ECIT | The Institute of Electronics, Communications and Information Technology

FPGA-based Realizations of Embedded Systems

Roger Woods

r.woods@qub.ac.uk

Programmable Systems Laboratory at Queen's (PSL@Q)
Institute of Electronics, Communications and Information Technology (ECIT)
Queen's University Belfast

Programmable Systems Laboratory at Queen's (PSL@Q)

- Focus on programmable platforms for data processing and telecommunications
  - **WiPhyLoc8:** Dynamic WiFi Positioning using Physical Layer Parameters for Location-based Services and Security (*NSF/DEL/SFI - Rice University, University College Dublin*)
  - **Rathlin:** Programmable solutions for intensive image processing applications (*EPSRC - Heriot-Watt, Andor, Xilinx, Thales*)
  - **Nanostreams:** Microserver architecture and a software stack for hybrid transactional-analytical workloads (*EU Framework 7 - IBM, Credit-Suisse, ACE, FORTH-ICS, Neueda, Analytics Engines*)
  - **ENPOWER:** Power optimization of heterogeneous platforms (*EPSRC - University of Bristol*)

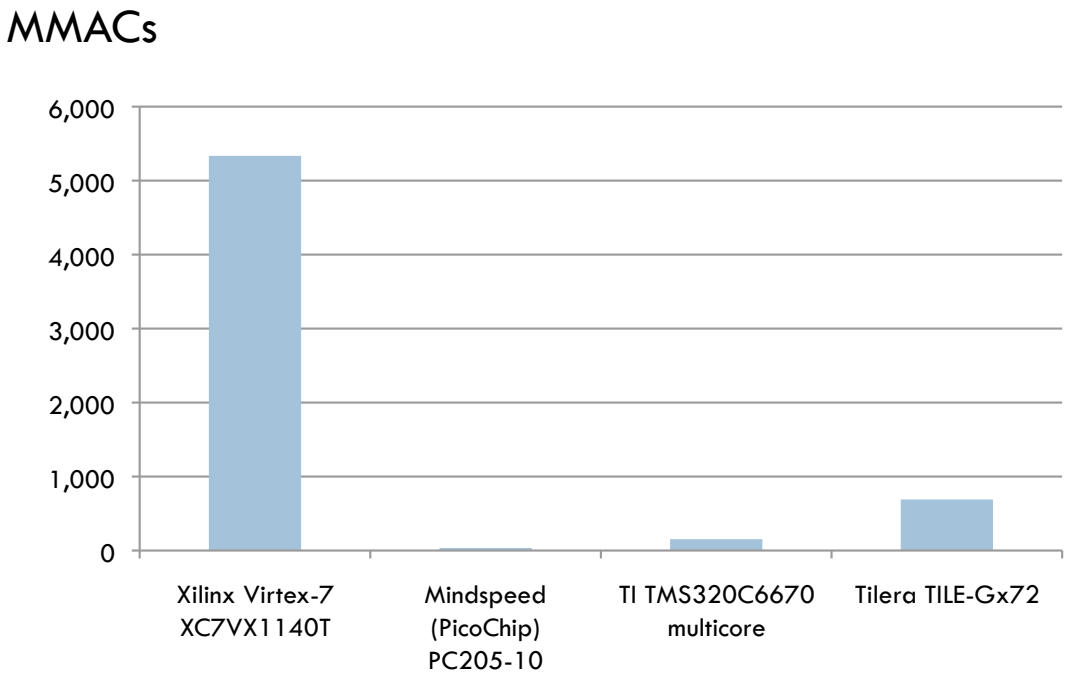
Contents

- Technology review
  - FPGA characteristics
  - SoC features
- Uncertainty perspective
  - Design phase and verification
- IPPro example
  - FPGA processor for image processing
  - Design flow
  - IPPro examples
- FPGA/GPU/CPU comparison
  - Heston modelling (Analytics Engines)

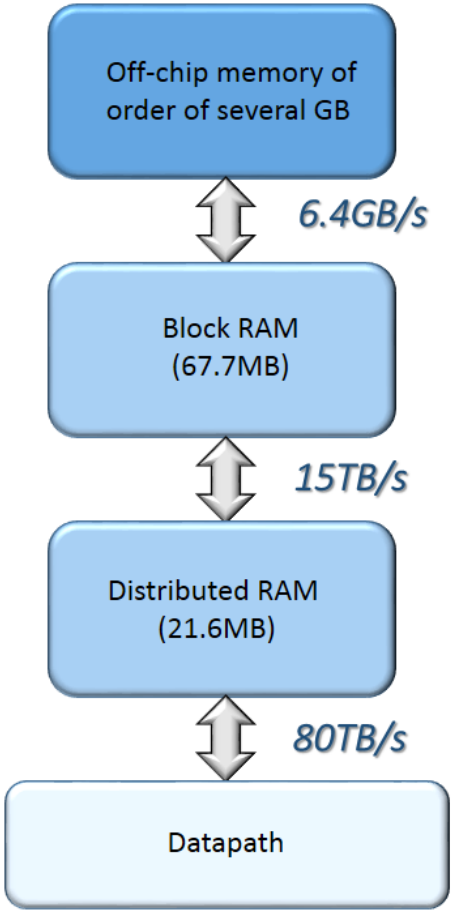
So why are we interested in FPGAs now?

- 10-15 years ago…..
  - Processors
    - Clock rate model
    - Good compiler flow
  - FPGAs
    - Complex glue logic, mostly telecommunications
    - "Special" programming languages
- Now
  - Processors
    - Shift to multi-core
    - Challenges of parallel programming
  - FPGAs
    - Defining SoC technology
    - Highly concurrent machine (but what do we do with it?)

FPGA features

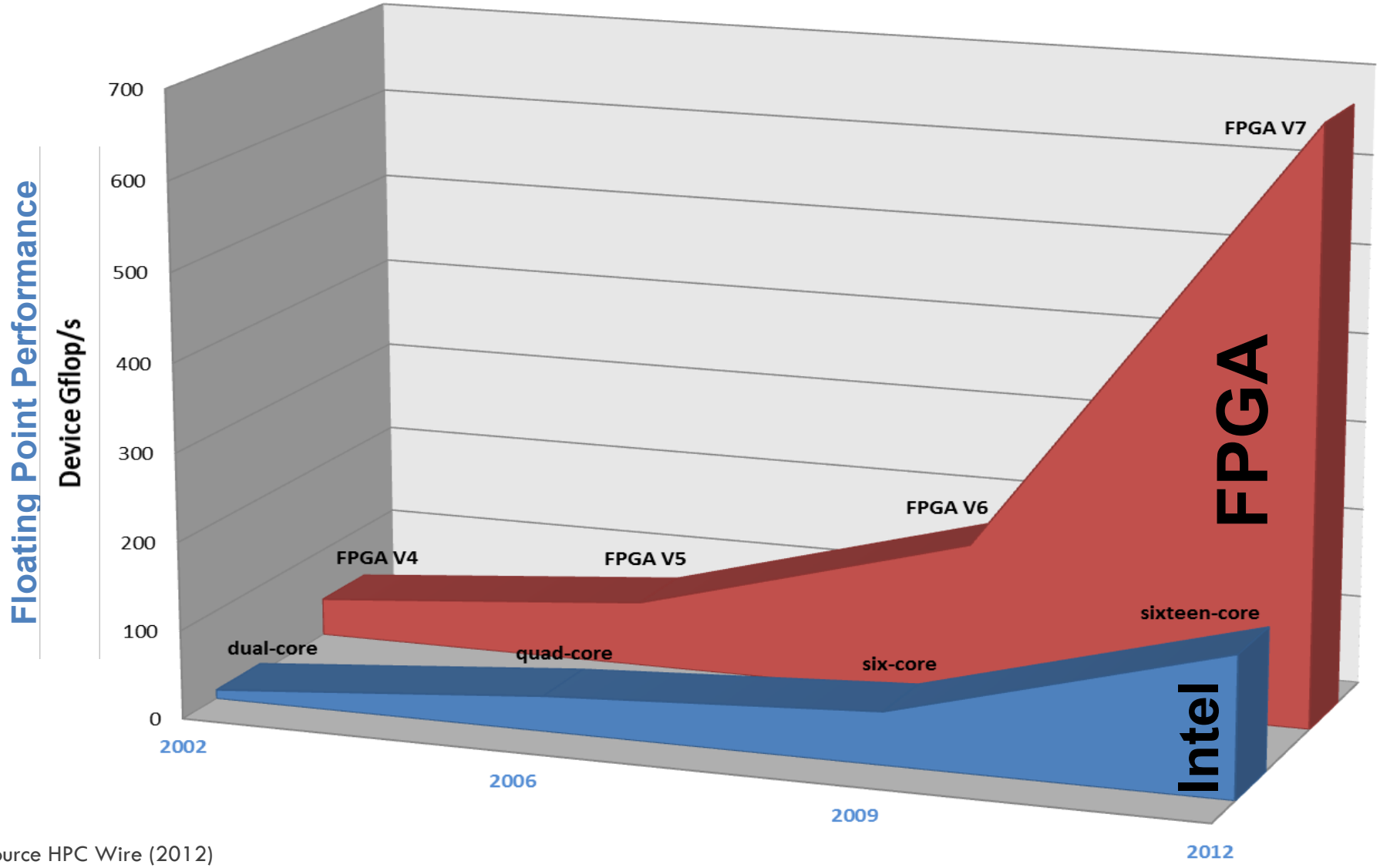


MAC capacity



Memory Structure (Virtex7)

Technology Performance



*Source HPC Wire (2012)

Implications for FPGA design

- High levels of concurrency
  - Scope for very high levels of parallelism
  - Suited to highly pipelined architectures
- Distributed high bandwidth architectures
- Need tight control of memory
  - Memory hierarchy (L1/L2 cache)

Contents

- Technology review
  - FPGA characteristics
  - SoC features
- Uncertainty perspective
  - Design phase and verification
- IPPro example
  - FPGA processor for image processing
  - Design flow
  - IPPro examples
- FPGA/GPU/CPU comparison
  - Heston modelling (Analytics Engines)

Uncertainty

- User Uncertainty
  - FPGA design verification
  - Behavioral simulation to system-level test
  - Design tools assumptions
  - In-circuit verification takes too much time
- Design Uncertainty
  - JTAG strategy
  - Embedded test points - limited access to internal signals

Design verification challenges

- In-circuit verification
  - Long process
  - Every part of the circuit should verified and debugged
- Debug different set of internal signals
  - Need to change your design to route the desired signals for debug purposes
- Design changes
  - Requires re-synthesis and place-and-route process

Removing the uncertainty

- Using optimized, verified, and debugged embedded processors
  - Challenge to use SoC FPGA features effectively
  - Design focus moves to mapping of algorithms to architectures
- Programming environment
  - Needs to maintain a highly parallel, and efficient application implementation
  - Links to programming routes

Architectural Challenges

- FPGAs based soft core processors
  - Granular datapath width
  - Embedded memory hierarchy
  - Broad high speed internal bandwidth
  - High performance computational units e.g. Xilinx DSP48E1

A careful design and architectural choices can result in a small high performance, programmable processor that can be used for different class of applications

- Adopt coarse-grained instead of fine-grained implementation ethos

Contents

- Technology review
  - FPGA characteristics
  - SoC features
- Uncertainty perspective
  - Design phase and verification
- IPPro example
  - FPGA processor for image processing
  - Design flow
  - IPPro examples
- FPGA/GPU/CPU comparison
  - Heston modelling (Analytics Engines)

Image processing algorithms

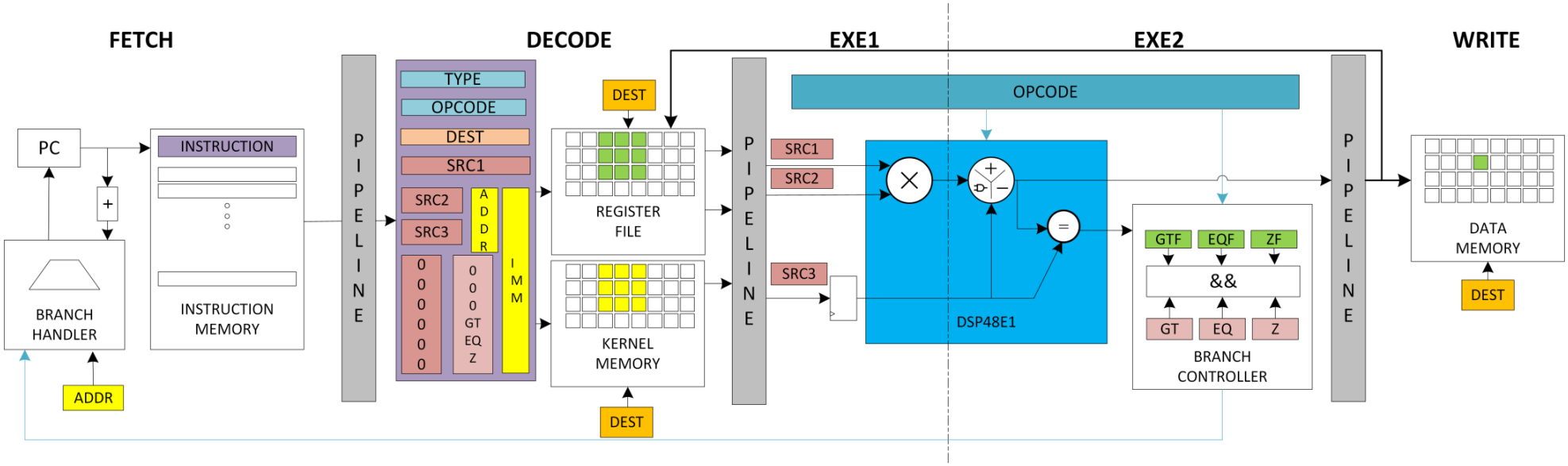
Operation	Domain	Definition
Point	Spatial	Output depends on single input
N'hood/Local	Spatial	Output depends on input & neighbours
Global	Spatial	Output depends on whole image
Geometric	Spatial	Output needs a whole image
Temporal/ frame based	Frequency	Output computed over several video frames

Image Processing Processor (IPPro)

- IPPro is a multi-core processor architecture designed to accelerate image processing applications.
- Image processing applications exhibits large amount of parallelism that can be exploited using Single Instruction Data Multiple Data (SIMD) processing.

Image Processing Processor (IPPro)

- Following is the Single core IPPro datapath
- Multicore IPPro core does not need:
  - Instruction Memory
  - Program Counter



IPPro features

Resource	Size (bits)	Architectural Decision
Local Register File	32x16	• Support different window sizes e.g. 3x3,4x4.. • Fetch 3 operands every cycle to accelerate image pre-processing • Hide data transfer time from main memory
Shared Kernel Memory	32x16	• Store frequently used fixed value, filter kernel to accelerate pre-processing operations • Efficient memory resources utilization • Reduce Code size
Local Data Memory	32x16	• Store intermediate results locally having minimum access time.

IPPro features

Resource	Size (bits)	Architectural Decision
Pipeline Stages	5	• Better performance • Reduced branch penalty compared to deep pipelined processors
Flag bounded instructions	-	• Capable to support control flow in SIMD based multicore architecture • Conditional Execution e.g. MUL, MUL_Z, MUL_EQ, MUL_GT_EQ
Flexible Branch Controller	-	• Easy customizable depending on the conditional execution and flag status

Image Processing Processor (IPPro)

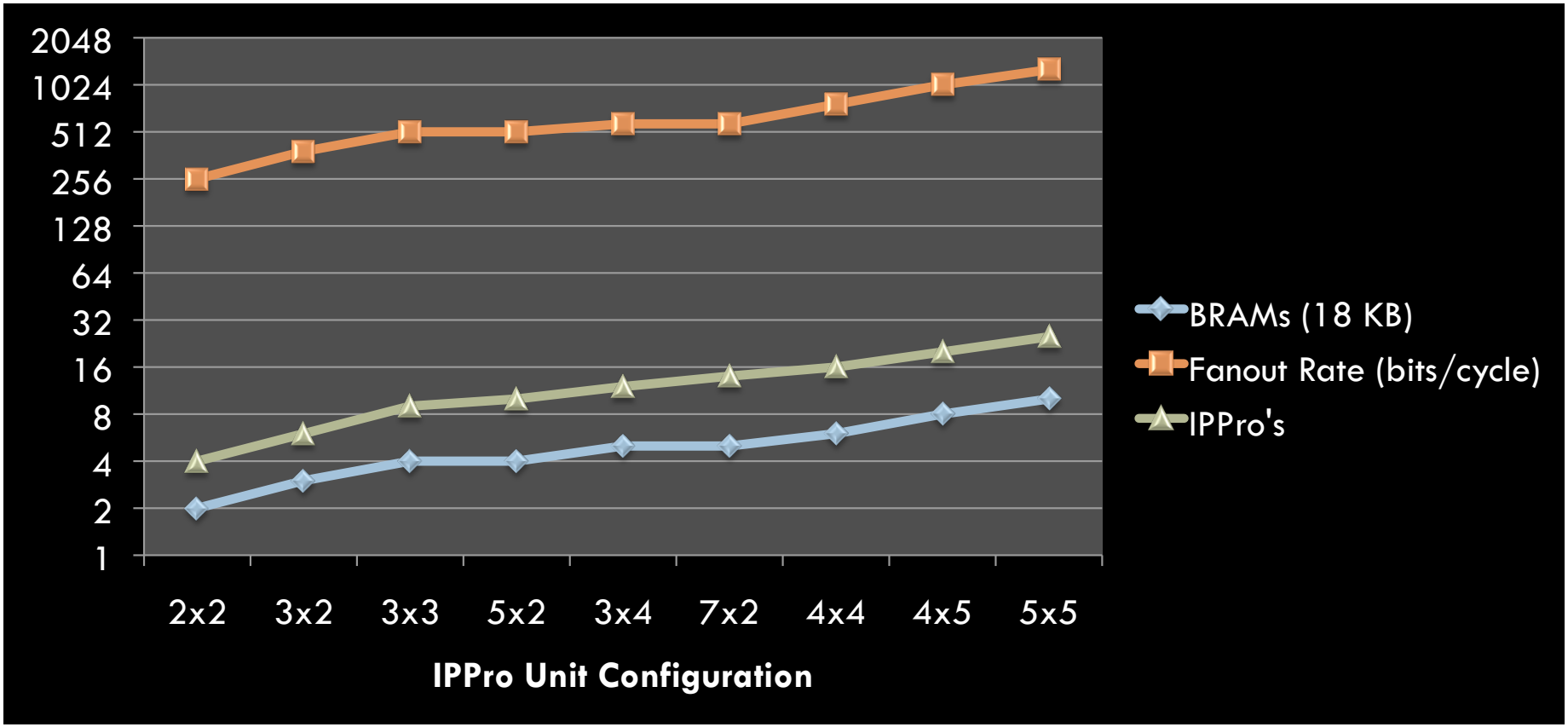
Performance	Virtex-6		Virtex-7
	XC6VLX240T -2	XC6VLX240T -3	XC7VX550T -3
Freq. (MHz)	440	509	526

Area	Virtex-6 XC6VLX240T -3
Slice Registers	227
Slice LUTs	207
DSP48E1	1
BRAM	1

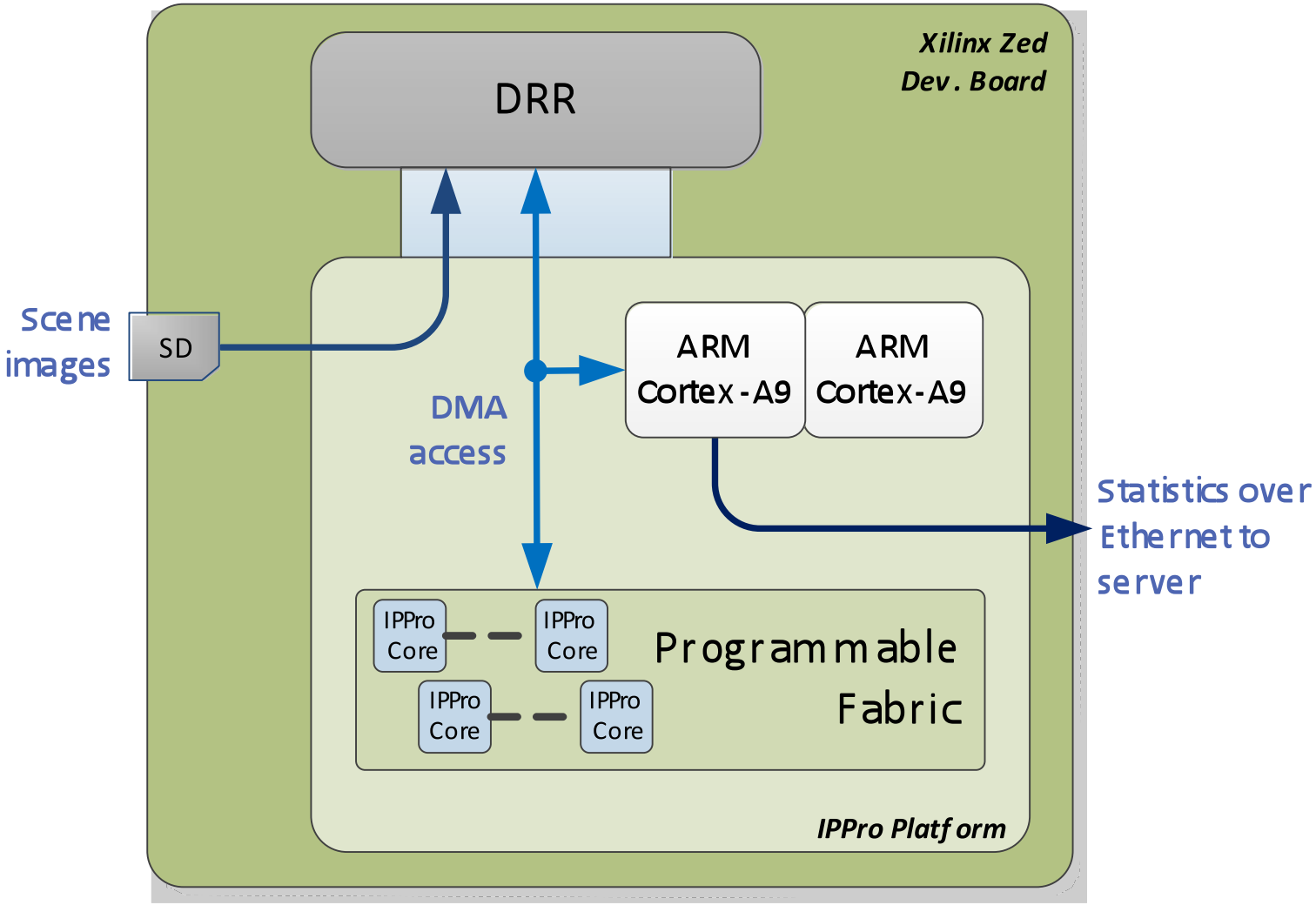
Typical preprocessing operations

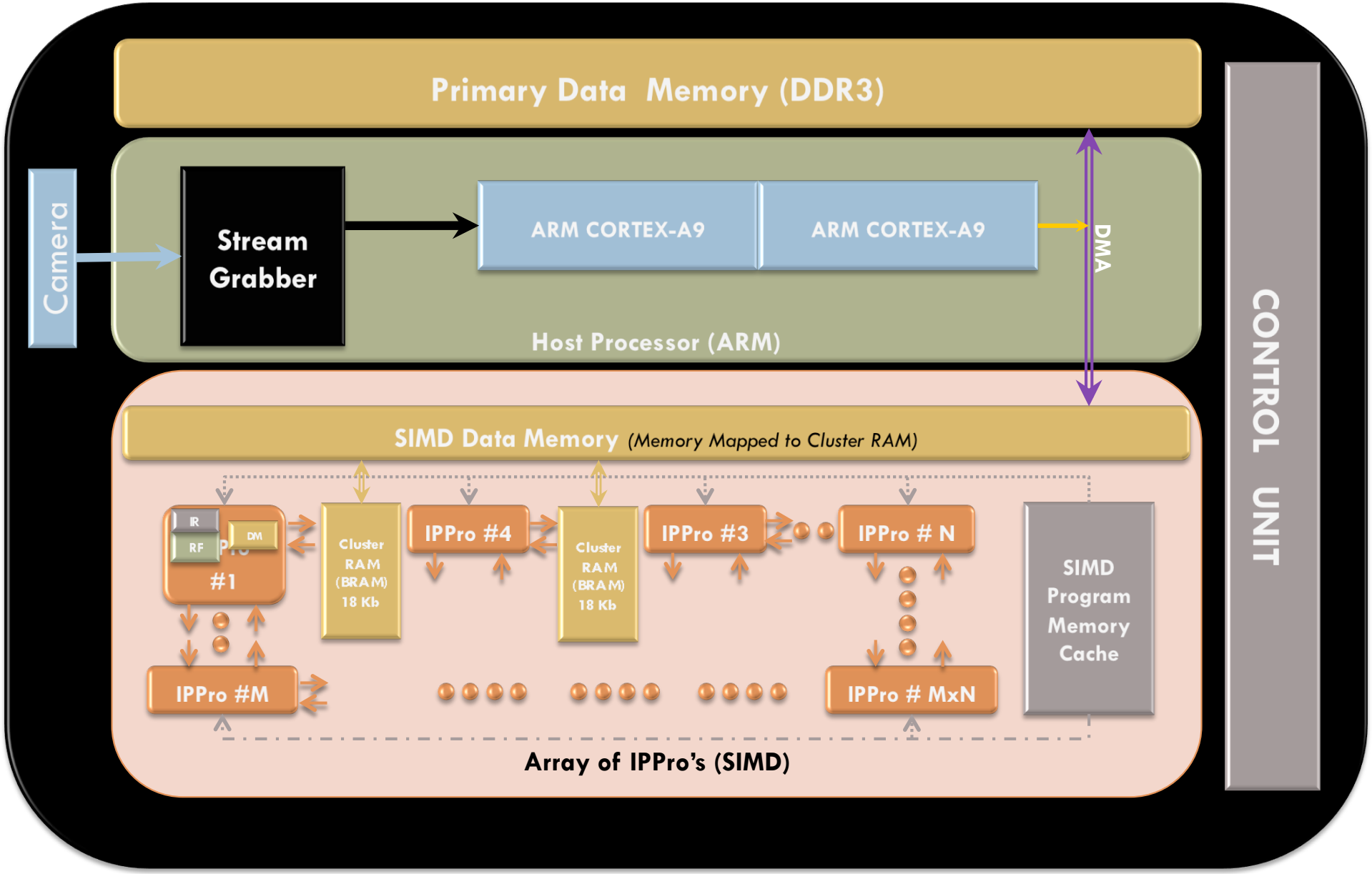
Area Operations	W. Size	2x2, 3x3, 4x4, 5x5, 3x4, 5x4
	1-D Conv.	[A B C]
	2-D Conv.	[A B C; D E F]
	3-D Conv.	[A B C; D E F; X Y Z] Examples: Gaussian, Sobel, Averaging, Median, Laplacian, Emboss, Sharpen
Point Operations	Threshold	O = (A > threshold)? 1 : 0
	Brightness	O = A ± brightness
	Contrast	O = (A – 0.5) * contrast + 0.5
	RGB to Grey scale	O = (2*R + 5*G + 1*B)/8 O = ((R<<2)+((G <<2)+G)+ B)>>3
Geometric Operations	Reflect	$x2 = x1 \;;\; y2 = -y1 + (2 * y0)$ $y2 = y1 \;;\; x2 = -x1 + (2 * x0)$
	Translate	$x2 = x1 + \beta x$ $y2 = y1 + \beta y$

IPPro core configuration

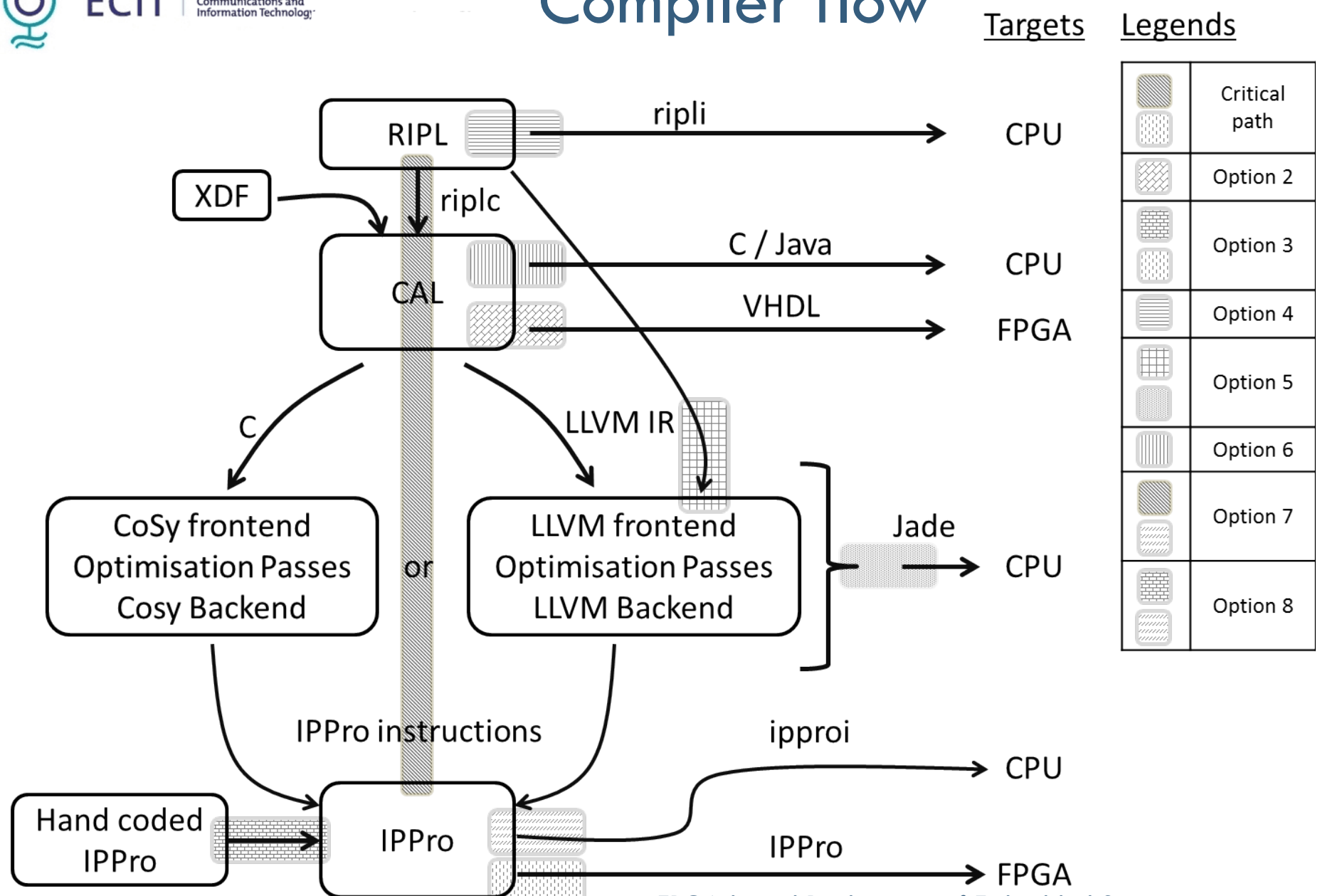


Zedboard prototyping environment




Primary Data Memory (DDR3)
Camera
Stream Grabber
ARM CORTEX-A9
ARM CORTEX-A9
DMA
Host Processor (ARM)
CONTROL UNIT
SIMD Data Memory (Memory Mapped to Cluster RAM)
IR
RF
DM
#1
Cluster RAM (BRAM) 18 Kb
IPPro #4
Cluster RAM (BRAM) 18 Kb
IPPro #3
IPPro # N
SIMD Program Memory Cache
IPPro #M
IPPro # MxN
Array of IPPro's (SIMD)

Compiler flow

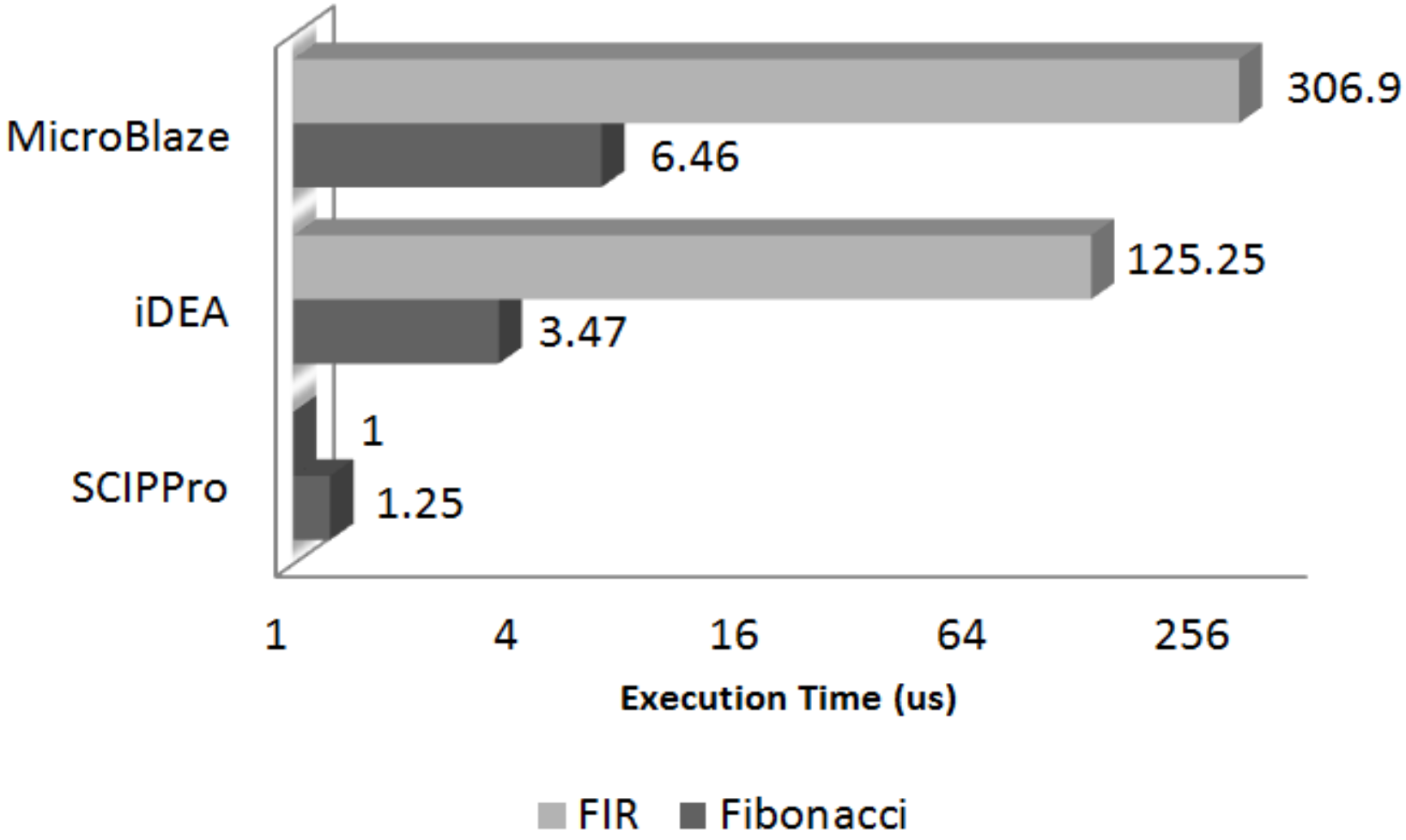


IPPro examples

- Two examples:
  - 5 stage FIR filter
  - Fibonacci series calculation (0,1,1,2,3,5,8,13,21..)

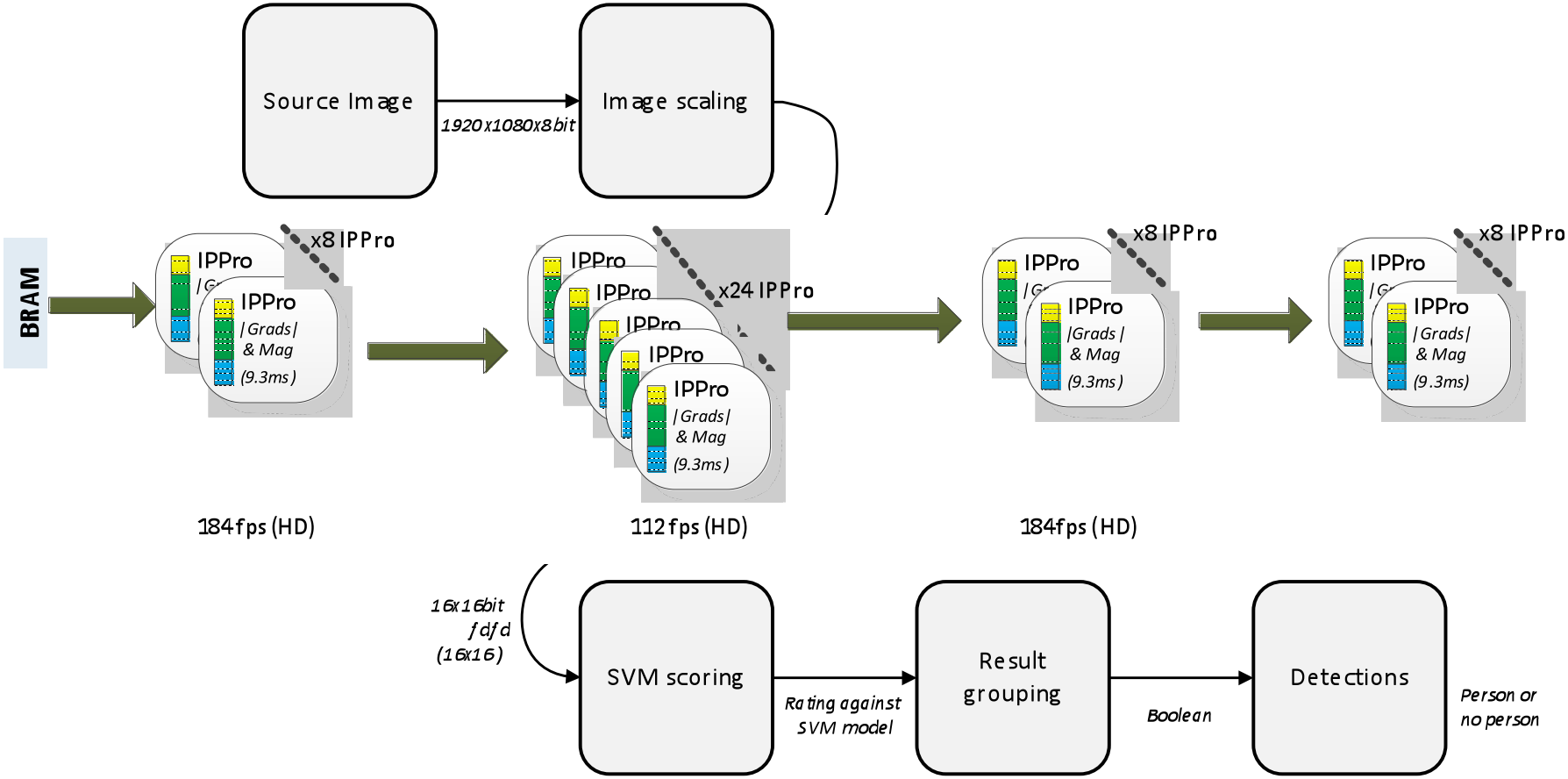
Application	Inst. Count	Clock Cycles	# of nops	Exe. Time(us)
Fibonacci				
SCRIP	253	553	295	1.25
SCRIP(Delay.S)	253	353	95	0.80
iDEA	256	1,414	1149	3.47
MicroBlaze*	1085	1115	25	5.31
FIR				
SCRIP	355	360	0	0.81
iDEA	8,609	51,121	42503	125.25
MicroBlaze*	2976	2990	9	14.2

IPPro examples



Extension to larger examples

- Histogram of gradients



Contents

- Technology review
  - FPGA characteristics
  - SoC features
- Uncertainty perspective
  - Design phase and verification
- IPPro example
  - FPGA processor for image processing
  - Design flow
  - IPPro examples
- FPGA/GPU/CPU comparison
  - Heston modelling (Analytics Engines)

Analytics Engines

Hardware Accelerated Analytics for Big Data

- Fast analytics kernels running on acceleration cards
- Performance beyond software with higher data volumes at lower cost

Key Benefits

- Off loading from data server CPUs to accelerator card for reduced work load
- Potential 10x to 100x performance improvement (over CPU/ GPU) for high throughput low latency processing
- Boosting mid-high range server performance and reuse of legacy infrastructure

Big Data

Analytics Engines

- Big data will reach $32.4 billion in 2017, growing 6 times faster than the overall IT market*.

- Internet of Things
  - resulting information networks promise to create new business models
  - improve business processes
  - reduce costs and risks.

- Could create 58,000 jobs in Britain by 2017, contributing £216bn to the country's economy.

* IDC

Capabilities

Analytics Engines

CPU	Acceleration of up to 4X can be achieved though combining our IP and optimisation of existing software code.
GPU	Improvements in excess of 100X can be delivered by offloading specific processes on to GPU hardware.
FPGA	Speedups of 2900X have been demonstrated by integrating our FPGA IP with existing software systems.

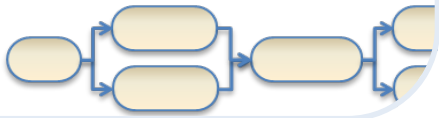
Capability Areas

Analytics Engines

Kernel Design

- Design of high performance data processing applications

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi \frac{k}{N} n}.$$



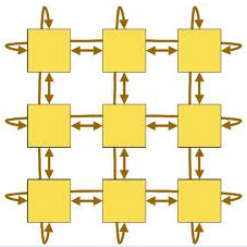
Prototype

- Rapid prototype and proof-of-concept in SW/HW

```
FFT(X) {
  n=length(x);
  if (n==1) return x;
  m = n/2;
  X = (x_{2j})_{j=0}^{m-1};
  Y = (x_{2j+1})_{j=0}^{m-1};
  X = FFT(X);
  Y = FFT(Y);
  U = (X_{k mod m})_{k=0}^{n-1};
  V = (g^{-k}Y_{k mod m})_{k=0}^{n-1};
  return U+V;
}
```

Parallelization

- Algorithms to massively parallel implementation



Delivery

- Enterprise platform acceleration

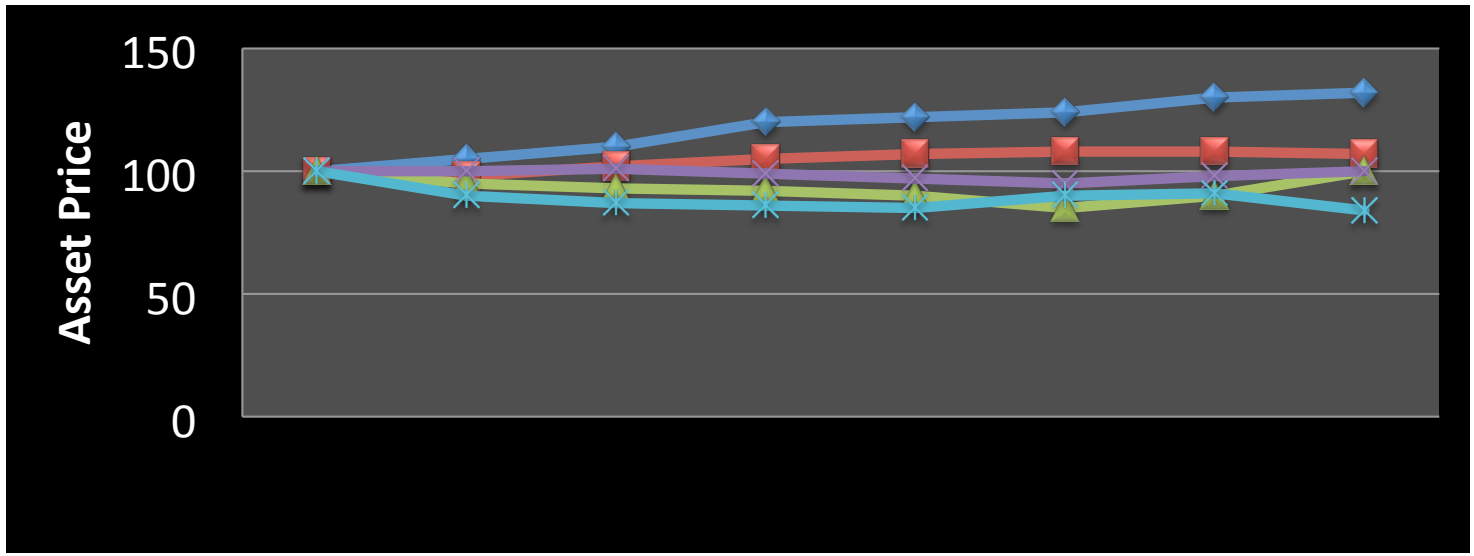
Heston Model Detail

- **Black-Scholes** algorithm has been used as a means of computing risk however it does **not allow volatility.**

- **Different market behaviours** which give cause this volatility. These include:
  - ***Leverage Effects:*** A firm's equity value is calculated as the net present value of all of its future income. This must account for all assets and debits, all of which can have different relative volatilities.
  - ***Supply and Demand:*** Downward insurance is vital as equities tend to be held for longer so downward rather than upward protections become more important.
  - ***A Decline in stock price:*** Declining stock prices are more volatile than increasing stock prices and will affect the portfolio rebalancing.

Heston Model Detail

- Components:



Heston Model

Analytics Engines

- Main computational bottleneck in Heston model
- Benchmarking across multiple architectures

	CPU (std::rand ())	SIMD CPU (SSE4/Well512)	GPU (GTX 660)	FPGA (Virtex 7)
Uniform numbers per second	120×10^6	700×10^6	10×10^9	128×10^9
Speedup		X6	X83	X1000

- Not the full story
  - FPGA can be pipelined
  - Throughput may not decrease as computational steps added (e.g. normal transform)

Summary

- Highlighted the processor capability of FPGA – hints a shift into FPGA design capabilities.
- Increased shift towards domain specific design tools
- Application to initial designs examples – extension to more complex image processing
- Immense interest in data processing/analytics – possible fruitful area for FPGA