GPUs, Co-Processors, ASICs, and FPGAs

Sommerakademie in Leysin
AG 2 – Effizientes Rechnen

Maximilian Schäffeler

TU München

August 2016

Outline

1. Challenges to HPC
   - Functioning of a regular CPU
   - Limits of Moore's Law
   - Problems with general purpose processors
   - Need for Heterogeneous Architectures
2. Co-Processors
   - Intel Xeon Phi
   - GPUs
3. ASICs
4. FPGAs
   - Reconfigurable Computing
   - Applications

Functioning of a CPU

add eax, ebx

Functioning of a CPU

add eax, ebx

Instruction Cycle

- Fetch instructions
- Decode instructions
- Fetch operands
- Execute instruction

Functioning of a CPU

add eax, ebx

Instruction Cycle

- Fetch instructions
- Decode instructions
- Fetch operands
- Execute instruction

$\Rightarrow$ Single Instruction Single Data (SISD)

Limits of Moore's Law

Moore's law: number of transistors on a chip doubles every two years

Limits of Moore's Law

Moore's law: number of transistors on a chip doubles every two years

- Performance only increases by 40%
- Approaching physical limits

$\Rightarrow$ Increasing speed/number of cores not sufficient

Limits of Moore's Law

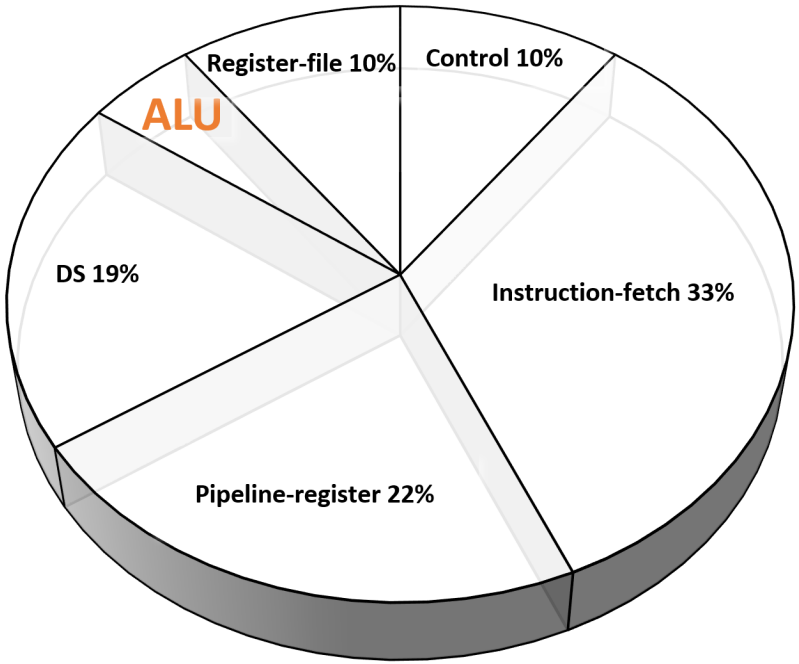
Moore's law: number of transistors on a chip doubles every two years

- Performance only increases by 40%
- Approaching physical limits

$\Rightarrow$ Increasing speed/number of cores not sufficient

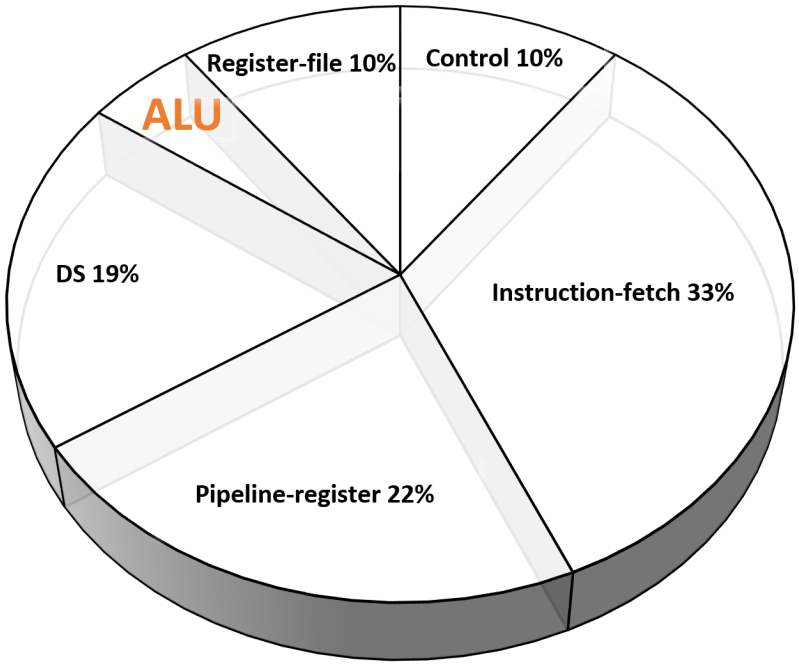
Problems with general purpose processors

- General Purpose $\Rightarrow$ Control Unit Overhead



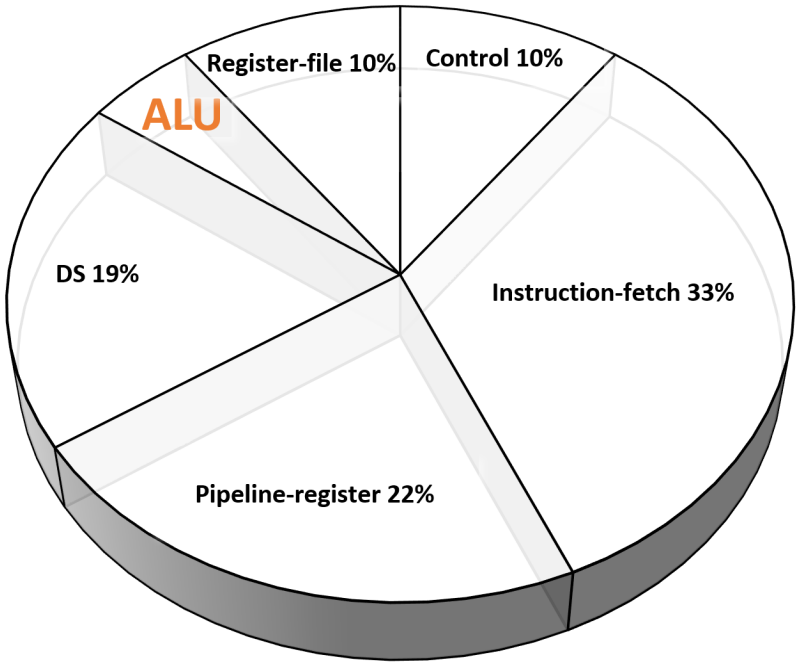
Problems with general purpose processors

- General Purpose $\Rightarrow$ Control Unit Overhead
- High power demand, supply costs are exploding



Problems with general purpose processors

- General Purpose ⇒ Control Unit Overhead
- High power demand, supply costs are exploding
- Low parallelism
- HPC: often a Single Instruction is executed on multiple Data (SIMD)
  ⇒ CPU has to fetch and decode for each operation



Need for Heterogeneous Architectures

Alternatives

Need for Heterogeneous Architectures

Alternatives

- Multi-Core CPUs
- Co-Processors
- Hardware Accelerators
  - ASICs
  - FPGAs

Heterogeneous Architectures are crucial for further increasing performance in HPC

1 Challenges to HPC
- Functioning of a regular CPU
- Limits of Moore's Law
- Problems with general purpose processors
- Need for Heterogeneous Architectures

2 Co-Processors
- Intel Xeon Phi
- GPUs

3 ASICs

4 FPGAs
- Reconfigurable Computing
- Applications

Definition

- Specialized processor to handle related tasks
  - floating point arithmetic
  - graphics processing
  - encryption
  - etc.

Definition

- Specialized processor to handle related tasks
  - floating point arithmetic
  - graphics processing
  - encryption
  - etc.
- Often bound to a family of processors
- Aim to increase performance and lower energy demand

Examples:

- Apple M7
- 80387 (FPU)

Intel Xeon Phi

- Coprocessor for Intel Xeon processors
- Highly parallel vector computations
- Vector capabilities / SIMD $\Rightarrow$ 3 teraFLOPS (floating-point)

Intel Xeon Phi

- Coprocessor for Intel Xeon processors
- Highly parallel vector computations
- Vector capabilities / SIMD $\Rightarrow$ 3 teraFLOPS (floating-point)
- Specifications
  - Up to 72 cores
  - 16GB on-chip memory
  - 215W power consumption

Features of GPUs

- Originally accelerators for 3D-Applications
  $\Rightarrow$ Today: General Purpose computing features

Features of GPUs

- Originally accelerators for 3D-Applications
  $\Rightarrow$ Today: General Purpose computing features

- Communication to host via PCI-Express

Features of GPUs

- Originally accelerators for 3D-Applications
  $\Rightarrow$ Today: General Purpose computing features
- Communication to host via PCI-Express
- Highly parallel
  Grid $\Rightarrow$ Block $\Rightarrow$ Warp $\Rightarrow$ Thread
- Single Instruction Multiple Data
  $\Rightarrow$ Applications with many floating point operations benefit most

Problems with GPUs

- High power consumption
- Difficulties of parallelization
- Relatively low performance gains

1 Challenges to HPC
- Functioning of a regular CPU
- Limits of Moore's Law
- Problems with general purpose processors
- Need for Heterogeneous Architectures

2 Co-Processors
- Intel Xeon Phi
- GPUs

3 ASICs

4 FPGAs
- Reconfigurable Computing
- Applications

ASICs

Application Specific Integrated Circuits

- Hardwired accelerators
  $\rightarrow$ Functionality cannot be changed after fabrication

ASICs

Application Specific Integrated Circuits

- Hardwired accelerators
  $\rightarrow$ Functionality cannot be changed after fabrication

Applications

- Bitcoin mining
- Car: Anti-lock, airbag
- Kitchen & Home Appliances

ASICs: Pros and Cons

Pros

- Excellent performance
- Low power consumption

Example: ASIC for video codec H.264 is 500 faster than a regular CPU

ASICs: Pros and Cons

Pros

- Excellent performance
- Low power consumption

Example: ASIC for video codec H.264 is 500 faster than a regular CPU

Cons

- High development and production cost for low quantities
  High quantities actually make ASICs cheaper than FPGAs
- No versatility, cannot be updated after production

1 Challenges to HPC
- Functioning of a regular CPU
- Limits of Moore's Law
- Problems with general purpose processors
- Need for Heterogeneous Architectures

2 Co-Processors
- Intel Xeon Phi
- GPUs

3 ASICs

4 FPGAs
- Reconfigurable Computing
- Applications

FPGA

Field Programmable Gate Array

Reconfigurable circuits used in HPC

- Configurable Logic Blocks as Gates
- LUT-based, select a boolean function (e.g. or)
  $\Rightarrow$ Saving configuration with SRAM or antifuse
- CLB connected through configurable wiring

FPGA

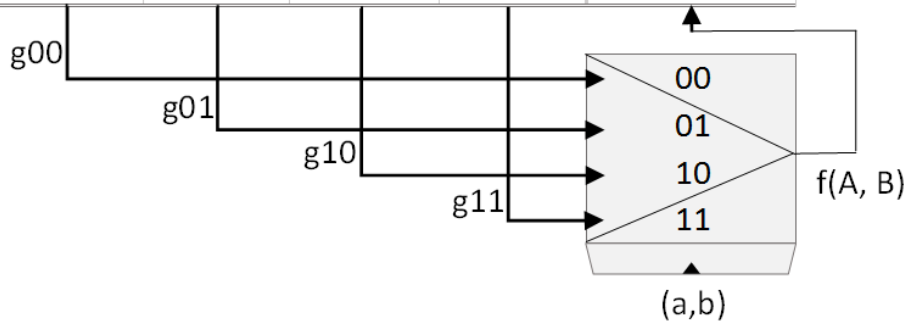
Field Programmable Gate Array

Reconfigurable circuits used in HPC

- Configurable Logic Blocks as Gates
- LUT-based, select a boolean function (e.g. or)
  $\Rightarrow$ Saving configuration with SRAM or antifuse
- CLB connected through configurable wiring
- Combined with host CPU (e.g. integrated ARM-Processors) for general purpose processing and boot-up, configuration and I/O
- Manufacturers: Altera, Xilinx

Configurable Logic Blocks (LUT)

#	g00	g01	g10	g11	f(A,B)
0	0	0	0	0	0
1	0	0	0	1	A and B
2	0	0	1	0	B disables A
3	0	0	1	1	A and B
4	0	1	0	0	A disables B
5	0	1	0	1	B
6	0	1	1	0	A exor B
7	0	1	1	1	A or B
8	1	0	0	0	not (A or B)
9	1	0	0	1	A coin B
10	1	0	1	0	no (B)
11	1	0	1	1	B implies A
12	1	1	0	0	not(A)
13	1	1	0	1	A implies B
14	1	1	1	0	not (A and B)
15	1	1	1	1	1



Reconfigurable Computing

Circuits, that can be configured with programming

- Hardware-Description-Languages (VHDL)
  $\Rightarrow$ configware & flowware
- Reconfiguration possible at runtime

Reconfigurable Computing

Circuits, that can be configured with programming

- Hardware-Description-Languages (VHDL)
  $\Rightarrow$ configware & flowware
- Reconfiguration possible at runtime

$\Rightarrow$ Slower, larger, less power-efficient than ASICs or microprocessors, but:

Superior to microprocessors $\equiv$ Reconfigurable Computing Paradox

Applications

Can be used in small projects because of low cost

- Encryption/Decryption
- Pattern Matching

Large number of memory accesses, often present in scientific problems promise huge improvements

Speedup factors of over 1000 possible
$\Rightarrow$ FPGAs are used in 70% of all embedded devices

Thank you!