Is High Level Synthesis Ready for Business?

A Computational Finance Case Study

Gordon Inggs, Shane Fleming, David Thomas and Wayne Luk

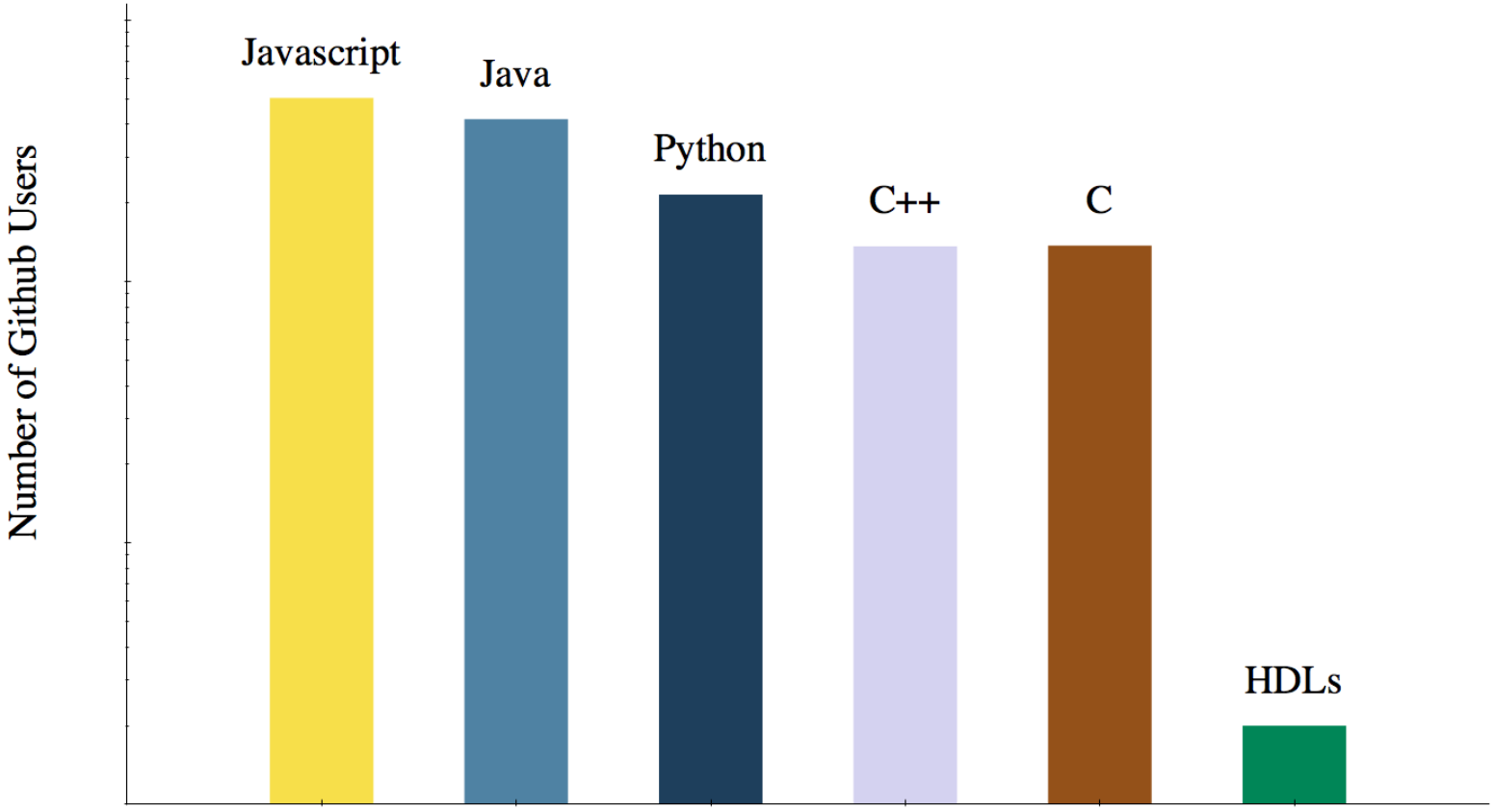
Circuits and Systems Group
Department of Electrical and Electronic Engineering

Custom Computing Group
Department of Computing

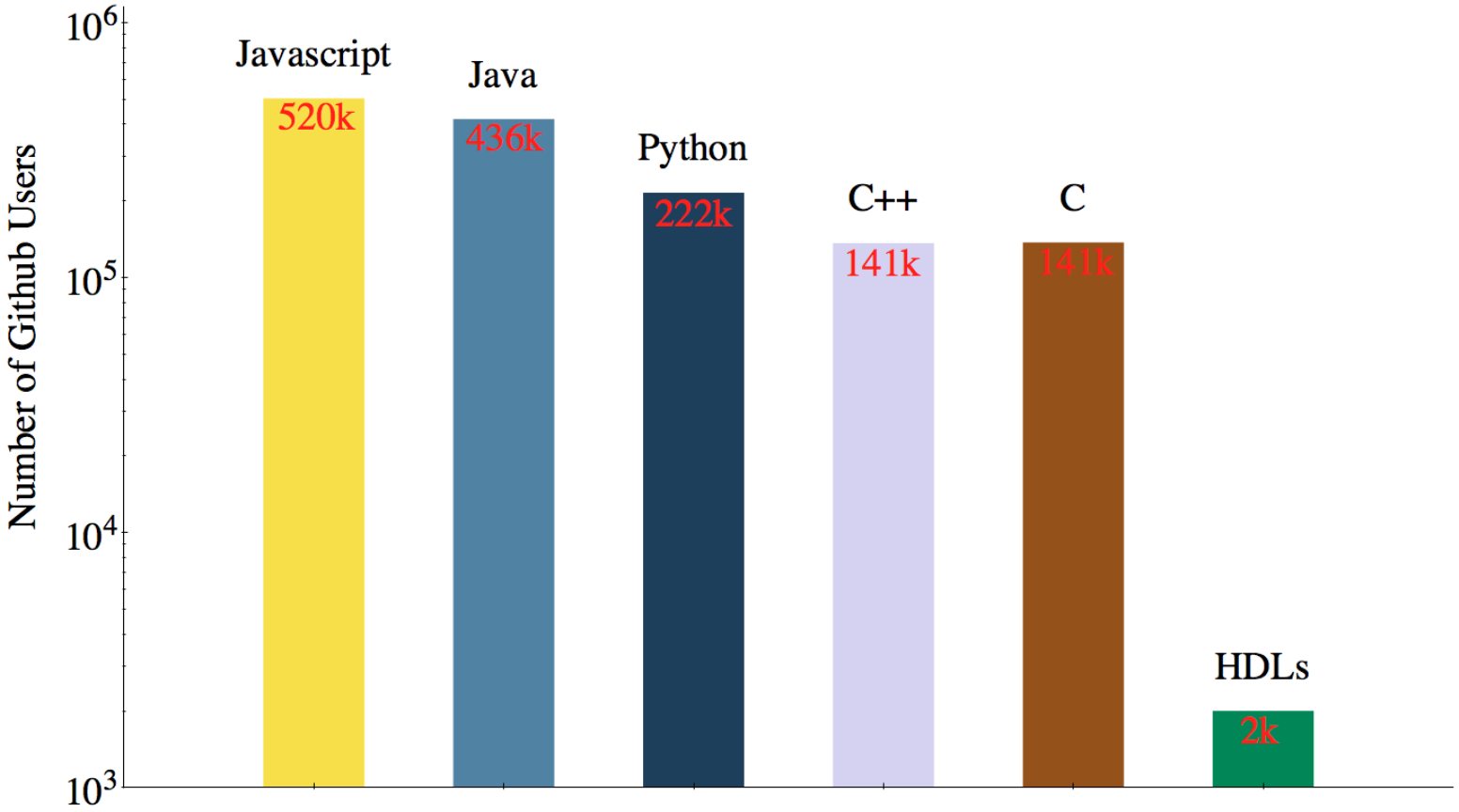
Imperial College London

- Who is HLS for?
- And what do they want with it?
- How good is HLS?
  - Development experience
  - Performance of designs
    - Standalone
    - comparatively

Who is HLS for?



GitHub Users (as of December 2014)



GitHub Users (as of December 2014)

Meet Jo

- Works in a large investment bank

- Experienced software engineer:
  - Optimisation
  - Multicore CPUs, GPUs, etc.

Jo’s Requirements for HLS

- Integrate with existing software framework
- Compete with other accelerator technologies
- HLS ecosystem, not tools alone

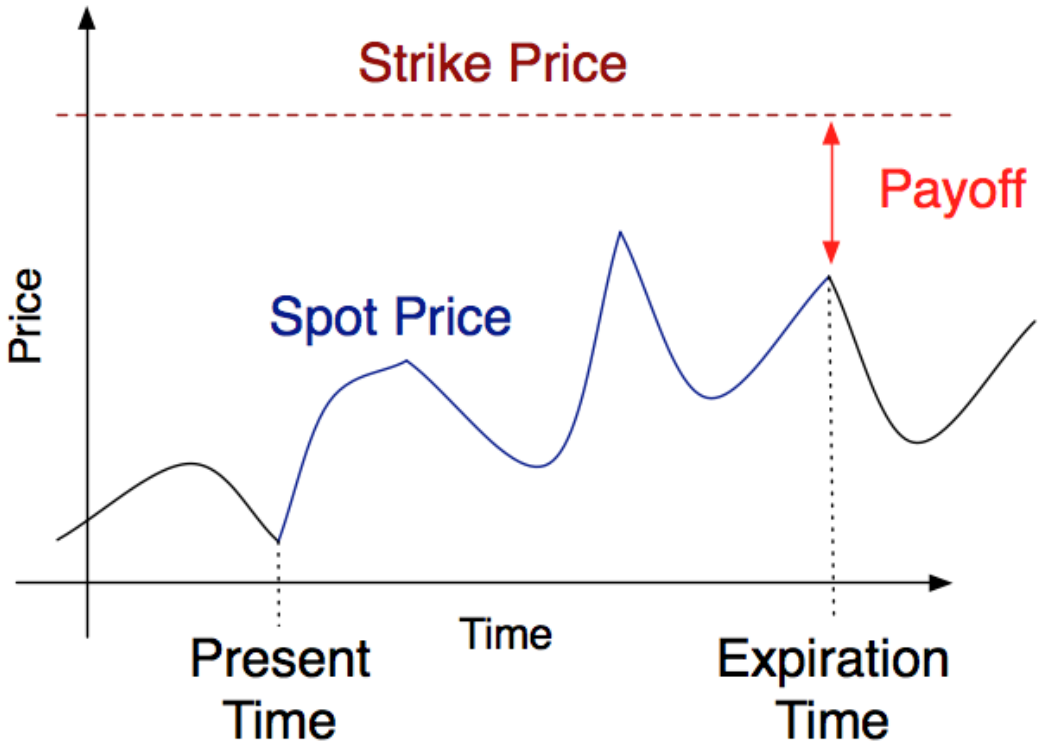
Our Contributions

1. Evaluate 3 HLS Tools quantitatively and qualitatively

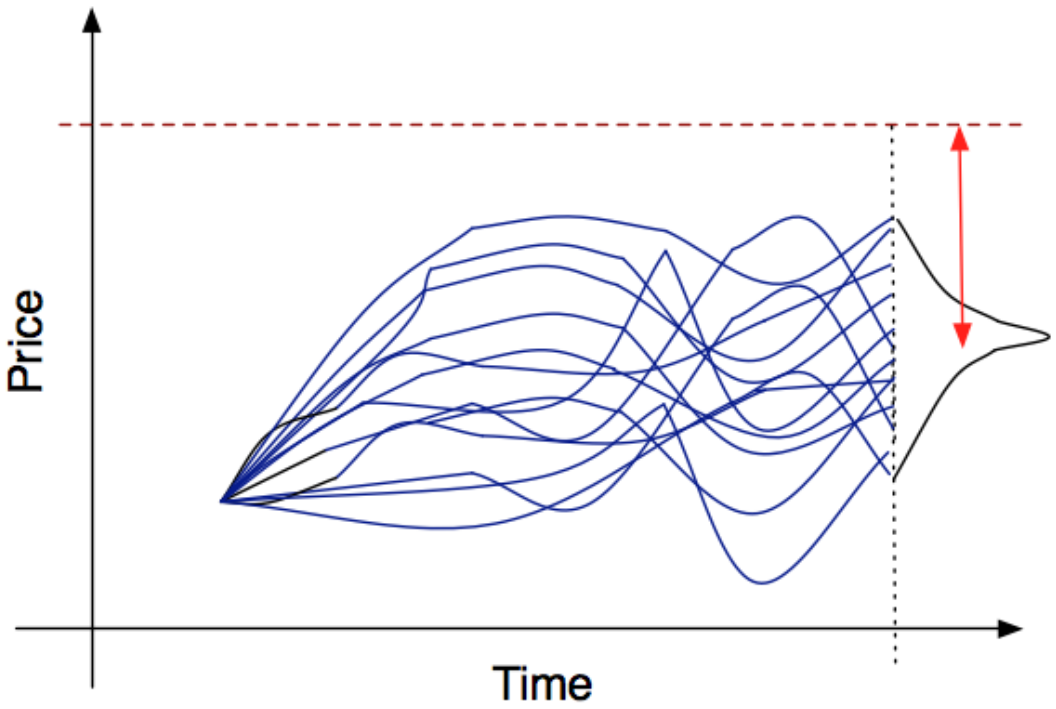
2. Compare to competing accelerator technologies - Multicore CPUs, GPUs and hybrid architectures

3. Comment on programming paradigm

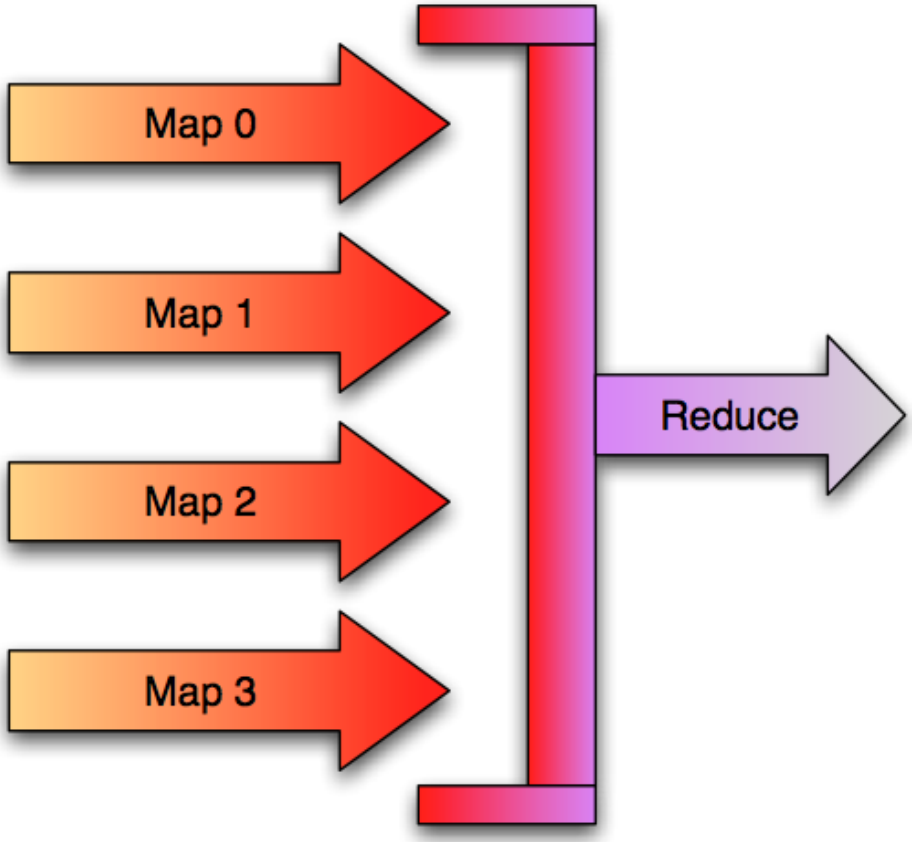
What does Jo want with it?



Overview of forward looking option pricing



Monte Carlo option pricing algorithm



MapReduce programming pattern

Monte Carlo Pricing as MapReduce

```
//Map
for(i=0;i<PATHS;++i)
{
 state = path_init(seed+i);

 for(j=0;j<PATH_POINTS;++j)
    state = path(state);

 value[i] = payoff(state);
}
```

```
//Reduce
for(i=0;i<PATHS;++i)
{
    result += value[i]/PATHS;
}
```

Monte Carlo Pricing for FPGAs

- *Tse, Thomas, et al.*, Imperial College London, 2011
  - Black-Scholes model-based Asian option
  - **313x speedup**, 2.76x energy saving over single CPU

- *de Schryver, Shcherbakov, et al.*, Technische Universität Kaiserslautern, 2011
  - Heston model-based barrier options
  - **26x speedup**, 1.8x energy saving over 2-core CPU

Jo’s HLS Test Suite

- Portfolio of options:
  - 12 Kaiserslautern Heston benchmark options
  - Imperial College Black-Scholes Asian option

- Large calculation:
  - 10 million simulations
  - 4096 discretisation points per simulation

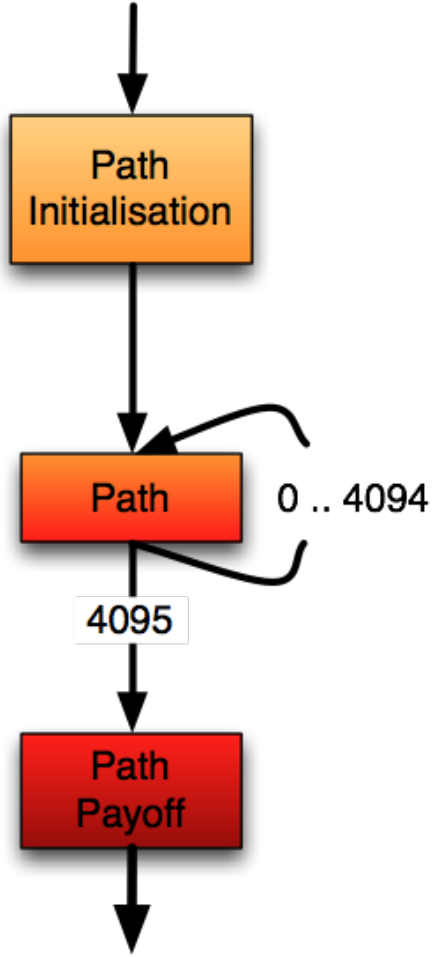
Jo's HLS Tools

- Xilinx Vivado HLS
- Maxeler MaxCompiler
- Altera OpenCL SDK

How would Jo use HLS Tools?

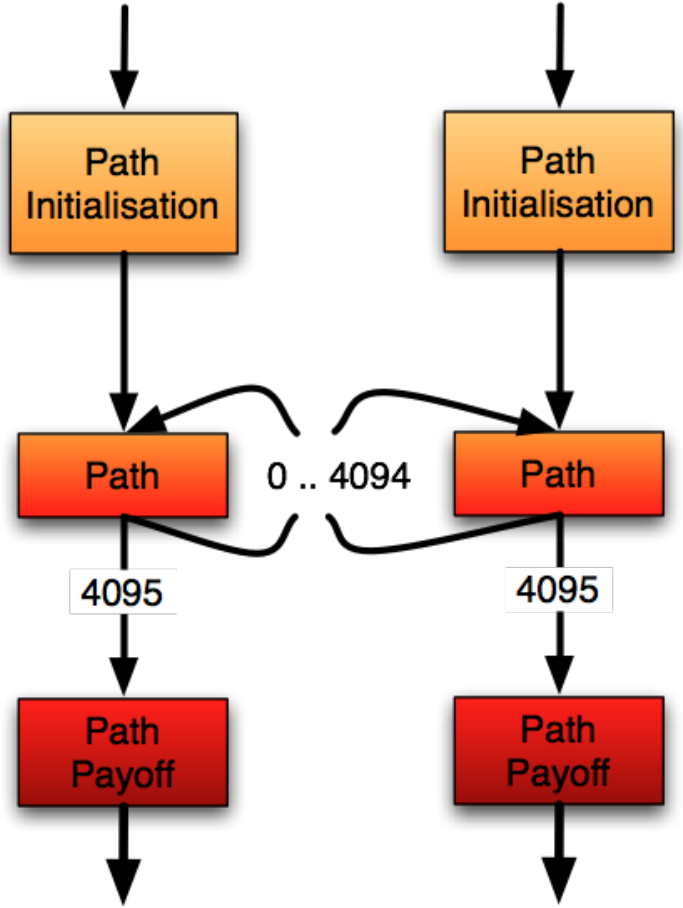
Jo's Initial Attempt

```
//Map on FPGA via HLS
for(i=0;i<PATHS;++i)
{
 state = path_init(seed+i);

 for(j=0;j<PATH_POINTS;++j)
    state = path(state);

 value[i] = payoff(state);
}
```



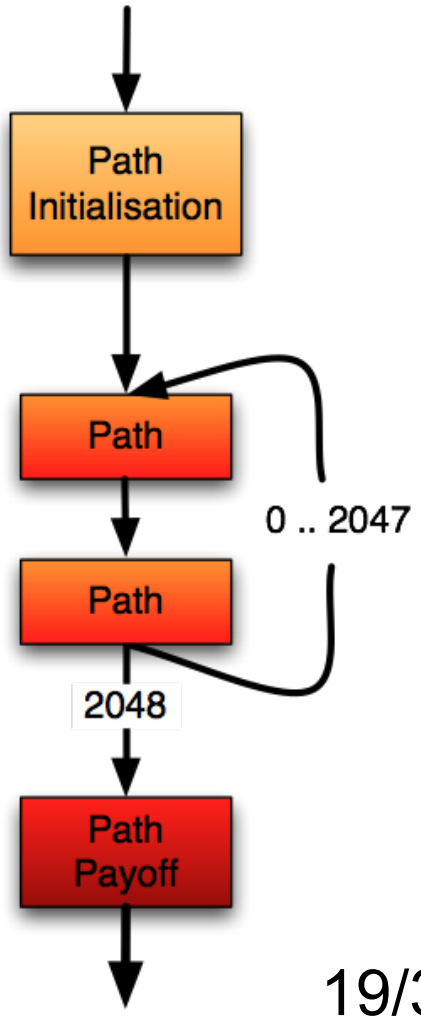
Jo's Task Parallelism

```
for(k=0;k<P;++k){
    for(i=0;i<PATHS;i+=P)
    {
     state = path_init(seed+i+k);
     for(j=0;j<PATH_POINTS;++j)
         state = path(state);

     value[i+k]=payoff(state);
    }
}
```



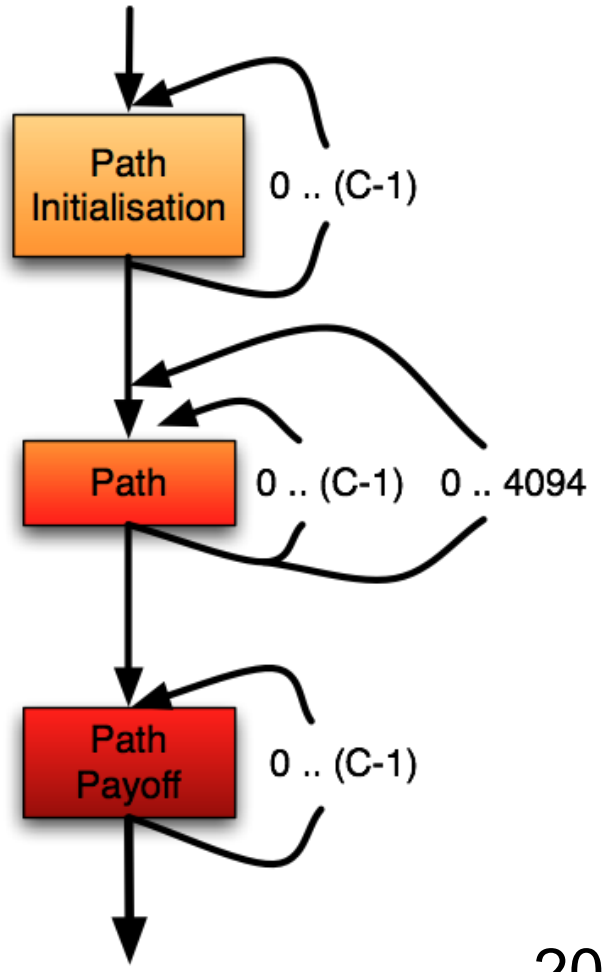
Jo's Pipeline Parallelism

```
for(i=0;i<PATHS;++i)
{
 state = path_init(seed+i);

 for(j=0;j<PATH_POINTS;j+=2)
 {
    state2 = path(state);
    state = path(state2);
 }
 value[i] = payoff(state);
}
```



Jo's C-Slowing

```
for(k=0;k<PATHS/C;k++)
{
 for(i=0;i<C;++i)
    state[i] = path_init(seed+k*C+i);

 for(j=0;j<PATH_POINTS;++j)
    for(i=0;i<C;++i)
        state[i] = path(state[i]);

 for(i=0;i<C;++i)
    value[k*C+i] = payoff(state[i]);
}
```



Jo’s Evaluation

- Development experience of ecosystem
- Handwritten HDL from literature
- Competing accelerator technologies:
  - Multicore CPUs
  - GPUs
  - Hybrid architectures

Jo’s Experimental Platforms

Jo’s HLS Platforms

Platform Vendor	Name	FPGA	Communication Technology	HLS Tool
Xilinx	ZC706 1.1	Xilinx Zynq 7Z045	AXI	Xilinx VivadoHLS 2013.4
Nallatech	P385-A7	Altera Stratix V GXA7	PCIe	Altera OpenCL SDK 13.0
Maxeler	Max 3424A	Xilinx Virtex 6XC6VSX475T	PCIe	Maxeler MaxCompiler 13.2

Jo’s FPGAs

FPGA	CMOS (nm)	Targeted Clockrate (MHz)	Lookup Tables	FlipFlop Registers	Block Rams	Digital Signal Processors
Xilinx Zynq 7Z045	28	100	218.6k	437.2k	545	900
Altera Stratix V GXA7	28	**250**	**622k**	**939k**	**2304**	768
Xilinx Virtex 6XC6VSX475T	**45**	200	297.6k	595.2k	1064	**2016**

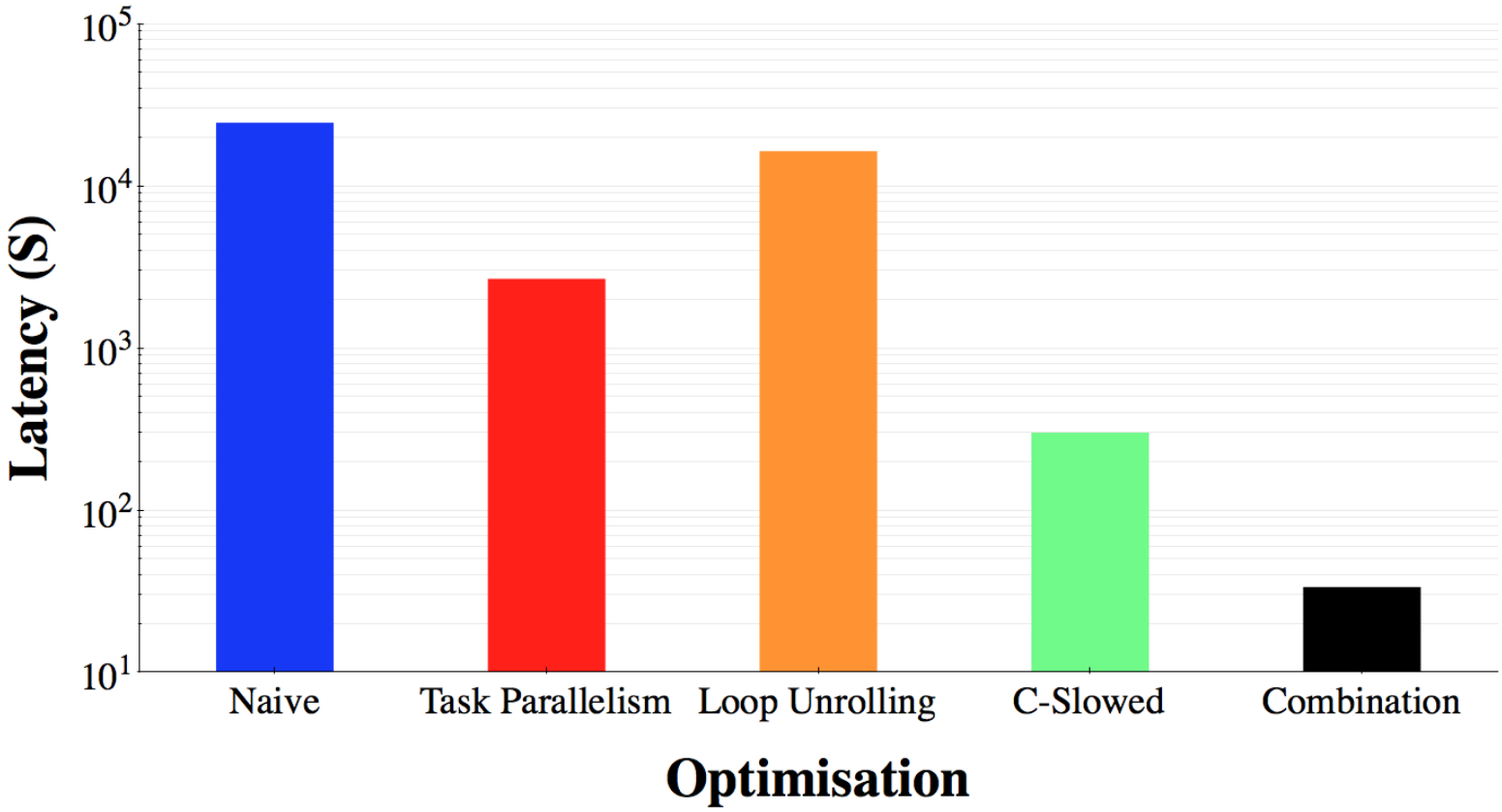
Jo’s Competing Accelerators

Platforms	CMOS Size (nm)	Clockrate (GHz)	Floating Point Parallelism	Tool
Intel Core i7-2600S	32	**2.8**	1	GCC 4.8
AMD Opteron 6272	32	2.1	64	GCC 4.8
AMD Firepro W5000	28	0.825	**768**	AMD OpenCL SDK 2.9
Intel Xeon Phi 3120P	**22**	1.1	256	Intel OpenCL SDK 2014

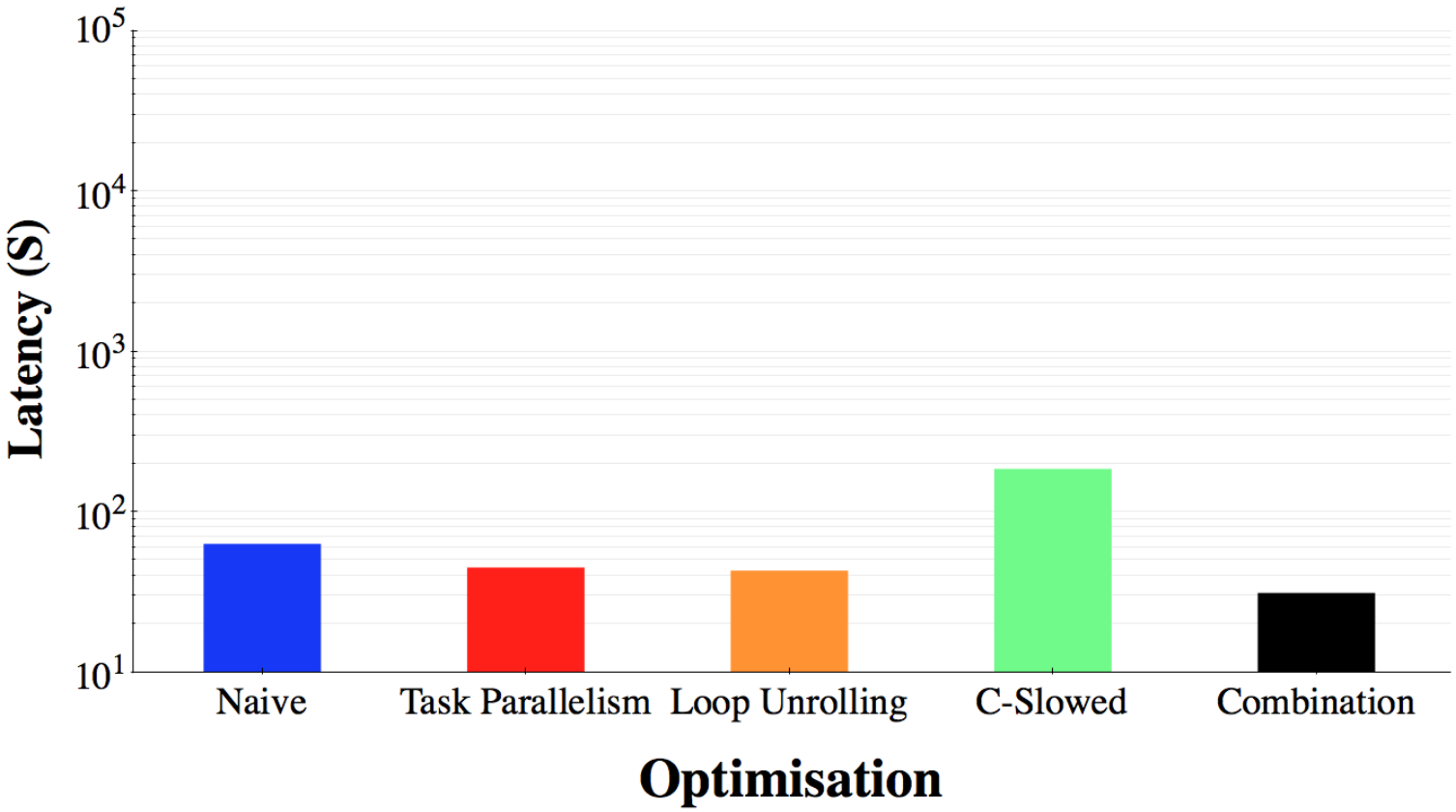
Jo’s Experimental Results

Tool	Lines of Code	Development Time	Integration Time	Compile Time
Maxeler MaxCompiler	**130**	4 weeks	**2 weeks**	**2 - 48**
Altera OpenCL SDK	72	1 weeks	0.5 weeks	2 - 8
Xilinx Vivado HLS	63*	**12 weeks**	1 week	2 - 6

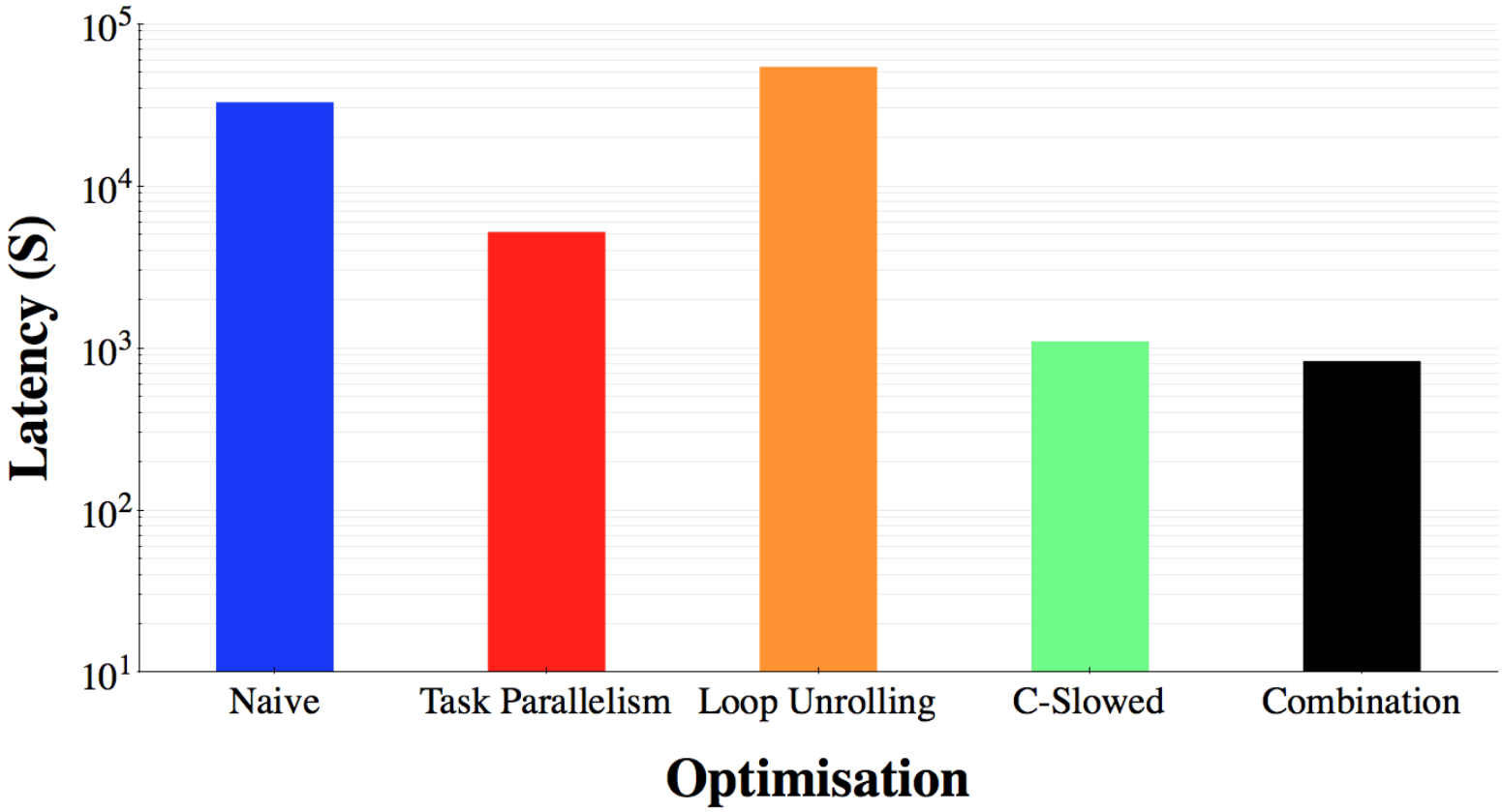
Development Experience



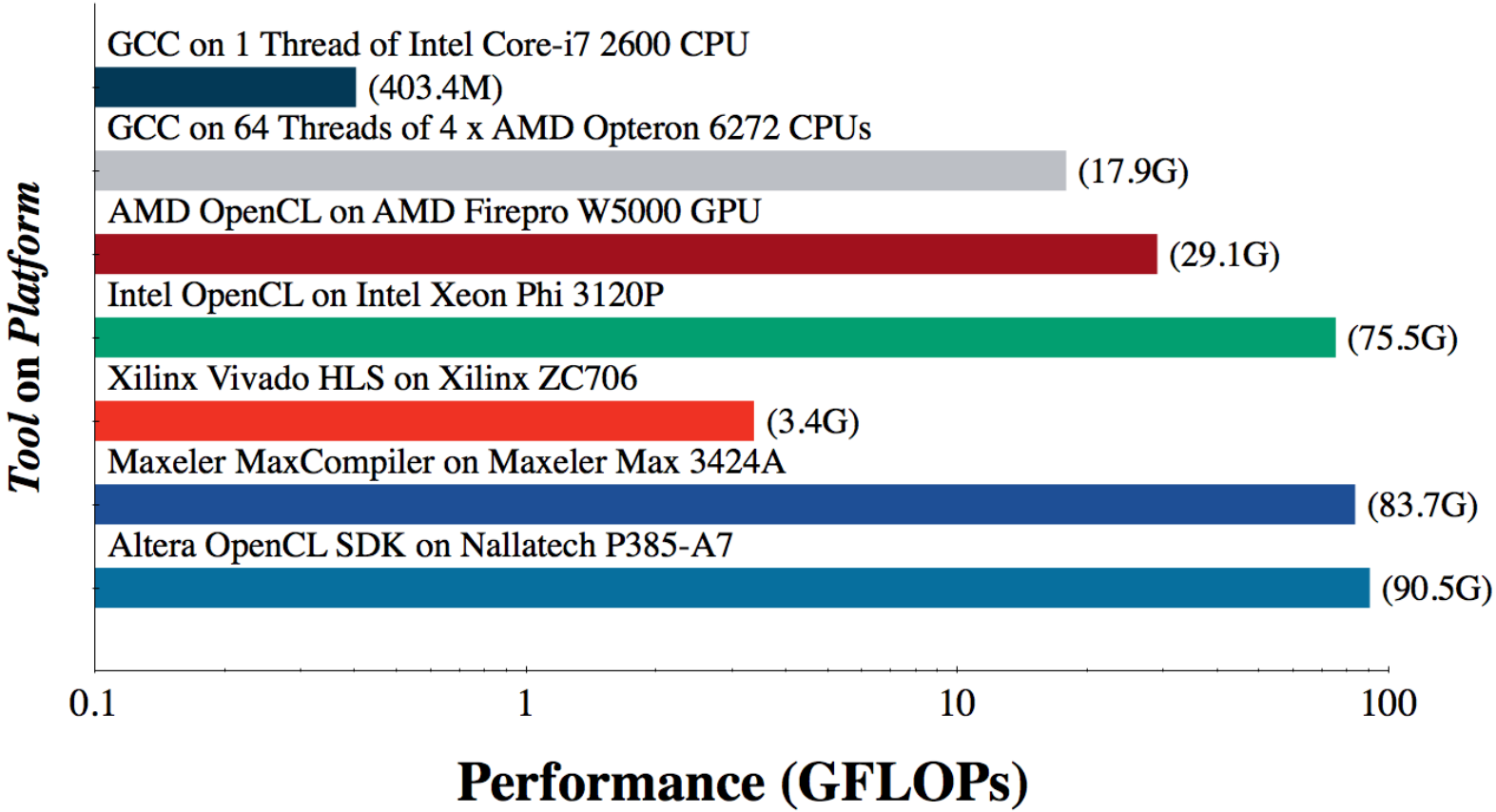
Maxeler MaxCompiler Latency Results



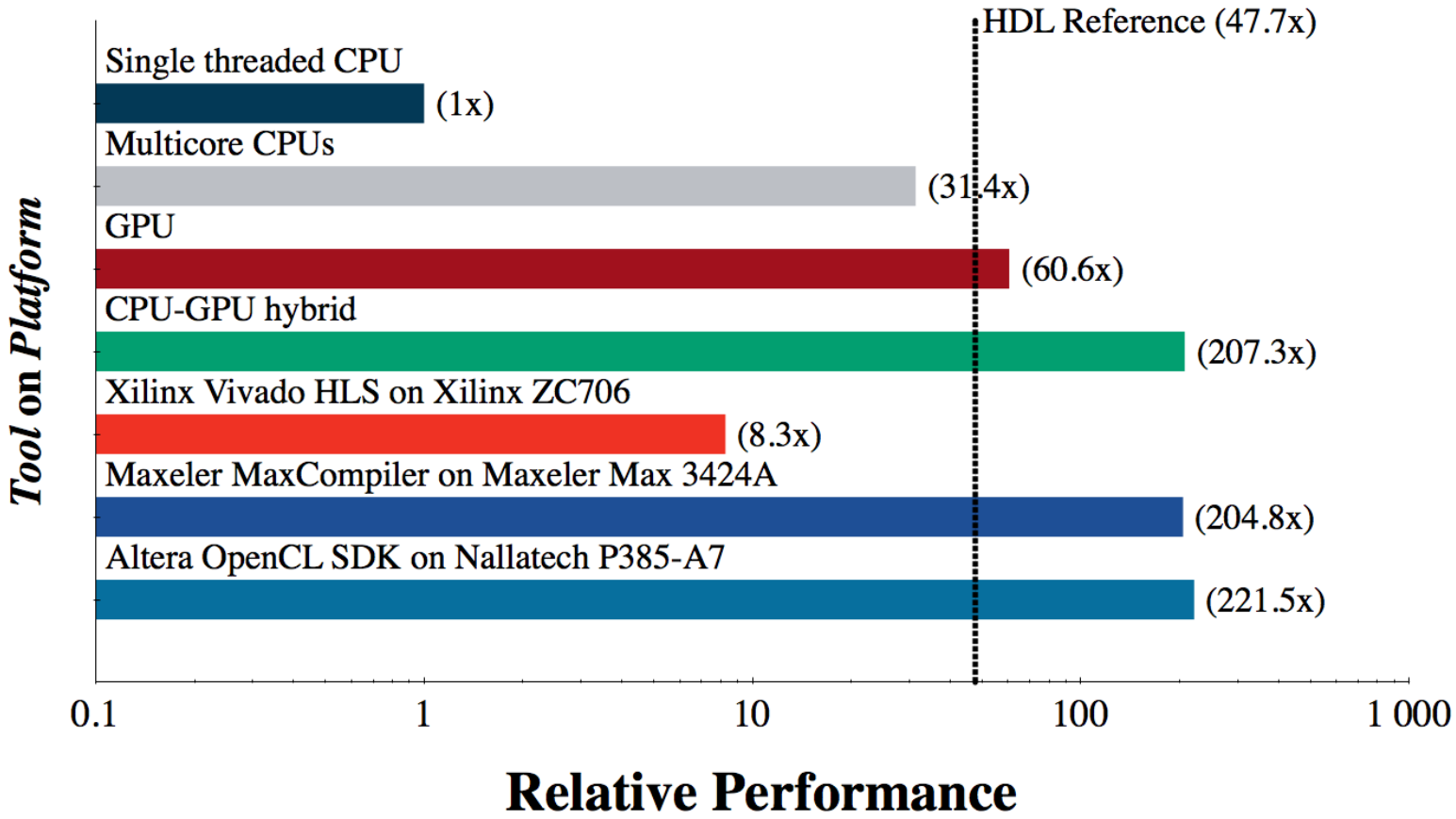
Altera OpenCL SDK Latency Results



Xilinx Vivado HLS Latency Results



Absolute Performance of HLS and Competing Platforms



Relative Performance of HLS and Competing Platforms

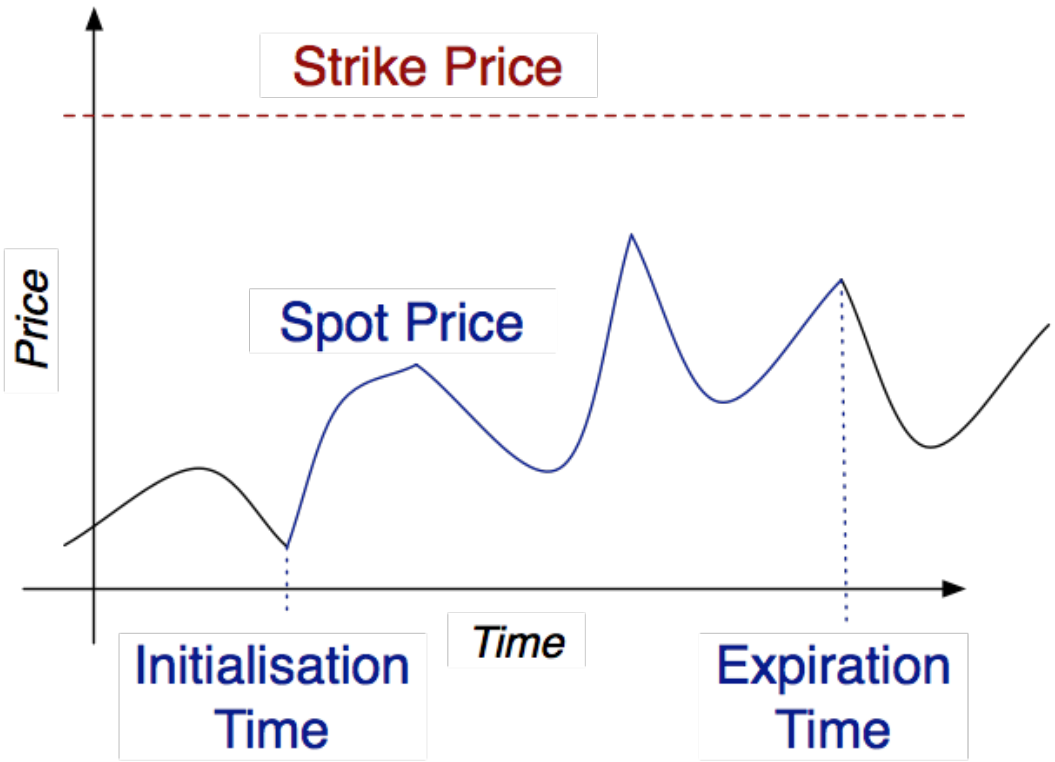
What would Jo Conclude?

- HLS is indeed ready for business
- Programming model matters
- ToDo:
  - Wider characterisation: energy, accuracy
  - Wider set of tools, especially Open Source

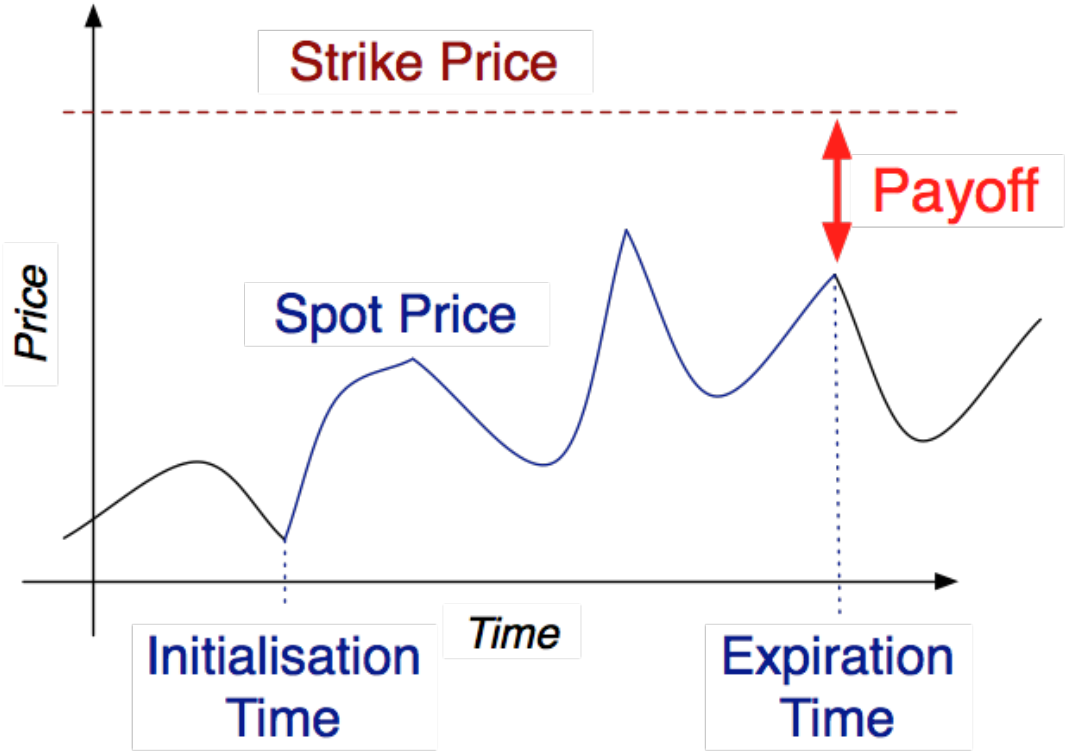
Thank You

Maxeler, Altera and Xilinx University Programs
Nallatech
UCT’s RRSG
South African National Research Foundation
Oppenheimer Memorial Trust

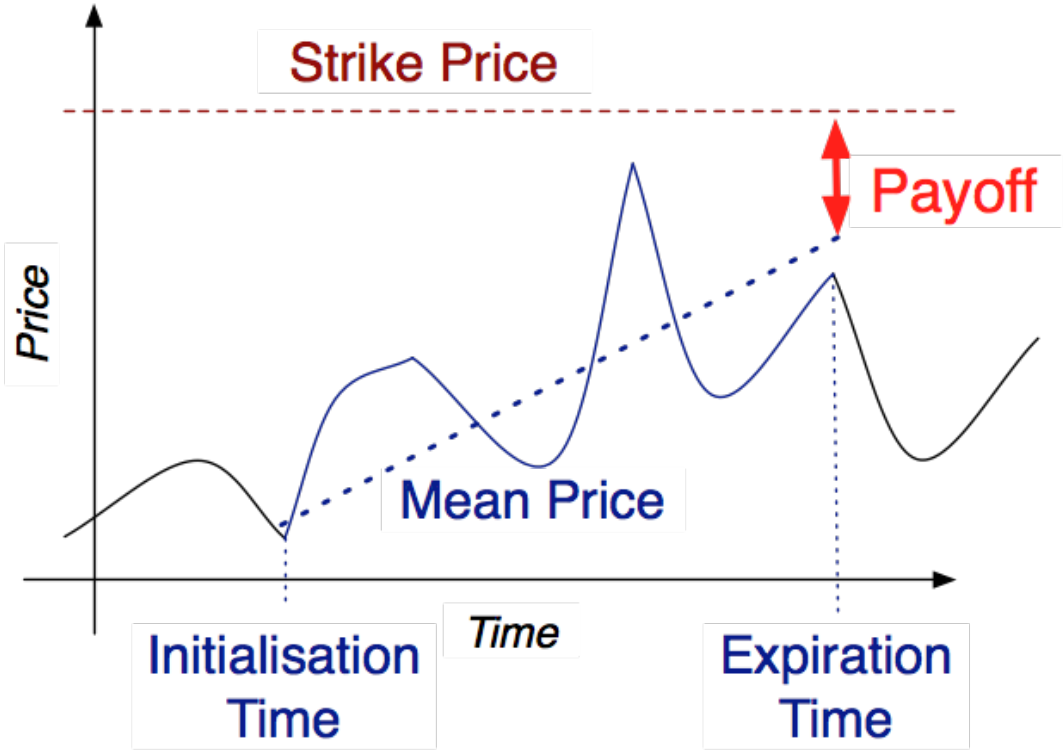
Overview of Option Pricing



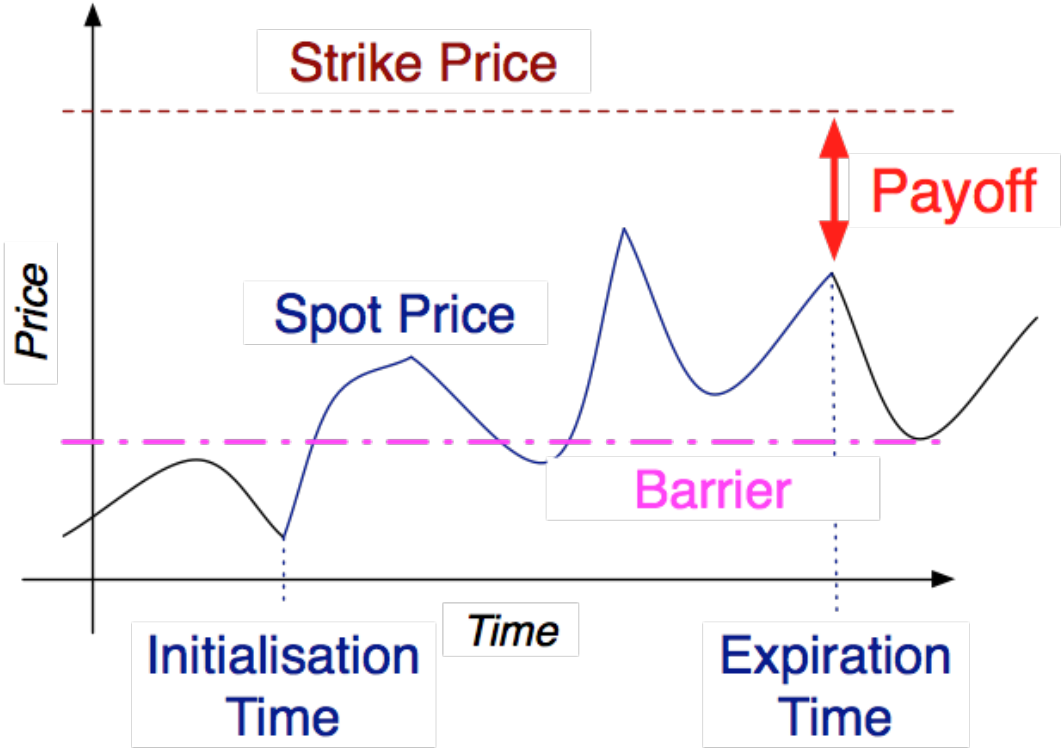
Overview of Option Pricing



European Option Pricing



Asian Option Pricing



Barrier Option Pricing

Jo’s HLS Platforms

Platform Vendor	Name	FPGA	Communication Technology	HLS Tool
Xilinx	ZC706 1.1	Xilinx Zynq 7Z045	AXI	Xilinx VivadoHLS 2013.4
Nallatech	P385-A7	Altera Stratix V GXA7	PCIe	Altera OpenCL SDK 13.0
Maxeler	Max 3424A	Xilinx Virtex 6XC6VSX475T	PCIe	Maxeler MaxCompiler 13.2

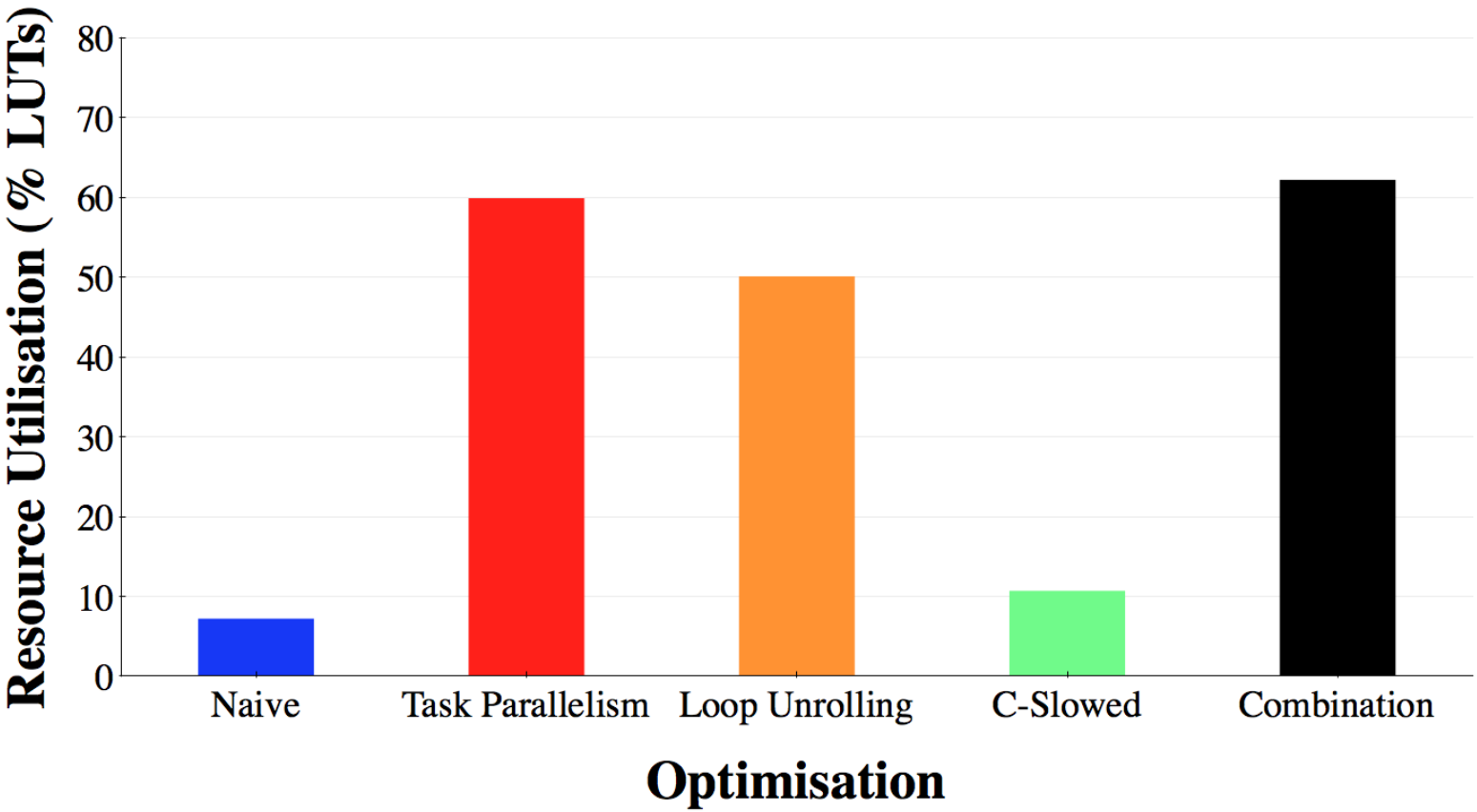
Jo’s FPGAs

FPGA	CMOS (nm)	Targeted Clockrate (MHz)	Lookup Tables	FlipFlop Registers	Block Rams	Digital Signal Processors
Xilinx Zynq 7Z045	28	100	218.6k	437.2k	545	900
Altera Stratix V GXA7	28	250	622k	939k	2304	768
Xilinx Virtex 6XC6VSX475T	45	200	297.6k	595.2k	1064	2016

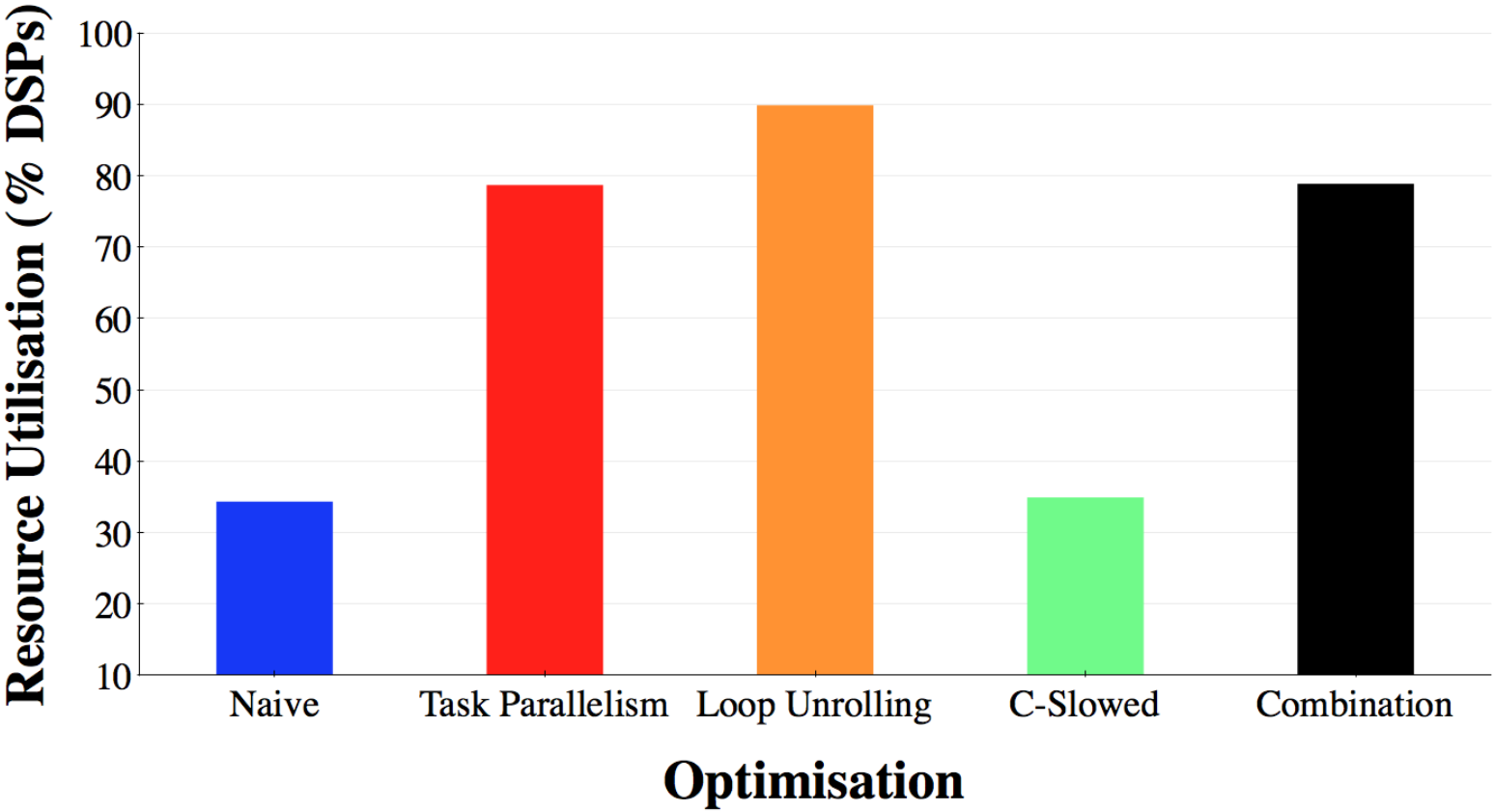
Jo’s Competing Accelerator Platforms

Platforms	CMOS Size (nm)	Clockrate (GHz)	Main Memory (GBs)	Threads	Tool
Intel Core i7-2600S	32	2.8	16	1	GCC 4.8
AMD Opteron 6272	32	2.1	128	64	GCC 4.8
AMD Firepro W5000	28	0.825	2	768	AMD OpenCL SDK 2.9
Intel Xeon Phi 3120P	22	1.1	6	256	Intel OpenCL SDK 2014

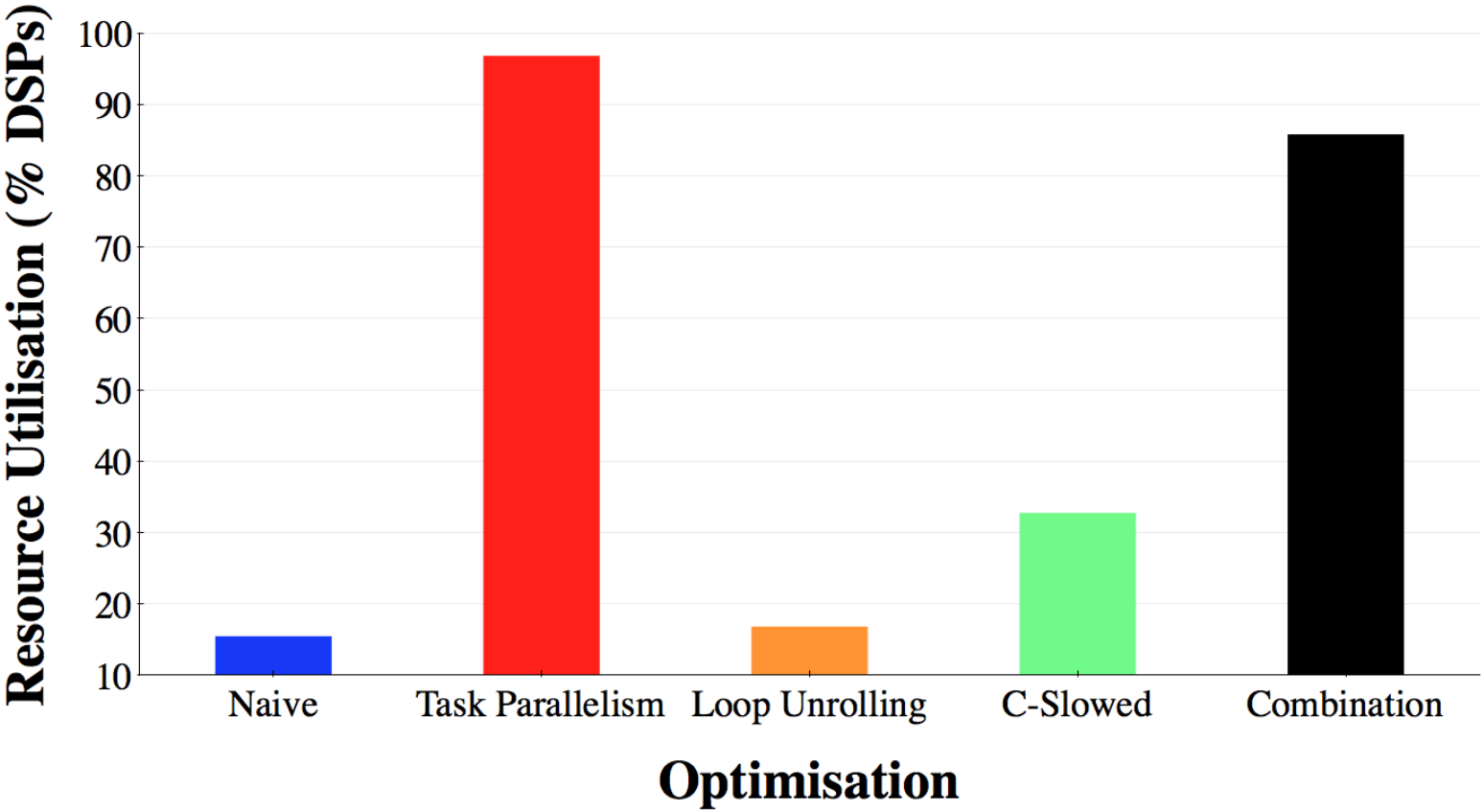
HLS Area Results



Maxeler MaxCompiler Resource Utilisation Results



Altera OpenCL SDK Resource Utilisation Results



Xilinx Vivado HLS Resource Utilisation Results