Reconfigurable Image Signal Processors for Nonlinear CMOS Image Sensors

PhD Final Oral Defence
Integrated Circuits and Systems

Maikon Antonio Ribeiro do Nascimento
Supervisor: Dr. Dileepan Joseph
UofA – ECE – January 9th 2020

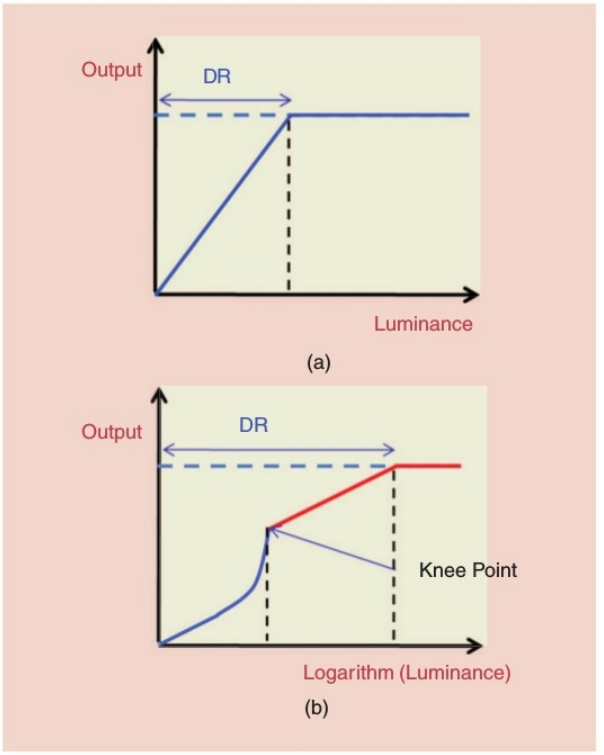
1. Introduction
Objectives and Background

- This thesis explores digital circuit design and digital system integration to realize an image signal processor (ISPr) for a nonlinear CMOS image sensor (CIS).
- All circuits and systems investigated are conceptualized to be generic and, thus, are applicable to a wide variety of nonlinear CISs.
- The platform of choice is an FPGA within a reconfigurable system-on-chip (SoC). Now that Moore's Law is dying, these SoCs are increasingly of interest.

1. Introduction
Objectives and Background

Nonlinear CMOS Image Sensors

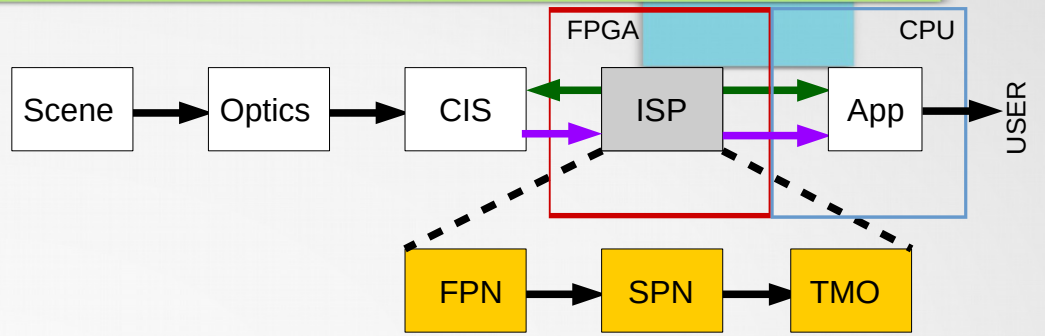
- A nonlinear CIS, e.g., logarithmic (log) or linear-logarithmic (linlog) sensors, can capture a high dynamic range in single exposures at video rates.
- An effective and efficient ISPr is essential to complete a real-time imaging system. **Image signal processor** (ISPr) refers to the integrated circuits and systems used to implement the ISPg.
- **Image signal processing** (ISPg) refers to the signal and image processing algorithms realized in real time.

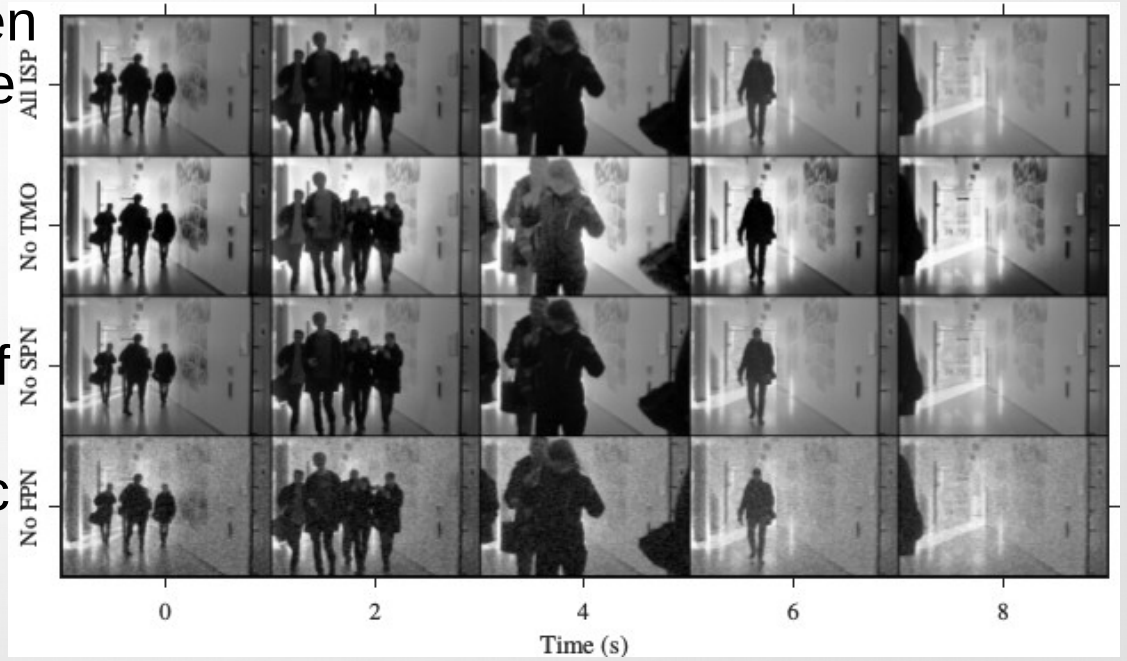


Taken from Kim [1]

1. Introduction
Scope and Methodology

- This thesis investigates digital circuits for FPN correction, SPN filtering, and tone mapping, essential ISPg operations for a nonlinear imaging system.
- It also investigates interfacing issues, when the circuits are realized in a reconfigurable SoC, where part of the system is in an FPGA and another part in a CPU on the same silicon.
- The methodology centers on the design of robust and scalable digital circuits, not to present one specific solution but a generic solution for a wide variety of problems.





2. FPN Correction and SPN Filtering

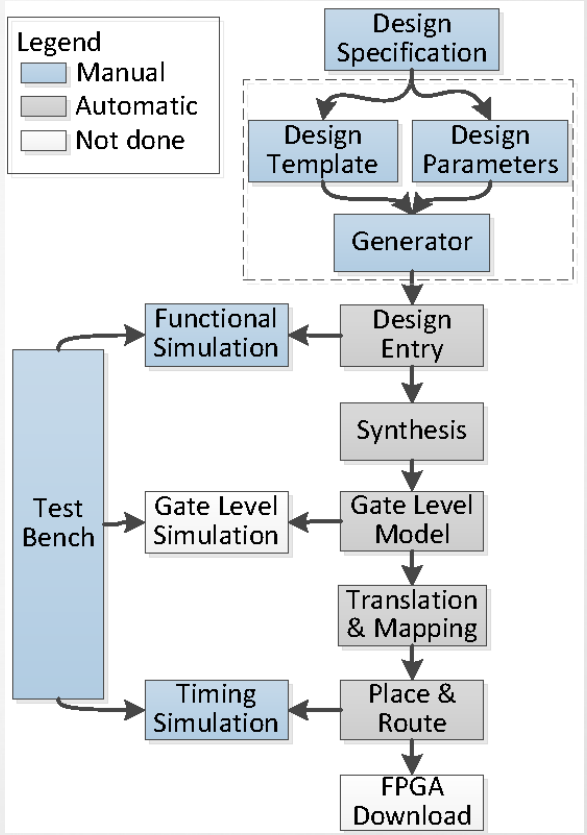
2. FPN Correction and SPN Filtering Introduction

- In a review paper, Kim [1] of Samsung writes in favour of linlog approaches, mainly to avoid motion artifacts of multi-frame linear approaches, but notes that fixed pattern noise (FPN) is more problematic.
- Like Hoefflinger [2], we favour digital circuits for nonlinear FPN correction, but we address salt-and-pepper noise (SPN) filtering and our methods are not restricted to classic logarithmic (log) sensors.

Taken from Hoefflinger [2]

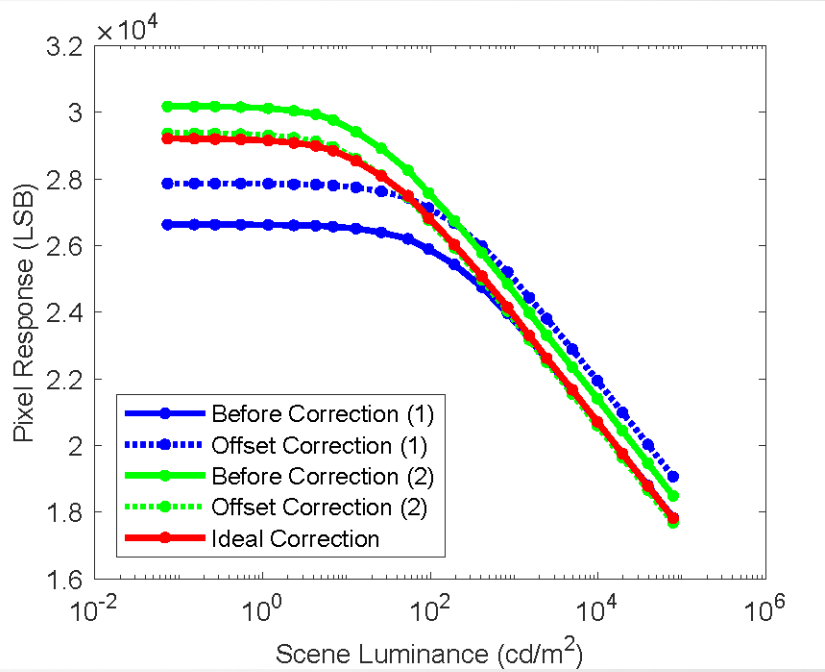
2. FPN Correction and SPN Filtering Background and Methods

- We propose methods that generate novel digital circuits for an ‘arbitrary’ image sensor, such as log or linlog.
- Our digital circuits are coded in VHDL, allowing them to be implemented in an FPGA or, eventually, an ASIC.
- Unlike a standard design flow, we interface a scripting language (MATLAB) to FPGA design tools, adding three aspects (dashed box) to make Design Entry automatic.



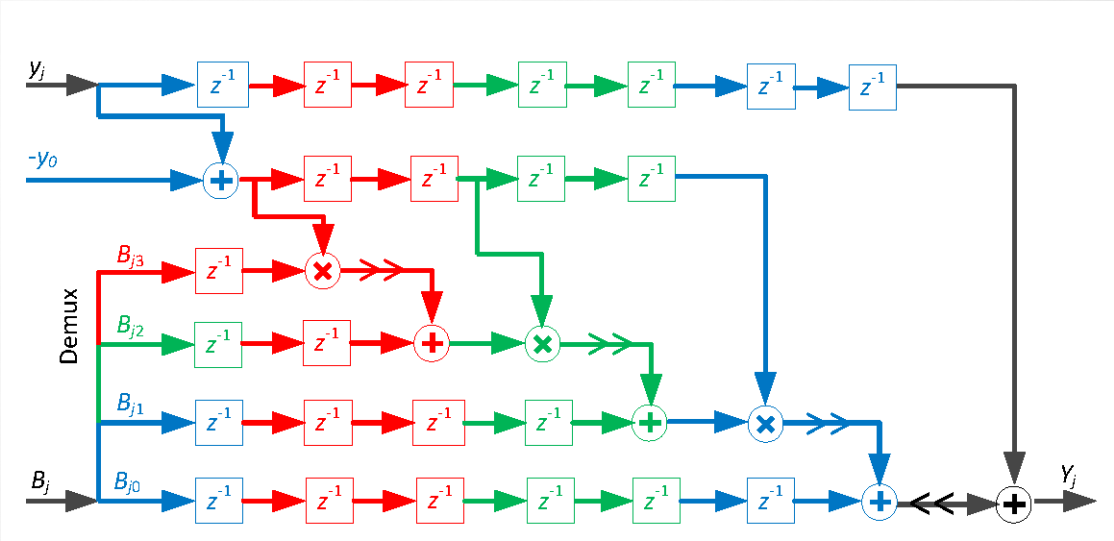
2. FPN Correction and SPN Filtering Background and Methods

- We exploit a recently-published algorithm [3] for polynomial-based FPN correction.
- Two pixels shown, from a log sensor, cannot be equated using offset correction alone.
- Ideal correction is achieved, for all pixels, using cubic polynomials. With linlog, higher degree polynomials may be employed.



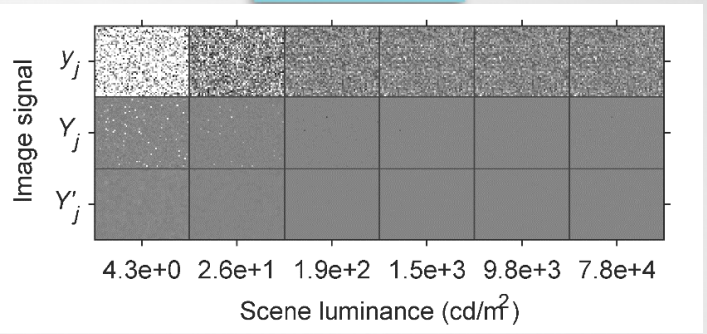
2. FPN Correction and SPN Filtering Background and Methods

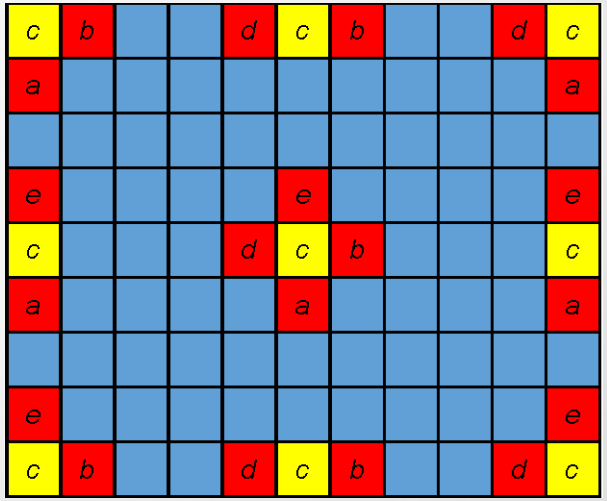
- Polynomial-based FPN correction may be realized, with a recursive pipeline circuit, using delays, bit shifts, and integer arithmetic.
- FPN correction coefficients enter from external memory. Signals are processed in hard real time, i.e., synchronous with a clock.
- Offset correction is shown here in black, offset and gain in black and blue, etc.



2. FPN Correction and SPN Filtering Background and Methods

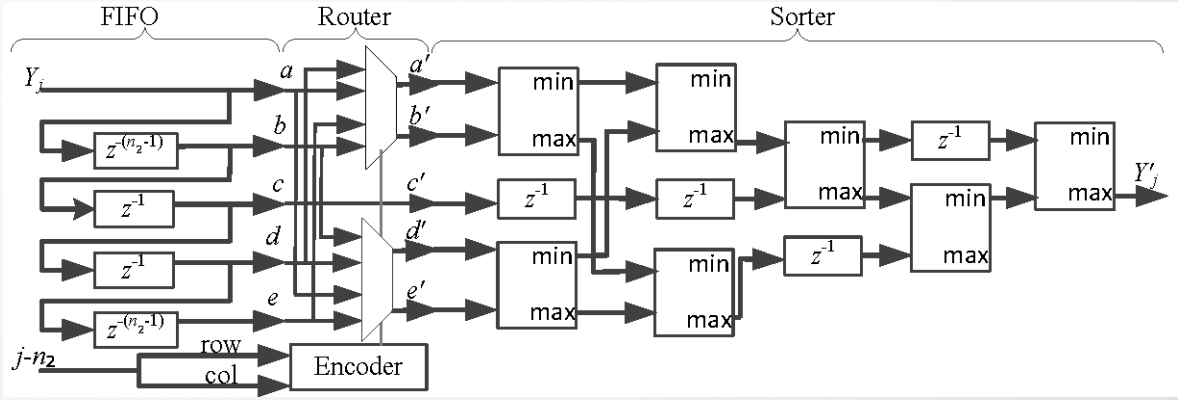
- FPN correction is not enough over a high/wide DR. Light-sensitive outliers, which cause dynamic SPN, are best tackled using median filtering.
- For monochromatic image sensors, the smallest interior window for a median filter has a five-pixel shape.
- To preserve image dimensions and avoid averaging, three-pixel windows are used at the boundaries.





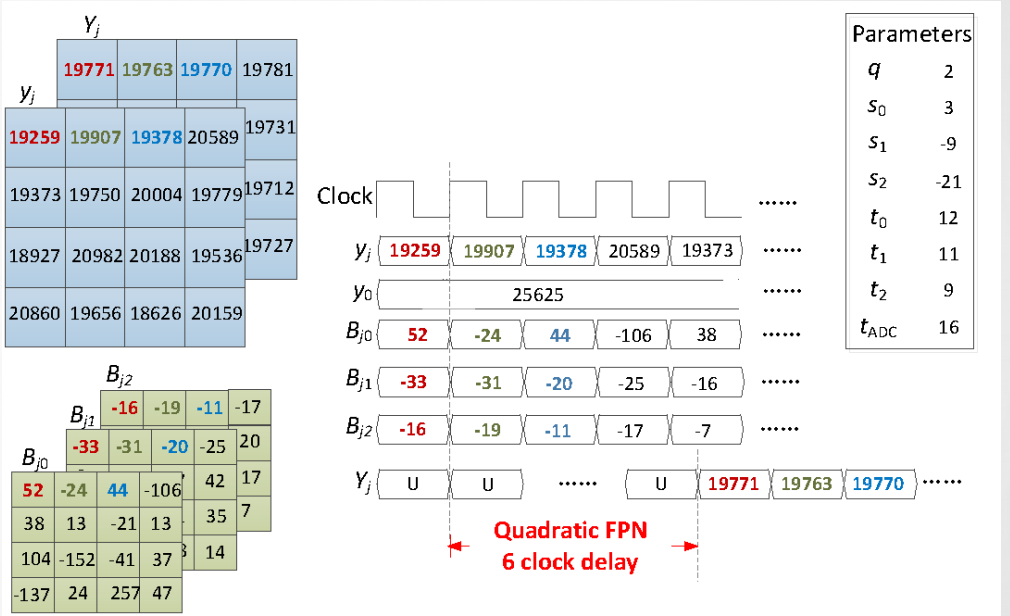
2. FPN Correction and SPN Filtering Background and Methods

- With a three-stage pipeline circuit, median filtering is also realized synchronously. Here, n_2 is the number of columns.
- The FIFO buffers two rows of pixels to produce a five-pixel window. The median value is obtained using a sorter.
- At boundary pixels, the router duplicates two carefully chosen values, which lets the same sorter obtain the correct median.



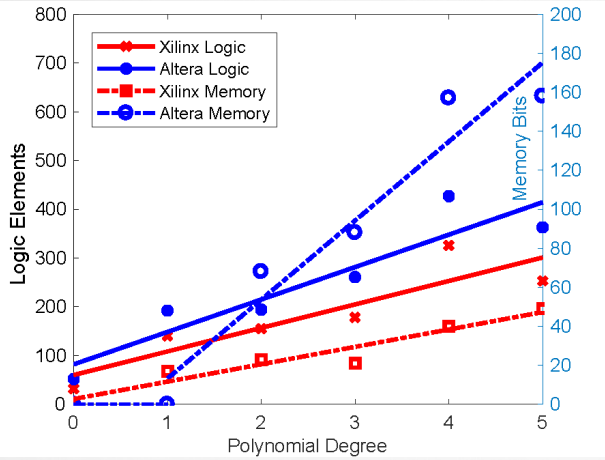
2. FPN Correction and SPN Filtering Results and Discussion

- After manual validation, illustrated, of very small test cases, larger cases were automatically validated using MATLAB. There was zero bit error.
- The FPN circuit was very efficient, e.g., it used 2.06% and 0.01% of the logic and memory in the Xilinx device for cubic polynomials.
- The same evaluation is made for the SPN filtering. Results are in the thesis.

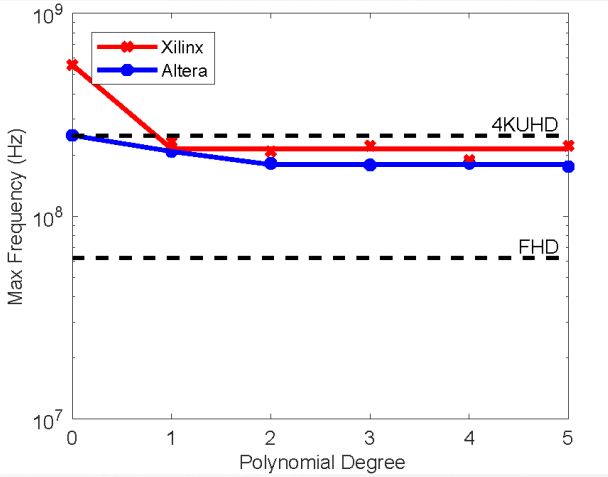


2. FPN Correction and SPN Filtering
Results and Discussion

- FPN correction complexity:



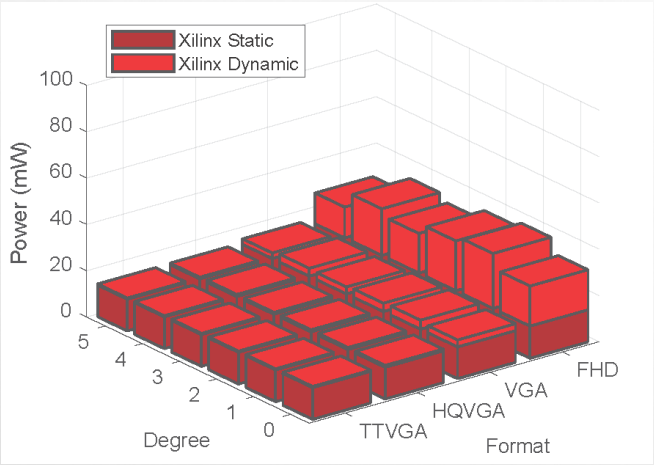
- FPN correction max frequency:

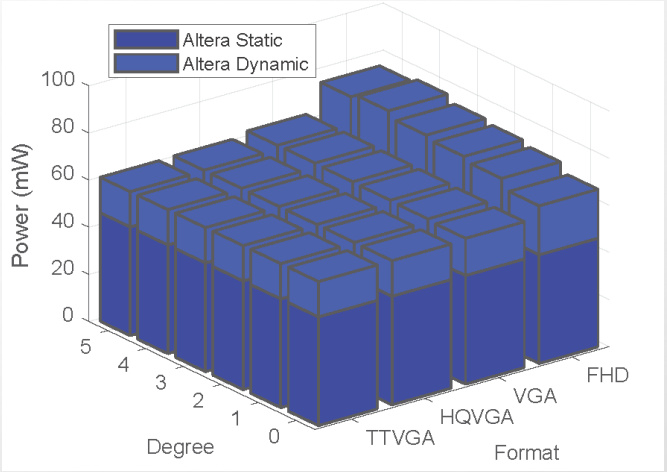


- Digital circuits were placed and routed for Xilinx XC6SLX4 and Altera EP3C5 devices, the simplest FPGAs in the lowest-cost families still in production.
- Using FPGA design tools (ISE 14.7 and Quartus 13.0), we evaluated circuit complexity, max frequency, and power consumption vs. key parameters.
- Video formats tested included small (TTVGA and HQVGA), medium (VGA), and large (FHD and 4KUHD) image sizes. We used static timing analysis to find the max frequency. Power consumption was estimated at 30 fps.

2. FPN Correction and SPN Filtering
Results and Discussion

- Power consumption was evaluated only for supported video formats, as determined by max frequency (FHD video was easily supported).
- Total power is generally low. Static power, which dominates the dynamic power, is consumed by the FPGA irrespective of operation.



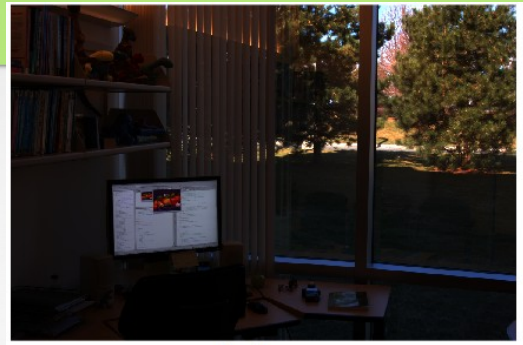


2. FPN Correction and SPN Filtering Conclusion

- Analog [4] or analog-and-digital [5] circuits to correct offset or offset-and-gain variation of linlog sensors suffer from limited SNDR in the log region. These works also have image size, frame rate, and power limitations.
- Hoefflinger's [2] digital circuits to correct offset, gain, and bias variation of log sensors worked well but cannot be applied to other nonlinear sensors, unlike the equally-efficient polynomial-based method adopted here.
- Our digital circuit methods, to correct and filter noise efficiently in hard real time, exploit a generic design flow to support a wide variety of sensors. We validated the methods by generating and simulating a variety of circuits for the simplest FPGA devices available.
- Similarly, our SPN filter outperforms a recent FPGA-based median filter [6]. Results are shown in the thesis.

3. Histogram-Based Tone Mapping

3. Histogram-Based Tone Mapping Introduction



(a)

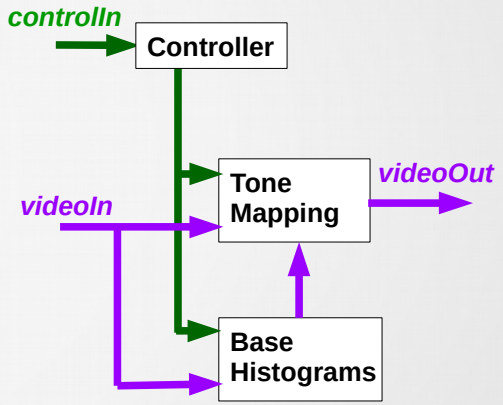
(b)

Taken from MATLAB example

- Tone mapping is an ISPg operator, a TMO, that adapts HDR images to LDR displays.
- Though extensively researched not all TMOs are designed for hard real-time operation nor are they validated using videos.
- The implementation of a TMO should also be simple enough to embed in low-cost platforms for imaging systems.
- Eilertsen *et al.* [7] surveyed TMOs for video processing, criticizing ones with video artifacts, and concluded the best ones are global and based on histograms.
- This work contributes, validates, and evaluates digital circuits to implement a global TMO based on histograms, suitable for ISPr integration in hard real time.

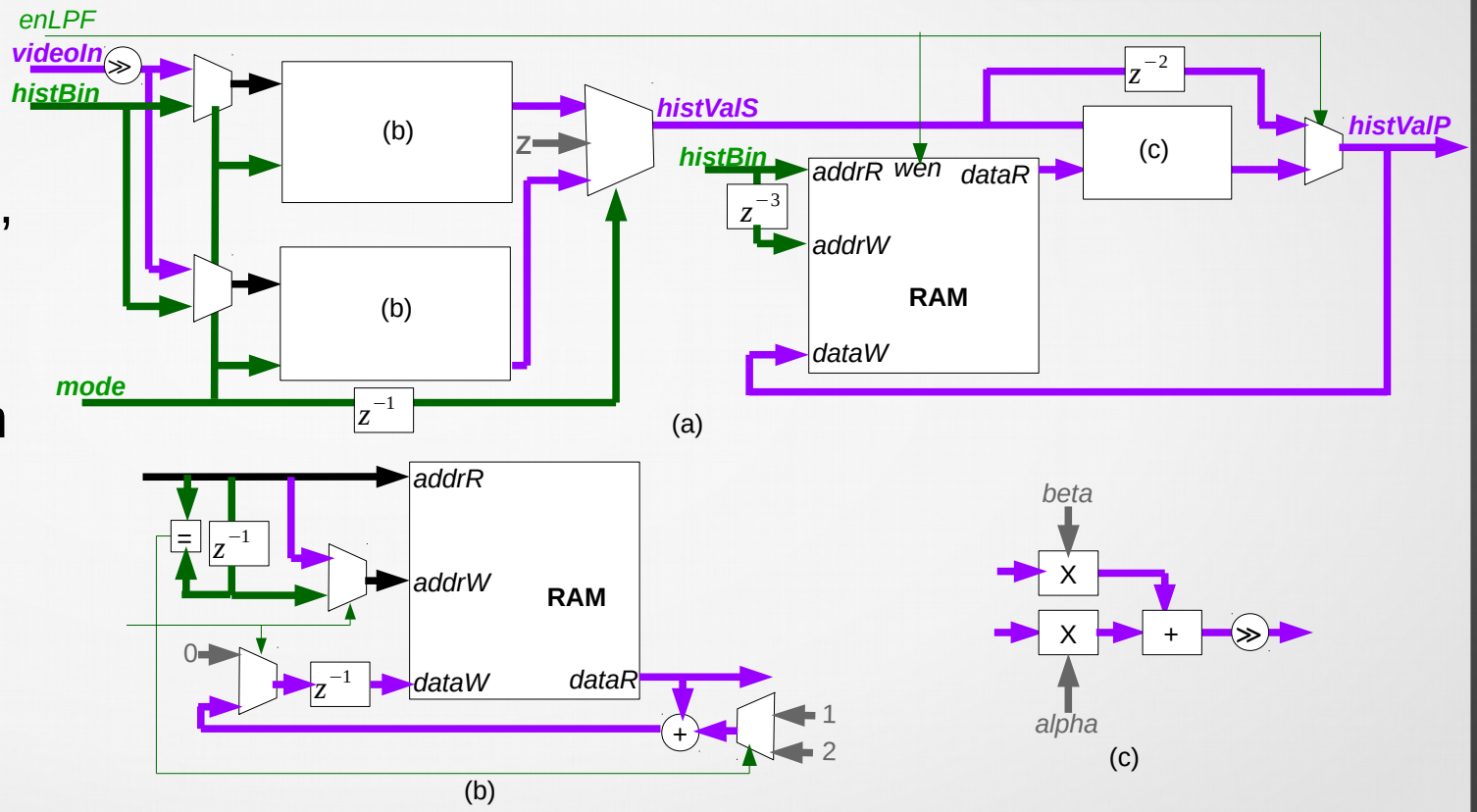
3. Histogram-Based Tone Mapping Background and Methods

- Li *et al.*'s algorithm [8] is similar to Ward *et al.*'s algorithm [9]. They both use histogram equalization, with histogram ceilings to control noise, to create a mapping function. Li *et al.* also apply a low-pass filter (LPF) to the scene histogram to avoid flickering artifacts.
- The top-level circuit design of our histogram-based TMO is shown, where the data paths are shown in purple with control signals in green. Thick lines and bold fonts identify multi-bit buses.
- There are two parallel data paths. One updates the base histograms while the other uses them to map tones. This modular approach favours the adaptation of the circuits, or parts thereof, to implement other algorithms.



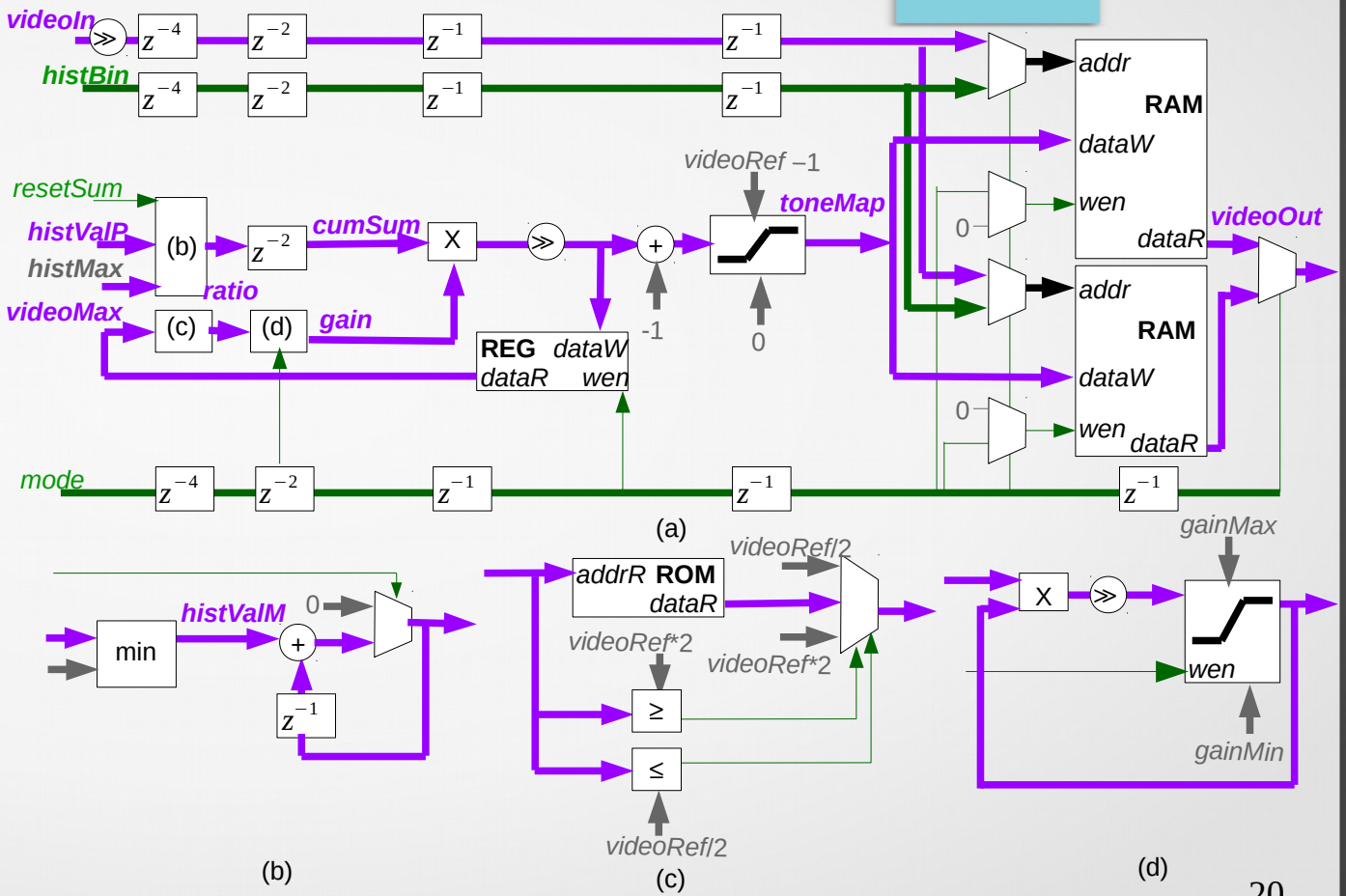
3. Histogram-Based Tone Mapping Background and Methods

- Circuit schematics of the base histograms module: It is comprised of simple components, such as RAMs, muxes, and delays.
- Three RAMs are used, two to store the scene histogram (ping-pong buffer) and one the perceived histogram.
- It is an efficient pipelined design, with an extra RAM to facilitate hard real-time operation.



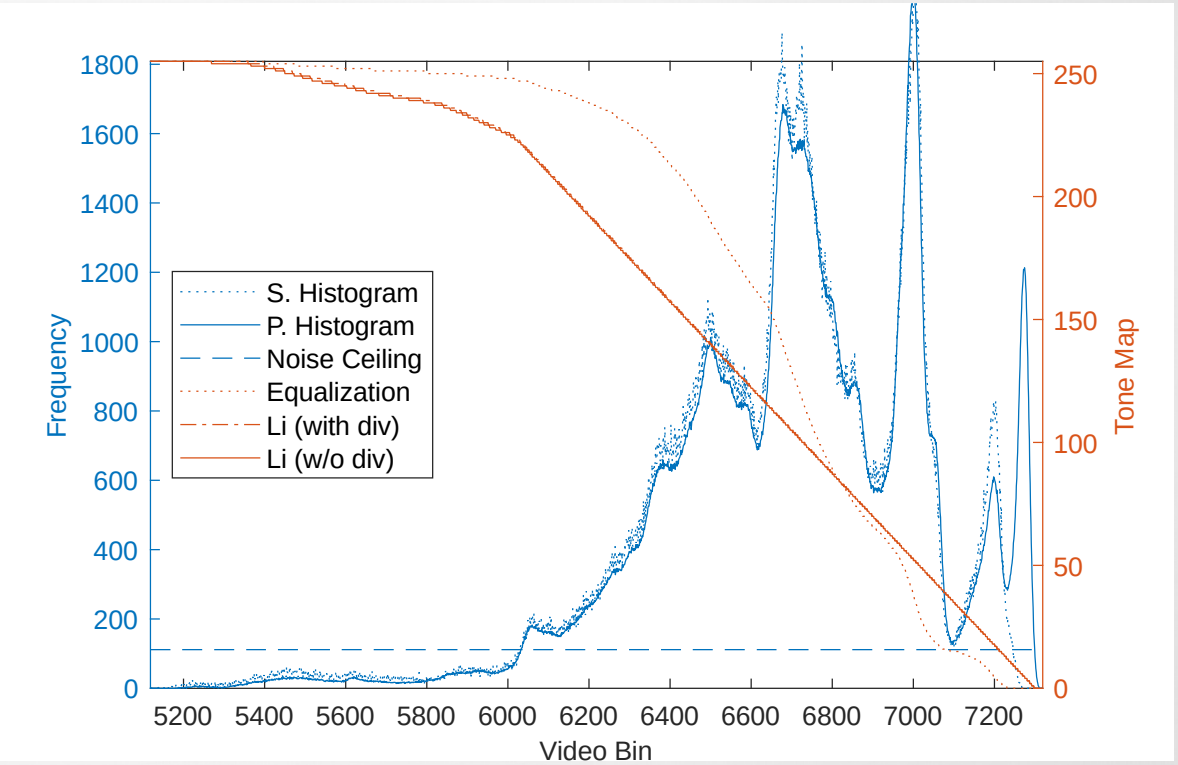
3. Histogram-Based Tone Mapping Background and Methods

- The tone mapping module has another ping-pong buffer, a design where one part applies the mapping, while another updates the mapping function based on previously-computed histograms.
- For high frequency operation, the original algorithm had to be modified by removing a division operation, replacing it with multiplications, look ups, bit shifts, and feedback.



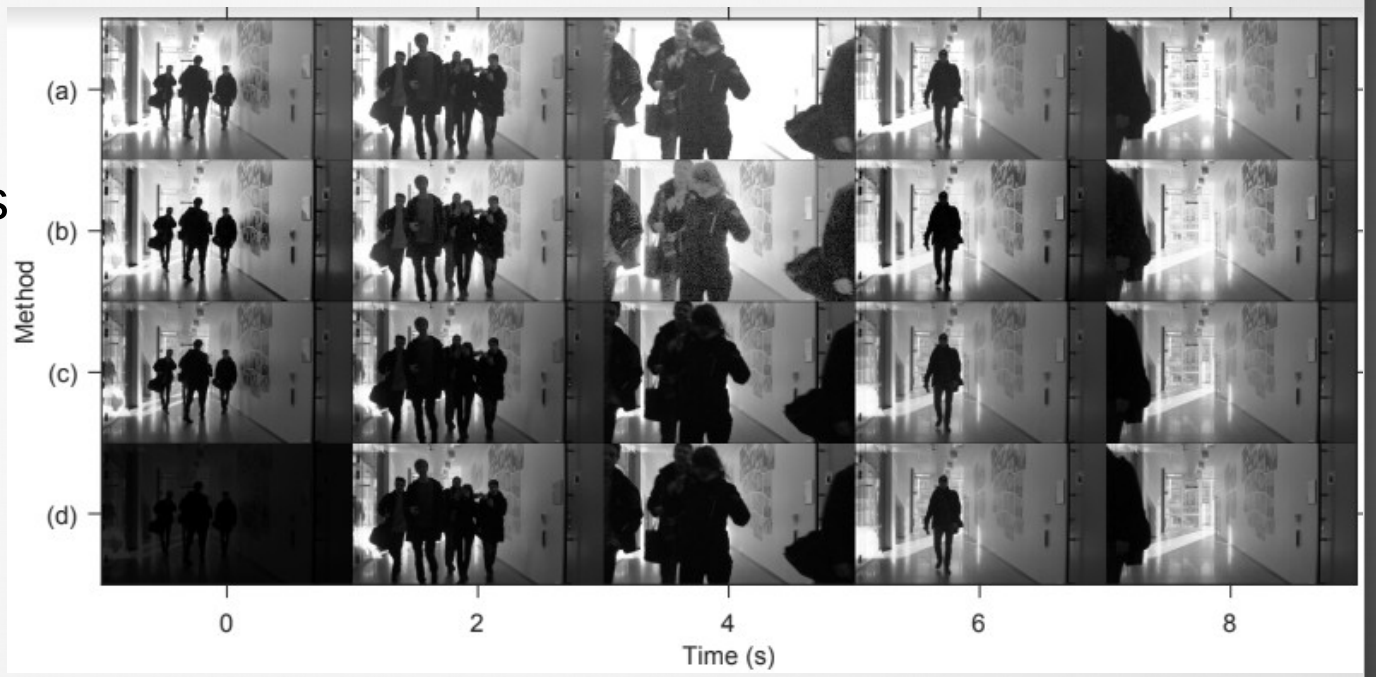
3. Histogram-Based Tone Mapping Results and Discussion

- Internal signals, related to the scene (S) histogram, perceived (P) histogram, and Li *et al.*'s tone map (w/o div), were tapped and validated.
- A bit true validation of these internal signals is a proxy for bit true validation of key internal states. Not only does this enable bit true validation of the external signals but also it means the design is more thoroughly debugged.

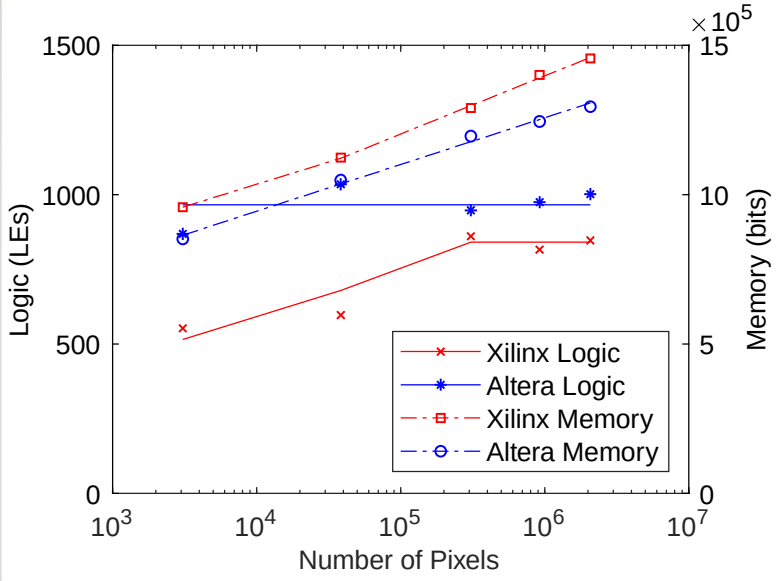


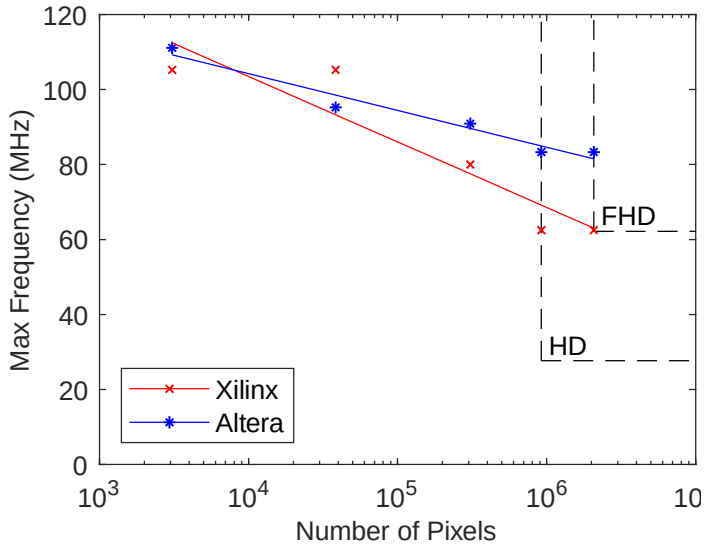
3. Histogram-Based Tone Mapping Results and Discussion

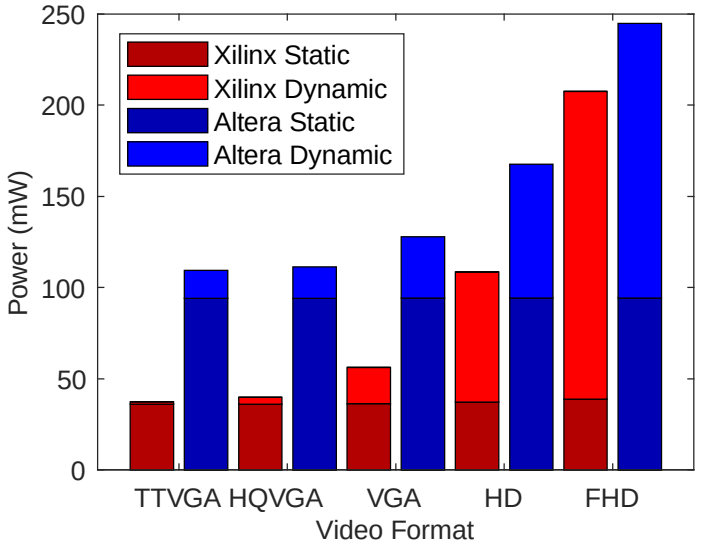
- HDR videos, based on literature data [10], were modelled to have the same response of a monotonic nonlinear CIS.
- Validating with a real video allows us to check for the presence of **visual artifacts** that would be objectionable to human observers.
- After the first 0.1 s, the without (w/o) division approach, realized both in software and hardware, is similar to the with division one, realized only in software.



3. Histogram-Based Tone Mapping Results and Discussion







- Memory grows linearly with the log of the number of pixels, while logic tends to be constant. The memory use, with Altera, is almost exactly accounted for by the five RAMs.
- Max frequency decreases roughly linearly with the log of the number of pixels.
- Dynamic power grows with the video format but remains on the order of the static power.

3. Histogram-Based Tone Mapping Conclusion

- Most of the FPGA designs have not been validated with HDR videos, like Popovic *et al.*'s [11] logarithmic compression with base variable algorithm that can process FHD in a high-end FPGA. Our work fits in a low-cost FPGA and has zero bit error.
- Video resolution is also a limitation for many designs that do not achieve HD, while our design can process FHD. Mann *et al.* [12] use a simple LUT in a low-cost FPGA, achieving HD, but they do not mention video validation.
- Our design is a well pipelined TMO suitable for low-cost embedded platforms. Due to its modular aspect, parts can be applied to other histogram-based algorithms.
- In order to achieve FHD in hard real time, we modified the background algorithm, replacing divisions with multiplications, look ups, bit shifts, and feedback. Synthesized circuits fit in a low-cost FPGA and we achieved bit true validation.

4. Reconfigurable System-on-Chip

4. Reconfigurable System-on-Chip Introduction

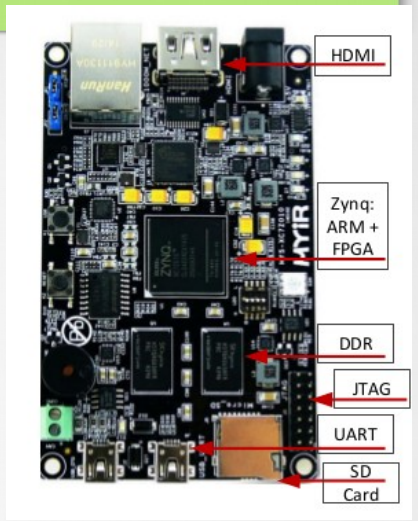
- AI, video processing, and autonomous independent devices require high processing data capabilities, which may require low latency, as well as restrictions in power, weight, and cost. With Moore's law slowing down, new platforms are in need and a heterogenous platform, like a reconfigurable SoC, is one of them.

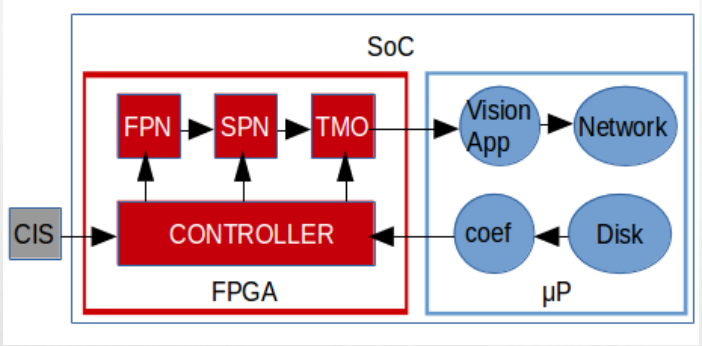
- We present a case study of a multi-stage ISPr implemented in a reconfigurable SoC platform. In our proposed architecture the FPGA is the master of the system. We demonstrate how to **configure** such a system, **strategies of debugging**, and **characterization of the workflow**.

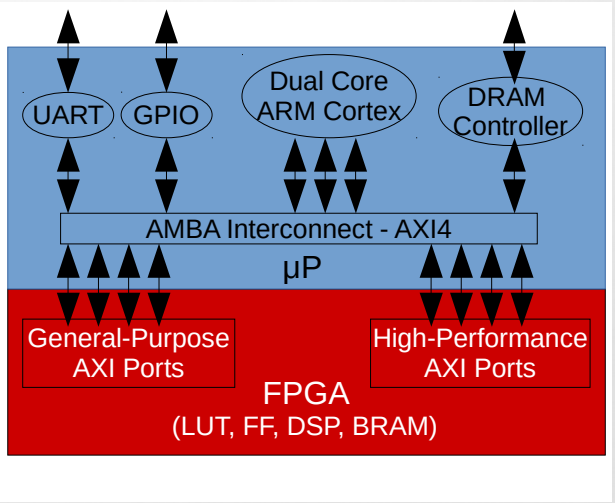
Taken from Arm [13]

4. Reconfigurable System-on-Chip Apparatus and Application

- We used a reconfigurable SoC manufactured by Xilinx, called Zynq-7000. It is composed of a dual-core ARM microprocessor (µP) supporting Linux and an FPGA. The board is manufactured by MYIR and features DDR RAM, an SD card, JTAG, ethernet, and HDMI.
- Some components we used, in red, are in the FPGA (LUTs, registers, DSP blocks, blocks of RAM), while others, in blue, are in the µP (DRAM controller for loading the video file, GPIO for interrupt signaling, UART for debugging). We developed a **controller** for external protocol communication (AXI4-S with DMA for direct memory access), unpacking the data coming from the app, and generating control signals for the SPN and TMO.

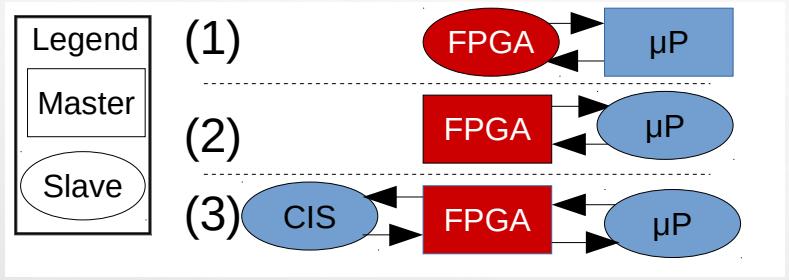


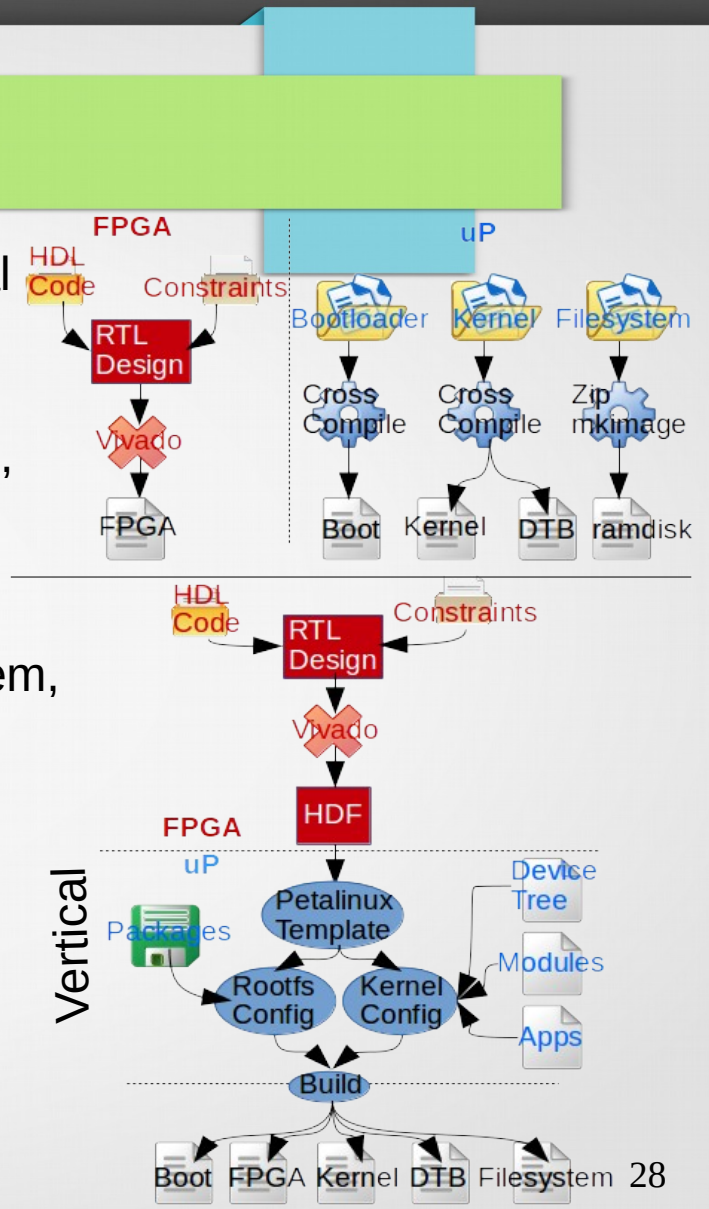




4. Reconfigurable System-on-Chip Interfacing Method

- We present two design flow methods: horizontal, a more independent methodology having FPGA and µP compilation separately; and vertical, emphasizing integration and making for easier configuration.
- This system makes the FPGA the master, which differs from literature approaches, (1), called hardware acceleration. (2) represents our system, where a CIS is simulated, whereas (3) would be the ideal architecture.

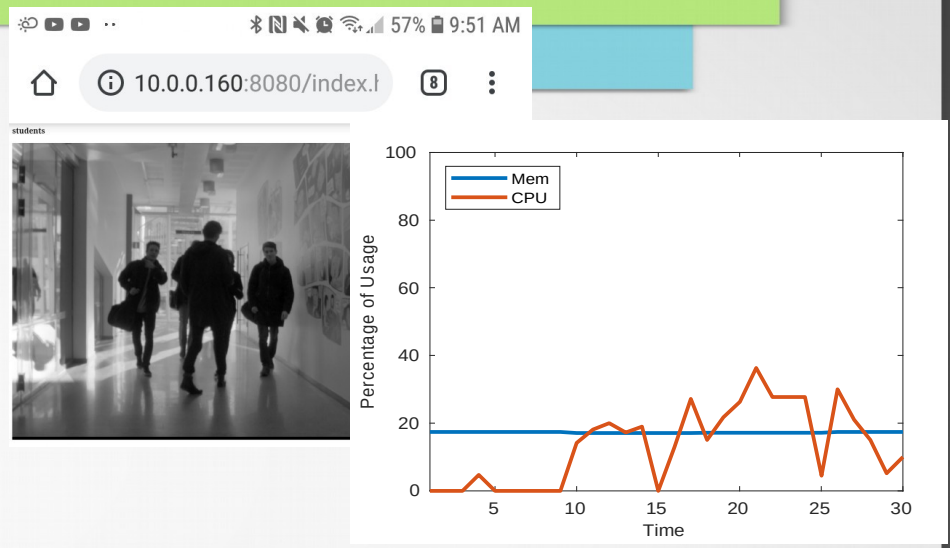




4. Reconfigurable System-on-Chip Results and Discussion

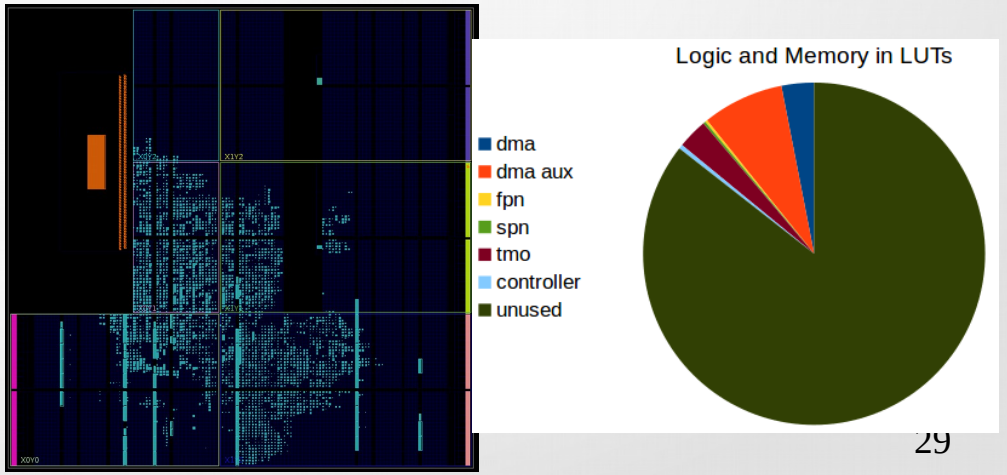
System Validation

- In the FPGA we validate with functional validation. In the µP side the validation starts with Linux devices drivers, DMA transfer, and interrupt request (IRQ) services.
- To validate the system a simple web server is implemented and the figure shows a screenshot of the web page. Using a oscilloscope we verified that IRQs were generated at 30 fps. And the trace shows the µP made busy from time to time though the system keeps processing the video.



System Evaluation

- The floorplan shows the layout of the ISPr for HD resolution. Extra constraints could change the FPGA inference. The complexity of the FPGA is evaluted in terms of LUTs, blocks of RAM, DSPs, and flip flops.
- The pie chart shows a breakdown of the LUTs resource. Additional pie charts are in the thesis.



4. Reconfigurable System-on-Chip Conclusion

- This work has achieved a SoC design flow for hard real-time ISPg of a nonlinear HDR imaging system.
- Our ISPr design used 14.5% of LUTs, 9.0% of flip-flops, and 31.4% of memory, of a Xilinx Zynq SoC, to process HD video at 25 fps. (This work was done prior to completing the faster TMO, without division.)
- One novelty of the design flow is that the FPGA is the master of the SoC platform, which includes a µP running Linux and will, in future, include a nonlinear CIS.
- This approach is especially suited for future heterogeneous computing applications involving HDR and computer vision.

5. Conclusion

5. Conclusion
Summary and Contributions

- **FPN**: Automatic recursive circuit generation for any nonlinear CIS. Scalable up to ultra HD (UHD) video resolution;
- **SPN**: Novel state-of-the-art circuit for filtering, featured on the cover of the *Journal of Imaging Science and Technology*, scalable up to FHD video.
- **TMO**: Efficient pipelined circuits for hard real-time tone-mapping based on histograms, scalable up to FHD, and with adaptable modules.
- **SoC**: Interface methodology for reconfigurable heterogenous computing device; case study of a low cost real-time ISPr suitable for HD video.

  - Published award-winning journal paper: “Digital Circuit Methods to Correct and Filter Noise of Nonlinear CMOS Image Sensors”.
  - Published conference proceeding: “System-on-Chip Design Flow for the Image Signal Processor of a Nonlinear CMOS Imaging System”.

5. Conclusion
Future Work

Nonlinear CMOS Imaging System

- Integrating an actual nonlinear CIS, in the imaging system, would allow us further investigations of how to exceed human vision. Some development work will include a new clock line to manage the coefficients being fetched from the RAM via DMA, and extra circuitry to buffer and synchronize the coefficients with the pixels coming from the physical CIS.
- Previously, coefficients and data were together resulting in a 64-bit word being transfered from the RAM using the DMA. Once they are separate the system will save bandwidth.

Computer Vision Application

- An automotive application could take advantage of the heterogenous computing composed by an FPGA, for fixed-point pixel-level ISPg, and a CPU, for operating system functionalities and high-level computer vision APIs. The HDR in single exposures would handle rich fast changing scenes.
- We developed several ISPr circuits that could be converted to intellectual property (IP) cores to commercialize the IP by, for example, making the IP cores available in a platform like Amazon AWS.

Appendix
Selected References

- [1] Wide Dynamic Range Technologies, Tae-Chan Kim (2014)
- [2] Hoefflinger, High-Dynamic-Range (HDR) Vision, Springer, vol. 26, 2007
- [3] Li *et al.*, “Using Polynomials to Simplify Fixed Pattern Noise and Photometric Correction of Logarithmic CMOS Image Sensors,” Sensors, vol. 15, pp. 26331–52, 2015
- [4] de Moraes Cruz *et al.*, “FPN Attenuation by Reset-Drain Actuation in the Linear Logarithmic Active Pixel Sensor”.
- [5] Storm *et al.*, “Extended Dynamic Range From a Combined Linear- Logarithmic CMOS Image Sensor”.
- [6] Latha and Sasikumar, “A Novel Non-linear Transform Based Image Restoration for Removing Three Kinds of Noises in Images”.
- [7] G. Eilertsen, R. Wanat, R. K. Mantiuk, and J. Unger, “Evaluation of Tone Mapping Operators for HDR-Video.,”Comput. Graph. Forum, vol. 32, no. 7, pp. 275–284, 2013
- [8] J. Li, O. Skorka, K. Ranaweera, and D. Joseph, “Novel Real-Time Tone Mapping Operator for Noisy Logarithmic CMOS Image Sensors,”Journal of Imaging Science and Technology, vol. 60, no. 2, pp. 020404-1–13, Mar. 2016
- [9] Larson *et al.*,”A Novel Non-linear Transform Based Image Restoration for Removing Three Kinds of Noises in Images”.
- [10] Kronander *et al.* ,”Unied hdr reconstruction from raw cfa data,” in IEEE International Conference on Computational Photography (ICCP), Apr. 2013.
- [11] Popovic “Performance Optimization and FPGA Implementation of Real-Time Tone Mapping”
- [12] Mann “REALTIME HDR (HIGH DYNAMIC RANGE) VIDEO FOR EYETAP WEARABLE COMPUTERS, FPGA-BASED SEEING AIDS, AND GLASSEYES (EYETAPS)”
- [13] Nick Heath, “'Moore's Law is dead': Three predictions about the computers of tomorrow”. Sept. 2018.