OpenCL Tutorial

David Castells-Rufas
Microelectronics & Electronics Systems Department
Universitat Autònoma de Barcelona
david.castells@uab.cat

Centre de Prototips i Solucions Hardware - Software
Center for Hardware – Software Prototypes & Solutions

FPGAs & Energy Efficiency

History of Energy Efficiency (G from Greeness)

- Computing Eras
- Operations per Joule

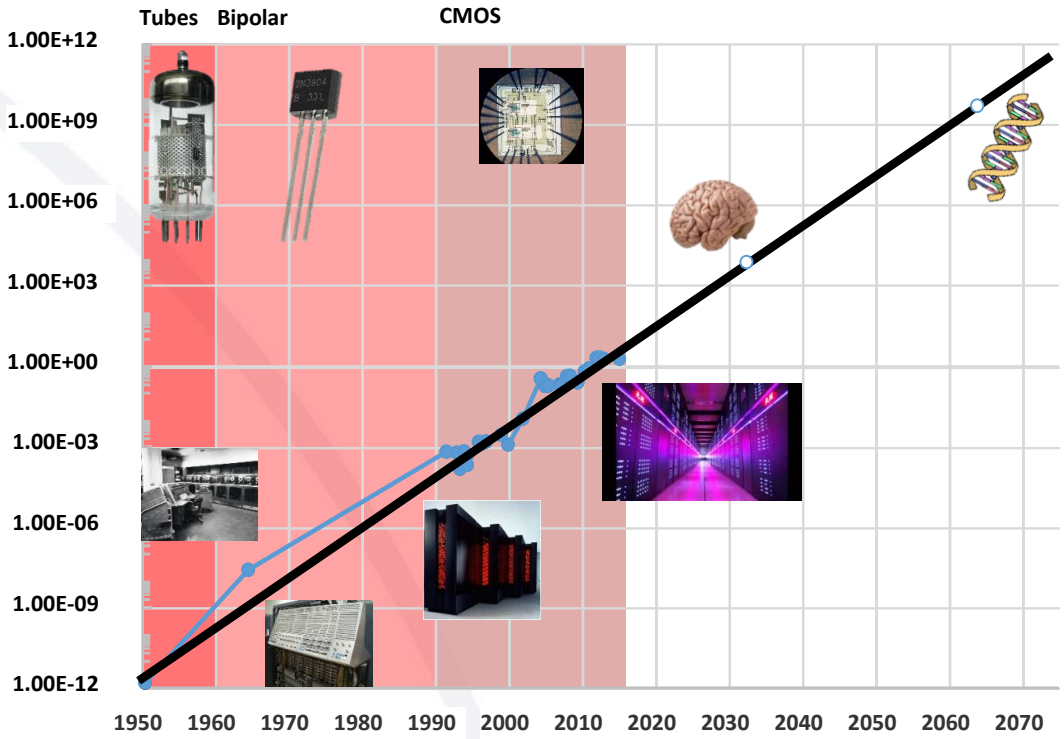
$$G = \frac{Op}{E} = \frac{Op}{T}\frac{1}{P}$$

- OPS per Watt

  THE GREEN 500 uses GFLOPS/W

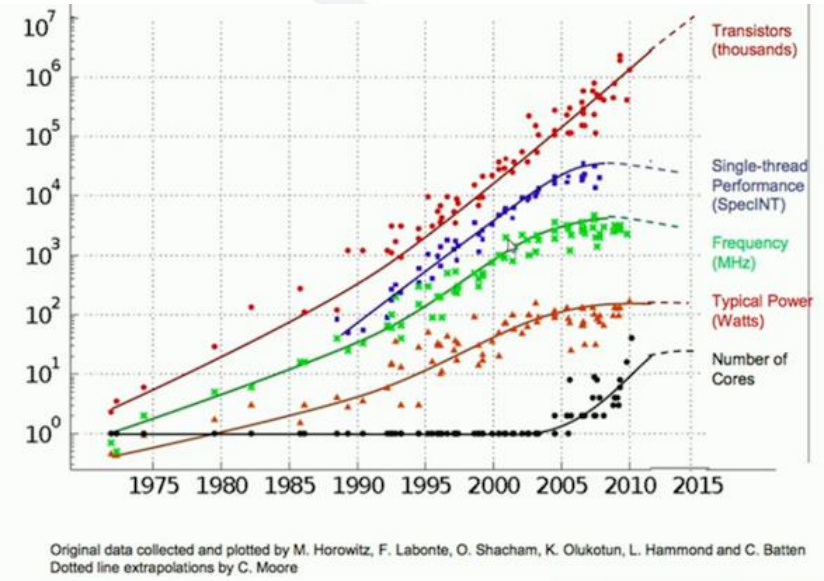
  Fujitsu A64FX=16.9 GFlops/Watt

- 12 orders of magnitude in 65 yr.
  - –Disruptive changes
  - –Architectural changes



Rules changed before 2010

- Dennard Scaling Rules were no longer valid
- Thermal Density

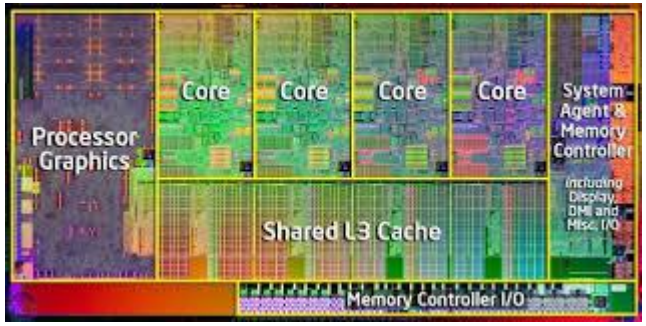


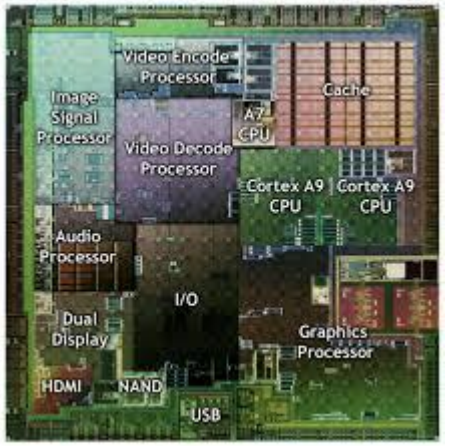
David Castells Rufas Ph.D. Thesis Defense 8/2/20

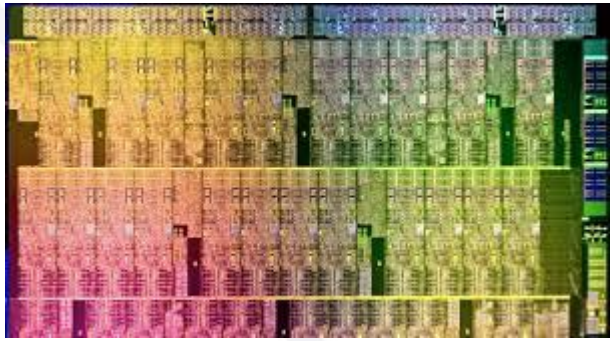
	Dennard (1974)	Taylor (2013)
Tran. Size tox, W, L	$1/s$	$1/s$
Devices	s^2	s^2
Voltage V	$1/s$	1
Current I	$1/s$	1
Capacitance C	$1/s$	$1/s$
Intrinsic delay CV/I	$1/s$	$1/s$
Power dissipation VI	$1/s^2$	1
Power density	1	s^2
Frequency	s	1

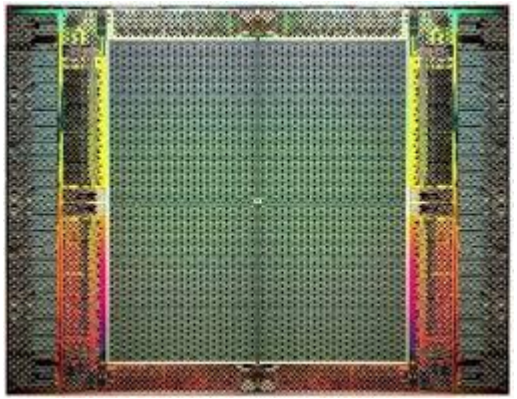
Alternatives

- Multicores
- HW Coprocessors
- GPUs
- FPGAs

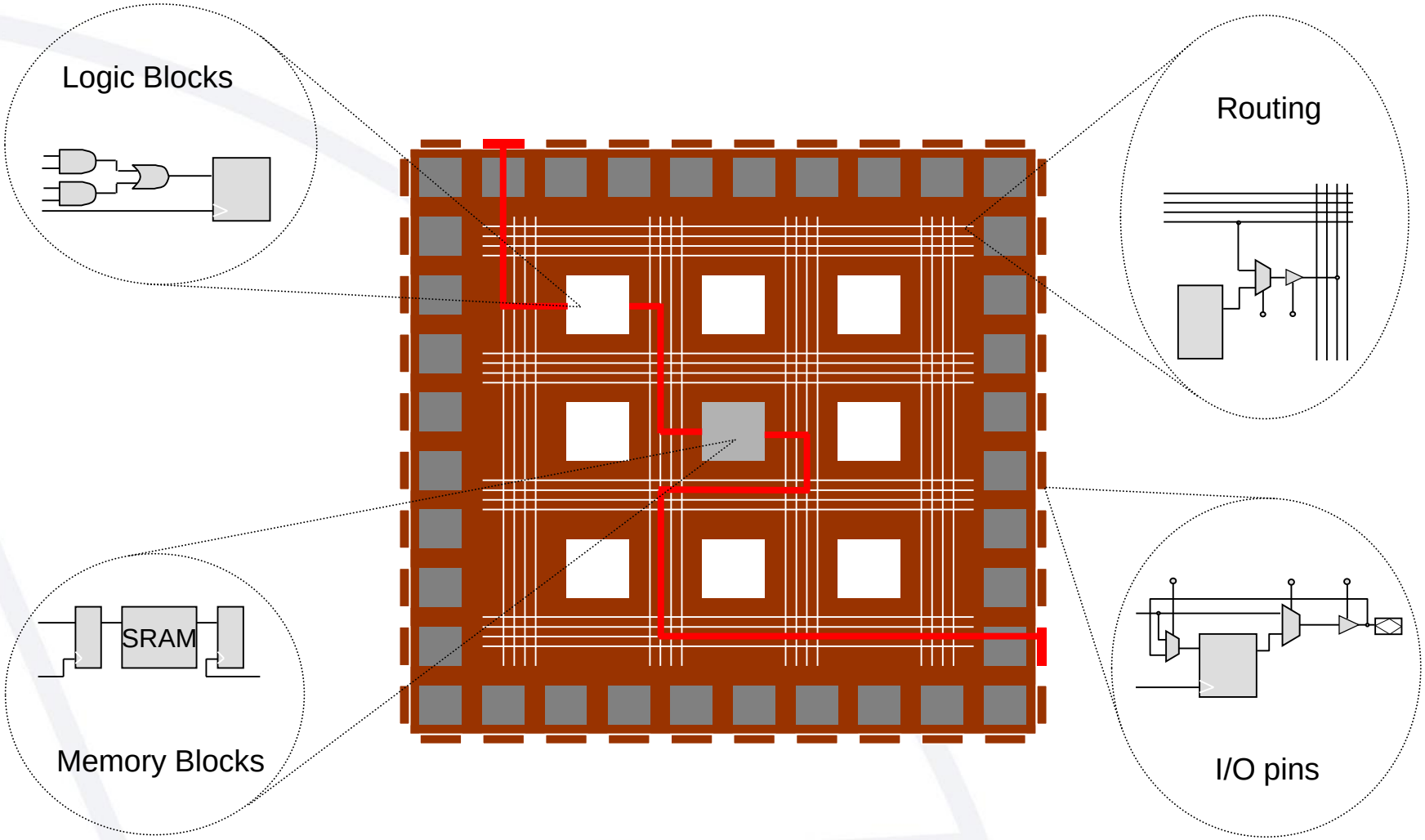






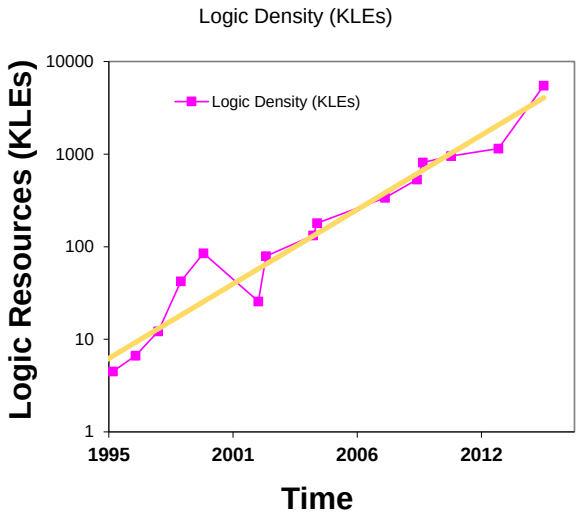


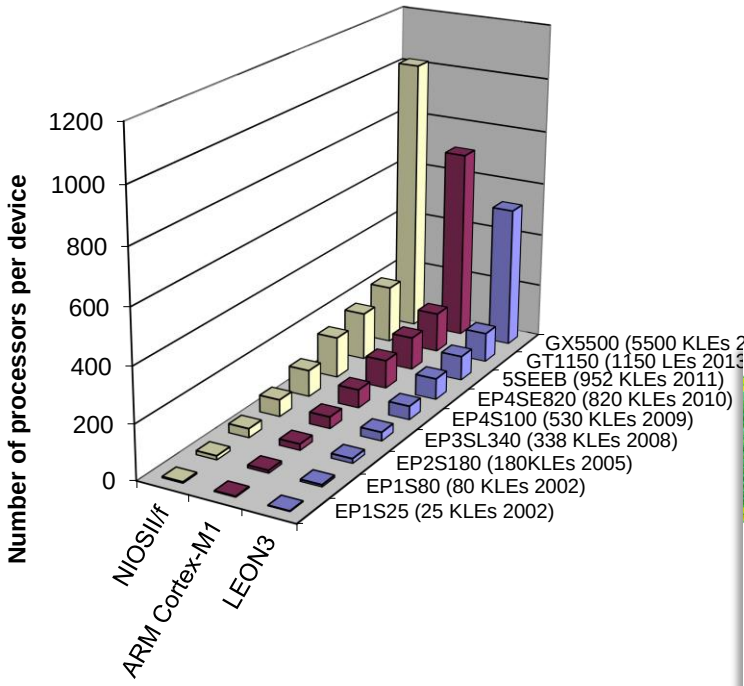
What is an FPGA?

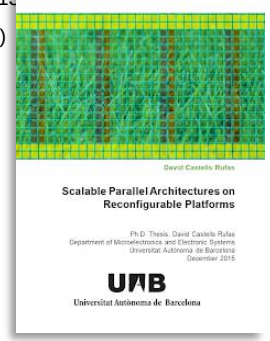


How “Big” is an FPGA?

- A simple RISC processor takes about 6 KLE
- FPGAs are early adopters of last technology nodes (11nm Intel node)
- Hundreds of soft-core processors can already be embedded in current big FPGA devices







Why FPGAs can be Energy Efficient ?

Energy Efficiency Drivers in CMOS era

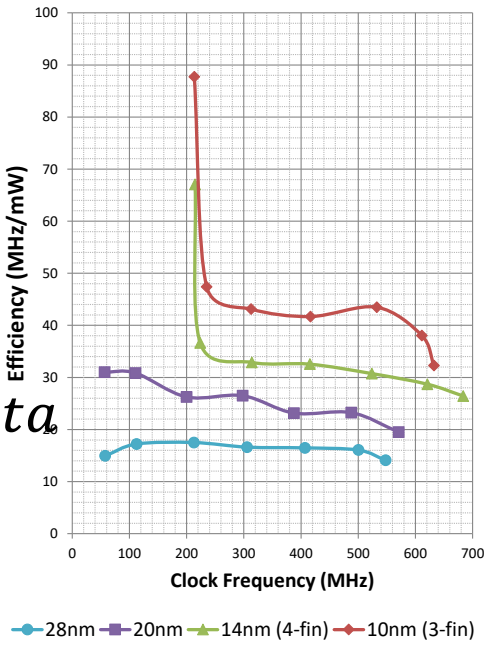
$$G = \frac{\mathrm{OPC} \cdot f_{clk}}{P} = \frac{\mathrm{OPC} \cdot f_{clk}}{P_{dyn} + P_{sta}} = \frac{\mathrm{OPC} \cdot f_{clk}}{N \cdot \alpha \cdot f_{clk} \cdot C \cdot V^2 + N \cdot V \cdot I_{leakage}}$$

$$G_{dyn} = \frac{OPC}{N \cdot \alpha \cdot C \cdot V^2}$$

$$G = \frac{OPC}{N \cdot \left(\alpha \cdot (C \cdot V^2) + \frac{V \cdot I_{leakage}}{f_{clk}} \right)}$$

Why FPGAs can be Energy Efficient ?

$$G = \frac{OPC}{N \cdot \left(\alpha \cdot (C \cdot V^2) + \frac{V \cdot I_{leakage}}{f_{clk}}\right)}$$



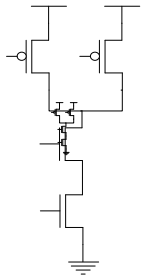
- Before Synthesis
  - $\nabla f_{clk} \Rightarrow \nabla C, \nabla N \Rightarrow \nabla P_{dyn}, \nabla P_{sta}$
- After Synthesis
  - $\nabla f_{clk} \Rightarrow \Delta \int P_{sta}$
- Empirical f_{max}

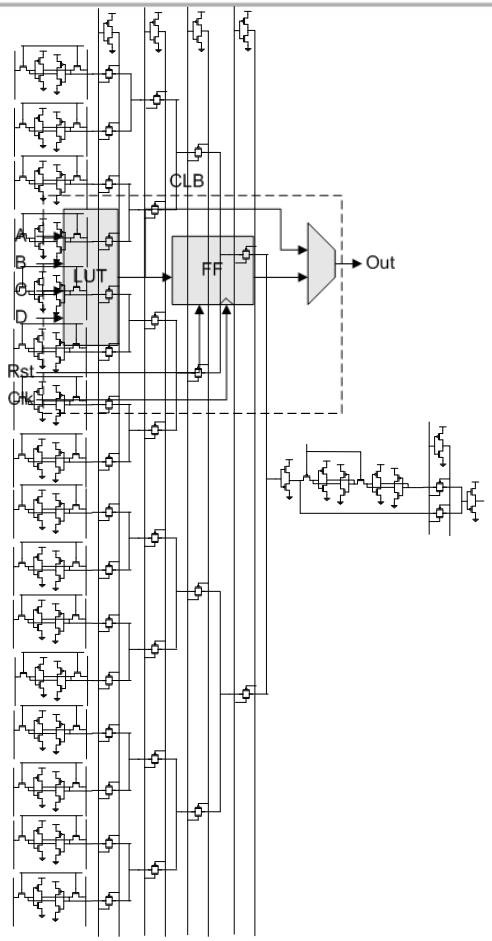
$$K_f = \frac{f_{clk\ IC}}{f_{clk\ FPGA}} \approx 20$$

Why FPGAs can be Energy Efficient ?

$$G = \frac{OPC}{N \cdot \left(\alpha \cdot (C \cdot V^2) + \frac{V \cdot I_{leakage}}{f_{clk}}\right)}$$

- FPGAs use more transistors





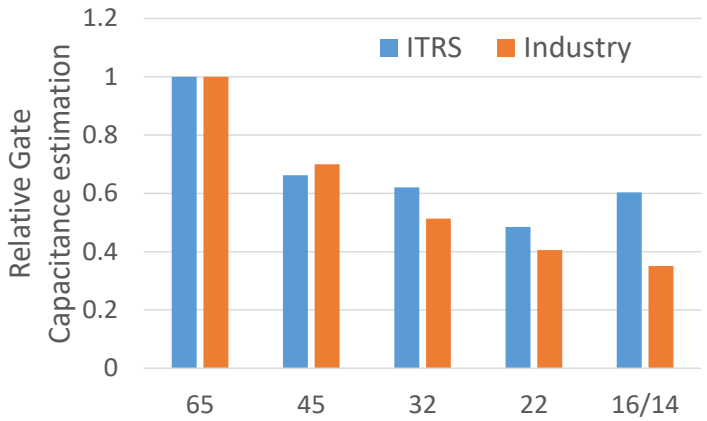
Circuits	Technology	Area Overhead
Small Benchmarks [Kuon07]	90 nm	23-55
Pentium [Lu07]	65 nm	53
OpenSparc, Atom, Nehalem [Wong11]	65 nm	17-27

$$K_N = \frac{N_{FPGA}}{N_{IC}} \approx 40$$

Why FPGAs can be Energy Efficient ?

$$G = \frac{OPC}{N \cdot \left(\alpha \cdot (C \cdot V^2) + \frac{V \cdot I_{leakage}}{f_{clk}}\right)}$$

- C is complex
  - –Transistor tech. & sizing, Fan-out, Wiring
- Isolate Tech. Node contribution
- ITRS predictions are not exactly inline with industry

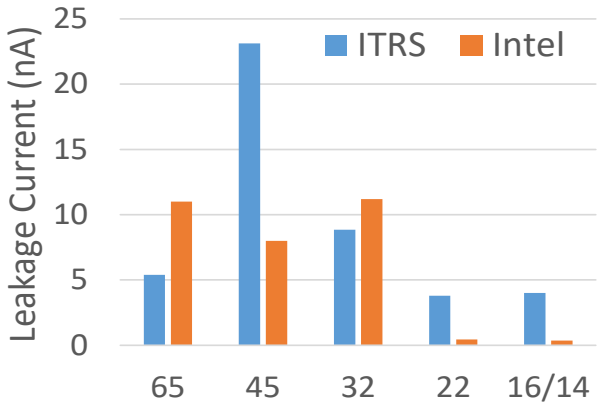


Node Name	65nm	45nm	32nm	22nm	14nm
C_g (aF)	72.9	49.65	41.32	32.44	34.78
Relative C_g	1	0.68	0.56	0.44	0.47

Why FPGAs can be Energy Efficient ?

$$G = \frac{OPC}{N \cdot \left(\alpha \cdot (C \cdot V^2) + \frac{V \cdot I_{leakage}}{f_{clk}} \right)}$$

- Leakage power is a major concern
  - –But due to N
- Fixed by tech. Node (sleep modes)
- Isolate Node Contribution.



Node Name	65nm	45nm	32nm	22nm	14nm
$I_{leakage}\ (nA)$	11	8	11.2	0.45	0.35
Relative $I_{leakage}$	0.98	0.71	1	0.04	0.03

Why FPGAs can be Energy Efficient ?

$$G = \frac{OPC}{N \cdot \left(\alpha \cdot (C \cdot V^2) + \frac{V \cdot I_{leakage}}{f_{clk}}\right)}$$

>1

1

<1

1/5

1/10

1/769

Von-Neumann Machine

Register File

Cache (L0)

Harvard Architecture

Pipelining

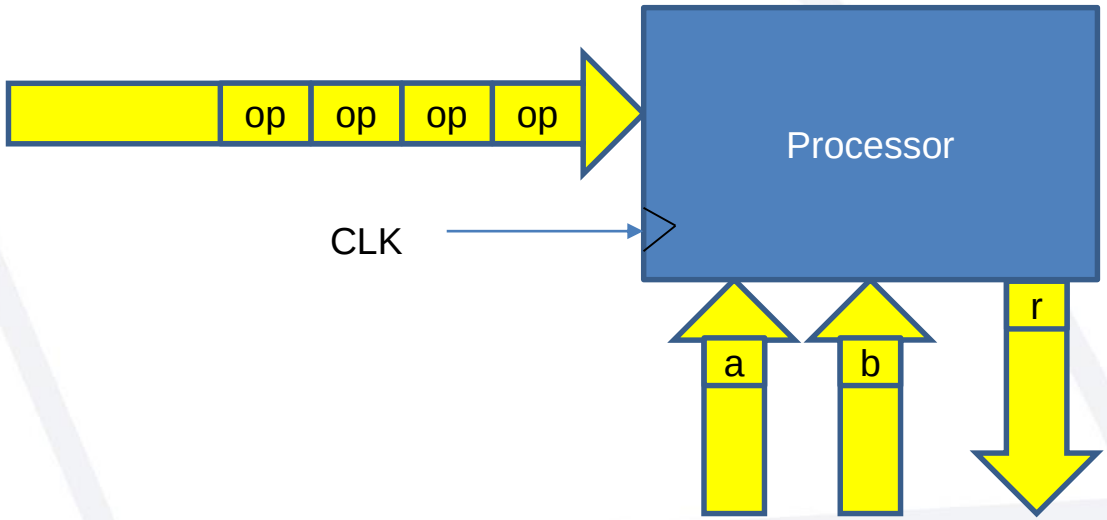
Branch Prediction

Cache Hierarchy

VLIW, Vector, Superscalar, SMT

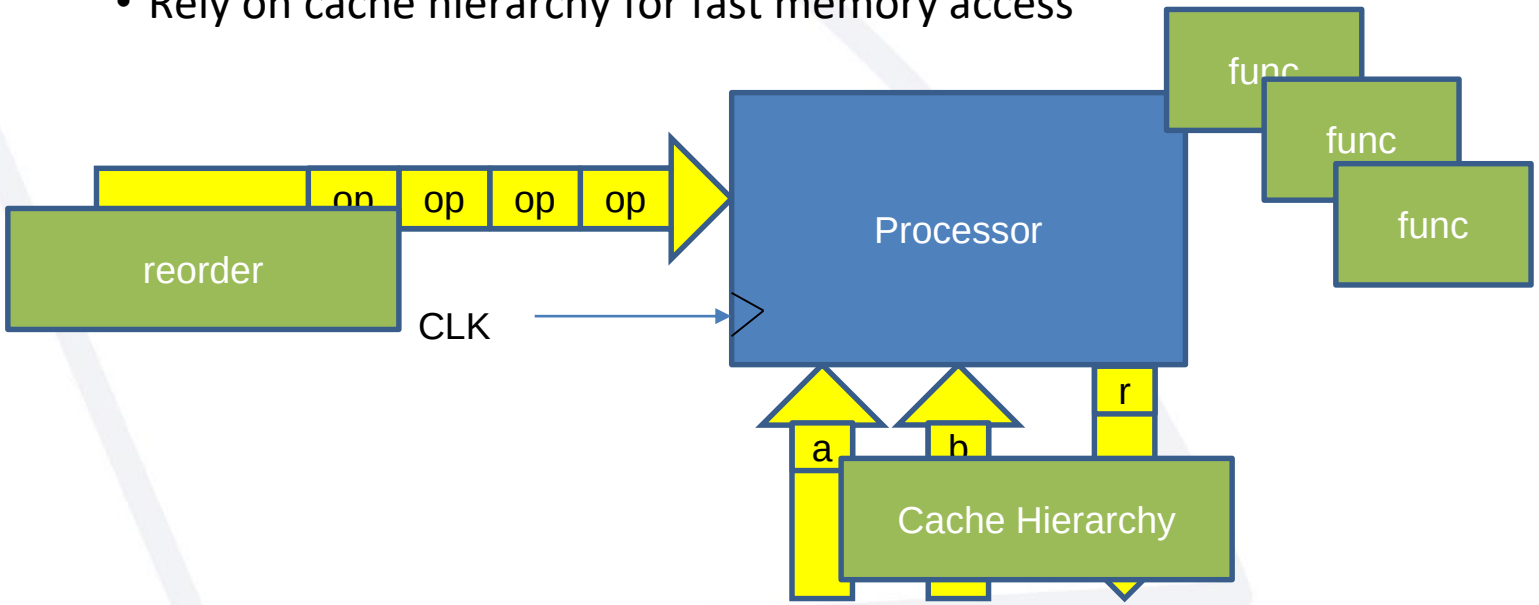
The Basic Fundamental Problems to increase OPC

- Frequency is limited
- Memory Access Latency > several clock cycles
- Data Dependency prevents parallelization



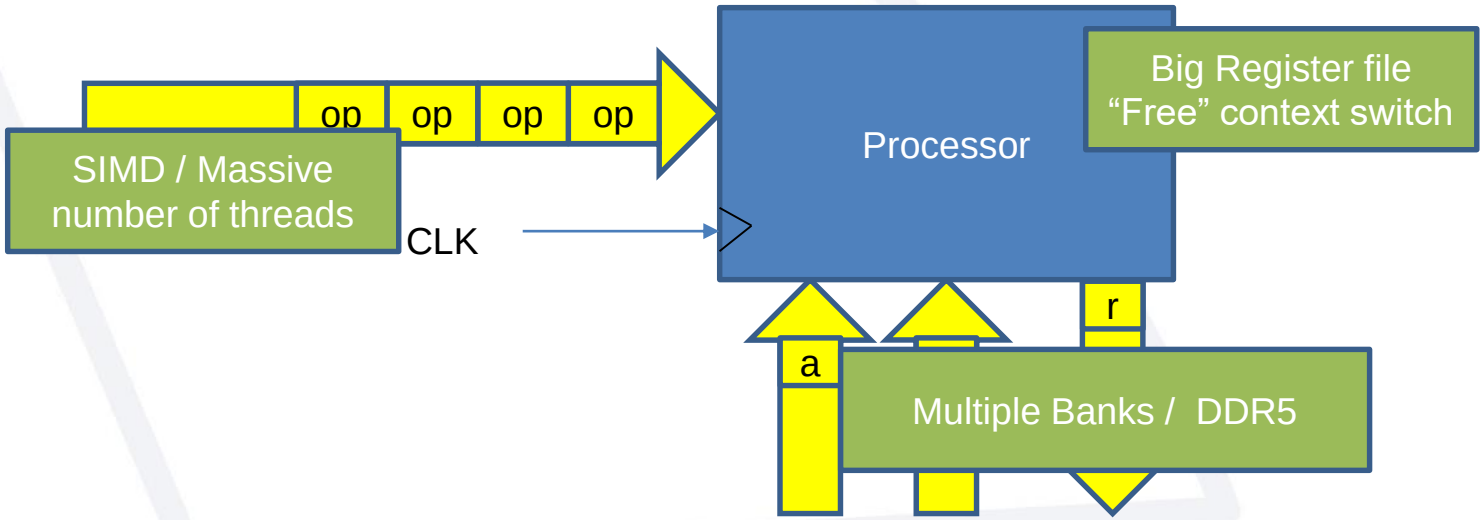
How Superscalar uP get the Job done?

- High frequency (3GHz)
- Have multiple functional units
- Have long instructions queues so that dependency analysis can be done and operations can be scheduled simultaneously
- Rely on cache hierarchy for fast memory access



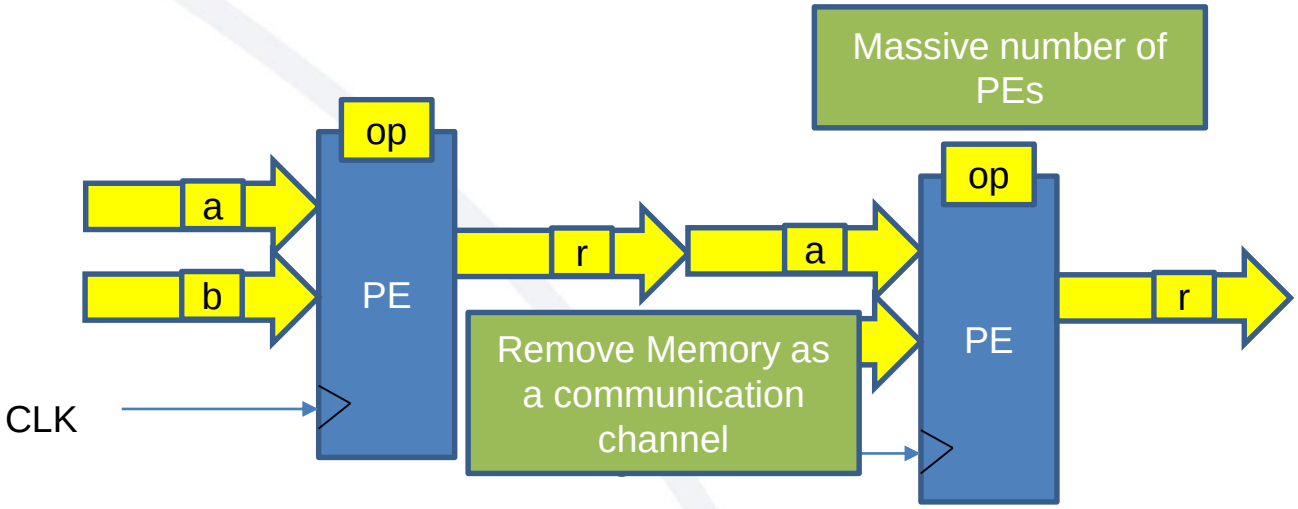
How GPUs get the Job done ?

- High frequency (2GHz)
- Have multiple SIMD processors and support for massive number of threads with fast context switch (1 clk)
- Very High Memory Bandwidth (DDR5), multiple Banks
- Avoid data dependencies by running different threads
  - data accesses from different threads are not dependent



What FPGAs can provide?

- Remove intermediate memory from computation datapaths
- Allow much higher number simultaneous computation units
- Fine grain (bit level) computation



- Problems
  - Lower frequency (due to overheads)
  - Difficult to program (HDL)

Questions?