TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Tim Häring

Institute of Computer Engineering, Chair of Processor Design

Masterproject

Partial Reconfiguration of FPGA Image Processing Pipelines with PYNQ // Dresden, August 20, 2021

Overview

PYNQ

- High abstraction layer
- Familiar to software developers
- Fast development cycles
- Modular design

Partial Reconfiguration

- Less resource utilization
- Flexible and dynamically adaptive
- Resilient

Image Processing

- Widely used
- Algorithms with similar structure
- `xfOpenCV` library for HLS synthesis

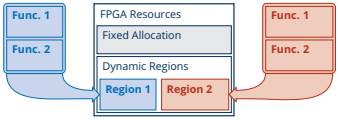


Figure: DPR resource usage.

Framework

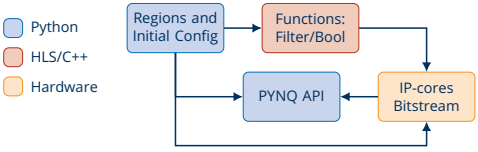


Figure: Framework components overview.

Implemented Components

- Python and `.tcl` scripts
- C++ filter implementations for HLS
- Automatic generation of hardware
- No PYNQ bindings, use `allocate`
- Fixed HW communication paths

Algorithm

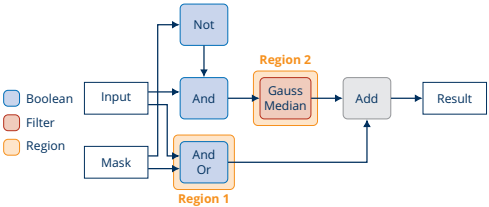


Figure: Implemented processing pipeline.

Overlays

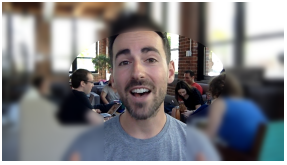
- Base: And/Median & Or/Gauss
- Streaming: no DMA
- Partial: reconfiguration regions
- Software: Cortex-A9@650 MHz

Benchmarking Parameters

- 100 MHz hardware clock
- Full HD images ($\approx$ 2 MB)

Results

(a) Input Image



(b) And/Gauss

(c) And/Median

(d) Mask

(e) Or/Gauss

(f) Or/Median

Figure: Inputs and results of the processing pipeline.

Results — Resource Utilization

Overlay	Slice LUT [%]	BRAM [%]	DSP [%]
Base	54.52	9.29	35.45
Streaming	62.43	16.43	12.18
Partial	31.78	5.36	10.91

Table: Resource utilization percentage of the implemented overlays.

Overlay Results

- Base: two pipelines in parallel
- Streaming: additional HDMI processing overhead
- Partial: shows advantages of partial reconfiguration

Results — Performance

Overlay	Latency [ms]	Performance [frames/s]
Base	22.73	43.98
Streaming	20.83	48.00
Partial	23.35	42.82
Software	365.03	2.73

Table: Performance results of different implementations.

Implementation Results

- Base: two pipelines in parallel
- Streaming: no DMA access
- Partial: reconfiguration takes $\approx 200\,\text{ms} \Rightarrow 10\,\text{frames}$
- Software: little processing power $\Rightarrow$ HW acceleration

Conclusion

Achieved

- Simple hardware acceleration framework ($\times$15)
- Easily usable by software developers through PYNQ
- Extension/HW modification requires HW knowledge
- Fixed communication paths

Outlook

- Integrate DPR into streaming overlay (no DMA required)
- Extend with NOC

Questions?