PowerPC440x5 integrated in Xilinx Virtex-5 FPGA
  - operations to accelerate consume 2/3 of CPU time



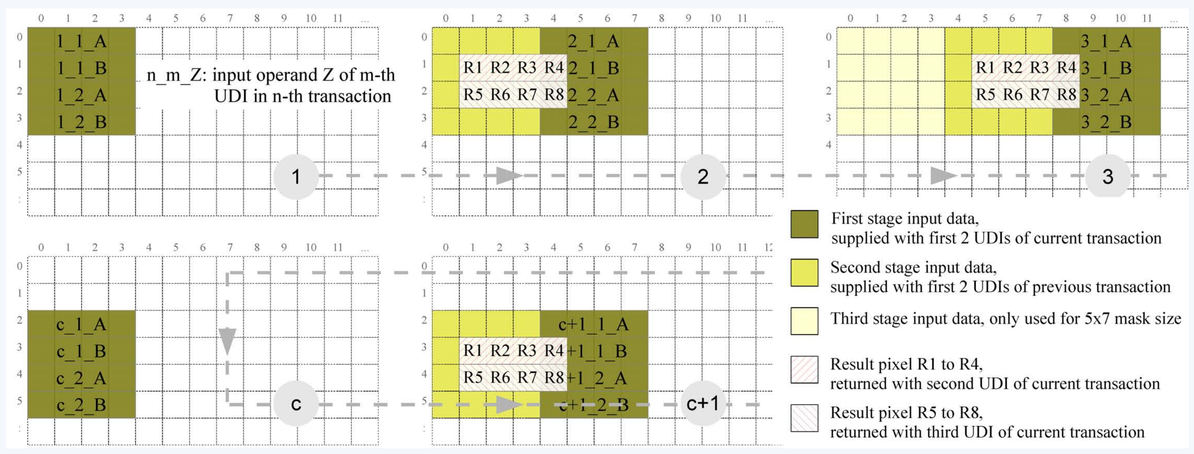
Coprocessor Architecture

- Hard macro: Auxiliary Processing Unit (APU)
  - monitors instruction pipeline within CPU
  - dispatches user-defined instructions (UDI)
  - uses 32-bit data interface (two operands & one result per UDI)
- Custom FPGA Logic
  - input data memory built from flip-flops for flexible parallel access
  - 8 identical filters per operation and mask size



Data Transfer Principal

- Transaction: Sequence of UDIs, provides operands and returns 8 result pixels
- Example for 3x3 mask size shown, uses only 1st- and 2nd-stage input data



Data Transfer & Pointer Offset

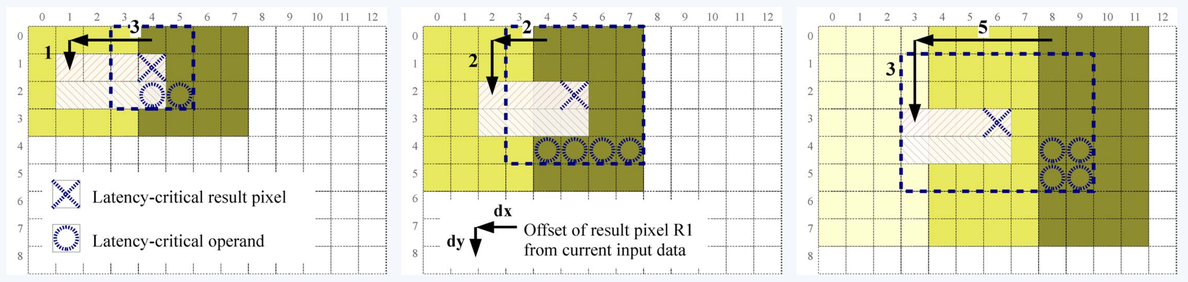
- Generalization to arbitrary mask sizes
  - let r and c be the number of rows and columns ($r, c = 1, 3, 5, \ldots$)
  - assume two operands and one result, of b bytes each ($b = 4$ in our case)
  - offset of result pixel R1 in y- and x-direction:

$$dy = \frac{r-1}{2}, \qquad dx = \left\lceil \frac{c-1}{b} \right\rceil \cdot b - \frac{c-1}{2}$$

  - dx/dy realized by offsets between source and destination pointers in SW
  - total size of input data array (rows $\times$ columns):

$$\text{memory size} = (r+1) \times b \cdot \left(\left\lceil \frac{c-1}{b} \right\rceil + 1 \right)$$

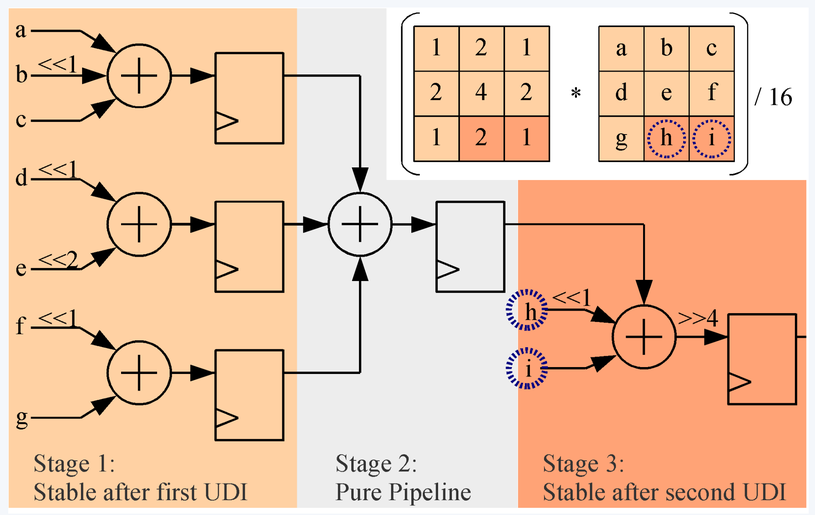
- Latency-critical pixel depends on largest number of 'late' input operands
- Those 'late' operands are critical for the overall transaction latency



3x3 5x5 5x7 / 7x5

Data Path Architecture

- Latency-critical pixel determines architecture of all 8 filter instances
- Optimally all latency-critical operands scheduled for last pipeline stage
- Example: 3x3 Gauss filter
  - two latency-critical operands can be processed in last stage without timing violations
  - multiplications are implemented by shift/add



Experimental Evaluation

- Measurement setup
  - 400/200 MHz CPU/TCC
  - 200 synthetic frames processed with 3 parameters
- Measurement results
  - relative frame rate gain between 18 and 105 %
  - gain increases with resolution and mask size
  - absolute frame rate now independent on mask size
- TCC complexity
  - 13'409 LUTs, 11'434 FFs

mov. obj.	mask size	reso-lution	frame rate [fps] CPU only	frame rate [fps] CPU+TCC	incr. [%]
0	3	QCIF	51.8	68.6	32
		CIF	12.7	16.9	33
		PAL	3.2	4.2	32
	5	QCIF	33.3	68.0	104
		CIF	8.1	16.6	105
		PAL	2.0	4.1	105
5	3	QCIF	35.2	41.7	18
		CIF	10.7	13.4	25
		PAL	3.0	3.8	29
	5	QCIF	24.9	40.5	63
		CIF	7.1	12.9	82
		PAL	1.9	3.7	95

Conclusions

- Substantial overall acceleration of VCA applications can be achieved by a TCC with modest parallelism (i.e. hardware complexity)
- A systematic approach to efficient data transfer between CPU and TCC in two-dimensional signal processing applications has been developed
- Outlook: Investigate further speedup through increased local storage

<juergen.wassner><klaus.zahn><ulrich.dersch>@hslu.ch