Theme: Towards Reconfigurable HPC
Lecture **6**

Platforms III – Programmmable Logic

Manfred Mücke

University of Vienna
Research Lab Computational Technologies and Applications

Inverted CERN School of Computing, 3-5 March 2008

Overview

- **Objectives**
  - Understanding how FPGAs work
  - Making sense of performance figures
  - Future developments
- **Contents**
  - The FPGA Advantage
  - Short History of custom logic devices
  - Introduction to FPGAs
  - FPGA Design Flow
  - FPGA Performance Figures
  - Softcores
  - Outlook

Where comes all that craze from?

The FPGA Advantage

FPGA in a Nutshell

- Create your own hardware
- From glue-logic to powerhouse (DSP, HPC)
- Exploit parallelism rather than clock frequency (~150MHz)
- Prototyping (flexible), embedded (low-power), real-time (fast), networked devices (all), consumer products (all + price)
- Price: $10 - $1000
- Altera + Xilinx: $1.2bn + $1.7bn (2006)

Following Moore (more or less)

Computing Power

- CPU: MIPS/f = const.
  CPUs turn additional transistors e.g. into larger caches or more elaborate branch prediction or additional cores.
- FPGAs: MIPS/f/area = const.
  FPGAs can translate higher device integration directly into additional (parallel) computing resources.

FPGA: More transistors ⇨ More computations/timestep

MIPS .. Million Instructions Per Second
f .. Design Frequency

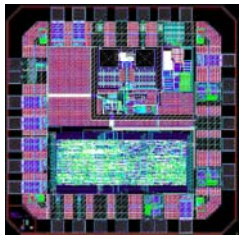
Where do FPGAs come from?

Short History of Custom Logic

ASICs

ASIC = Application-Specific Integrated Circuit

- Full-custom design
- Mixed-signal possible
- Highly efficient
- Matches exactly your needs
- Expensive (NRE)
- Extensive know-how required
- Long design cycles



QPLL ASIC, CERN MIC

Standard Logic Wired-Up

CPLD

- **CPLD** = Complex Programmable Logic Device
  = Logic Device + Configuration Memory
- Available since 1970's (and still evolving)
- Macro cells implement disjunctive normal form
  (A and B) or (A and (not C))
- Registered Outputs
- Different technologies to keep configuration (now Flash)
  +Low cost
  +Non-volatile configuration
  +Predictable timing characteristics
  - Limited functional complexity

FPGAs

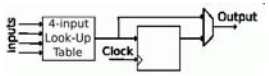
- **FPGA** = Field-Programmable Gate Array
- Invented 1984, evolved from CPLDs
  CPLDs separate logic and register
  FPGAs combine logic and register
- Emphasis on interconnect of small configurable blocks
  + Arbitrary complex functionality
  - Most area for interconnect
  - Variable timing
  - Demanding tool requirements
- ⇨ The "winning" (most versatile) architecture

How does it work?

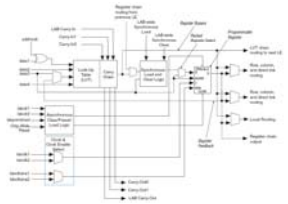
Introduction to FPGAs

Look-Up Tables (LUT)

- Any boolean function of n variables can be implemented by a memory of size 2^n ⇨ LUT = small memory
- LUTs in FPGAs use typically 4 inputs.
  ⇨ 1 LUT needs 16 configuration bits.

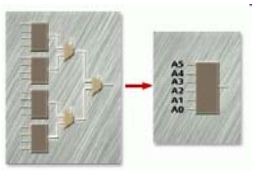


- Add a single-bit register
- Memory is typically SRAM
  (needs to be written at power-up)
- Find a marketing name
  Xilinx: Logic Cells
  Altera: Logic Elements
- Fit as many as possible (> 200.000)

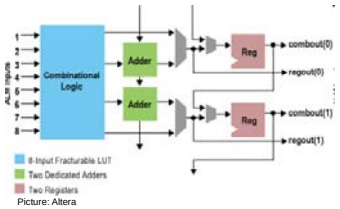


Bigger LUTs

- Balance routing and logic granularity
- Routing dominates ⇨ LUTs can grow



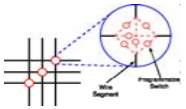
Picture: Xilinx

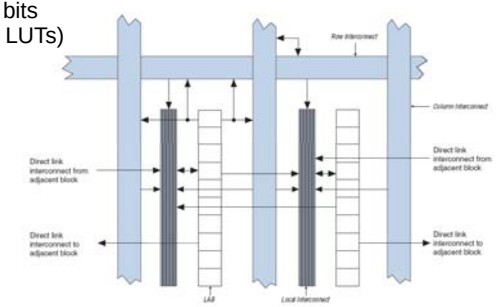


Picture: Altera

Interconnect

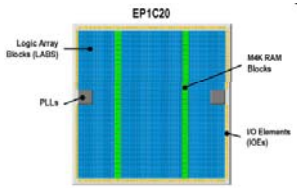
- Guarantee that all (many) LUTs can be interconnected
- Configurable Switches
- Configuration bits in SRAM (like LUTs)
- Most area is interconnect
- Huge delay variations





Basic FPGA

- **LUTs + Interconnect**



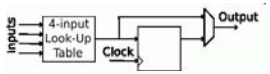
Altera Cyclone			Xilinx Spartan-IIE		
	EP1C6	EP1C20		XC2S 300E	XC2S 400E
LE	5.980	20.060	LC	6.912	10.800
Pins	185	301	Pins	329	410

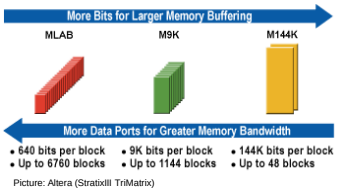
Memories

- LUTs come with 1bit of memory only (the register).
- If you need some memory, you waste a lot of LUTs ☹

⇨ Integrate some dedicated memory blocks

Each memory is accessible in parallel ⇨ Huge memory bandwidth possible





Picture: Altera (StratixIII TriMatrix)

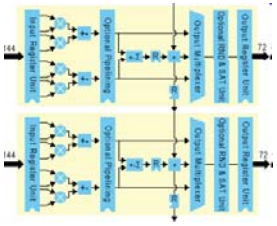
Hard IP Blocks

- Some functionality maps bad into LUTs (Multiplication)
- Some functionality is required by most designs

⇨ Provide fixed blocks for frequently requested functions

⇨ Multiplier
⇨ Adder
⇨ MAC
⇨ ALUs

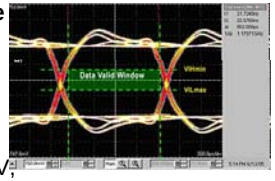
} DSP Block

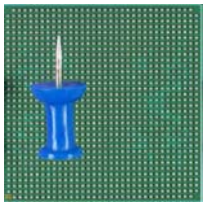


+ runs much faster (~500MHz)
+ costs no routing resources

I/O

- **Pins connect the FPGA to the outside world**
- **Originally low-speed TTL**
  Today: Everything ☺
  Especially:
  LVDS (SPI-4.2, SFI-4, SGMII, Utopia IV, 10 GbE XSBI, RapidIO, SerialLite.),
  RAM (DDR SDRAM, DDR2 SDRAM, DDR3, QDRII, QDRII+, RLDRAMII)
- **High pin-count (> 1000)**

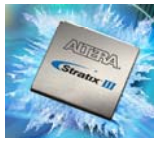


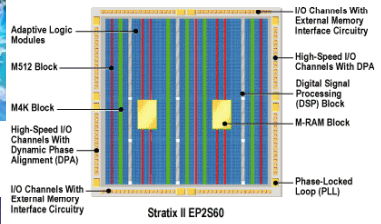


Fraunhofer IZFP (1152pin BGA)

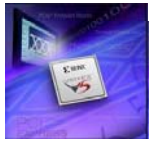
Contemporary FPGAs

- **Altera Stratix III**
- **Cyclone III**

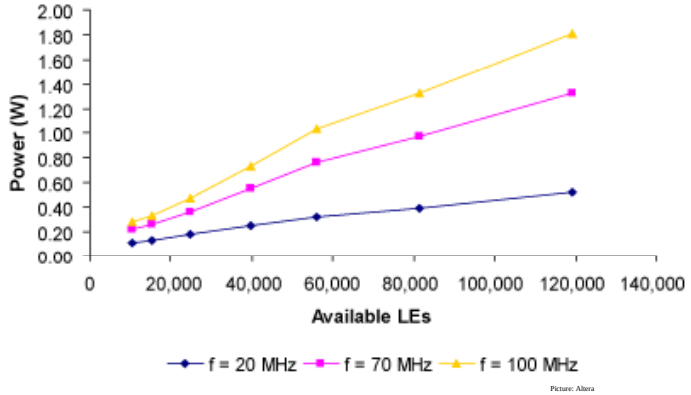




- **Xilinx Virtex 5**
- **Spartan 3**



Power Consumption (Cyclone III)

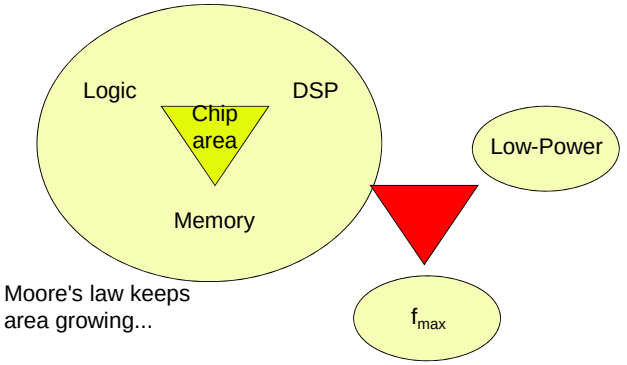


Picture: Altera

Power Consumption

- Startup power
- Total Power =
  core static power (const.) +
  core dynamic power (~f) +
  I/O power
- **Static power:** Device junction temperature, voltage and silicon process variation. **Device-dependent**
  ⇧ Shrinking feature size
- **Dynamic power:** FPGA resource utilization, routing utilization and circuit switching activity. **Design-dependent**
  ⇩shrinking feature size
  ⇧ Clock rate, total resources

FPGA Device Range



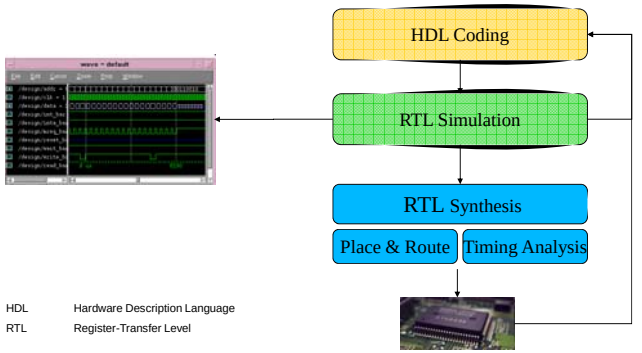
Moore's law keeps area growing...

...but makes timing and power closure harder and harder.

How to tell them what to do?

FPGA Design Flow

FPGA Design Flow



HDL Hardware Description Language
RTL Register-Transfer Level

VHDL (VHSIC HDL)

- Invented for description of ASICs (US DoD, 1980s)
- Based on Ada programming language
- Powerful, but complex modeling language.
- Subset is synthesizeable (IEEE Std 1076.6-2004) (mostly RTL)
- Powerful tools available for all FPGAs.
  ⇨ Simulator
  ⇨ Synthesizer
- Writing/Debugging/Optimizing VHDL is a tedious task!
  ⇨ For high-speed designs, there is no better way (yet).

Do you know, what you get?

FPGA Performance Figures

Performance Figures

- FPGA marketing likes to impress with big numbers!

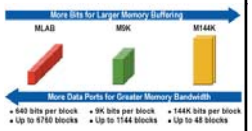
The GMACs

- Performance figure from the DSP world (1 MAC/cycle)
  1MAC: a = a + b*c
- DSP @ 500MHz ⇨ 0.5GMAC/s
  You can achieve this with a loop
- FPGA (64 MAC blocks @ 500 MHz) ⇨ 32GMAC/s WOW!
  1) Watch the bitwidth!
  2) Can your design run at 500MHz, too?
  3) Can you deliver data at 500MHz*64*xbits?
  4) Can your design cover all FPGA (MAC) resources?

Performance Figures

Total Memory Bits

- FPGAs have distributed memories of different sizes
  Every LUT can serve as mini-memory
  1. Know your desired size
  2. Then ask: How many blocks of size x.



Same for **accumulated memory bandwidth**

Logic resources: Compare with similar designs

- **FPGAs are not filled > 80%**
- **FPGAs run typically at ~150MHz**

Designing your own CPU

Softcores

SoftCores

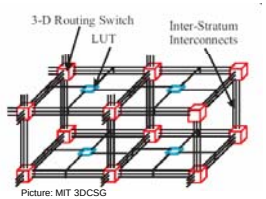
- **Softcores provide CPU-functionality implemented on FPGA**
- Optimize your CPU (HW stacks, MAC, ..) !
- Don't worry about CPU long-term availability.
- Cheap and slow (5%, 100-200MHz)
- Slow, but: Embedded Linux on Softcore ☺!
- Needs tight integration with software toolchain (custom compiler)

Where will FPGAs go to?

FPGA Outlook

3D Interconnect

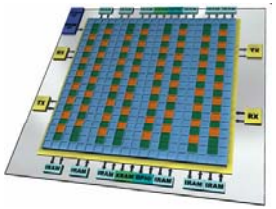
- 2D Interconnect consumes most of an FPGAs **area**.
- The longer, the harder to route.
  Routing (and power) is the limiting factor.
- 3D provides "shortcuts",
  enables higher logic and routing density.
- **Manufacturable?**



Picture: MIT 3DCSG

Bigger basic blocks

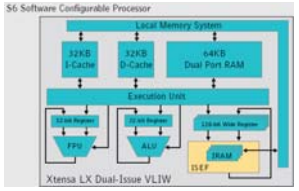
- Idea: Provide **coarser functional blocks**.
- Replace a set of LUT + interconnect with dedicated blocks (5/16bit)
- Configurable Blocks:
  - ALU
  - 16x16 MAC
  - Register
  - FSM
- Configurable Interconnect



+ runs at **1GHz**
+ **reduced routing** overhead
⇨ (very reduced) RISC Manycore

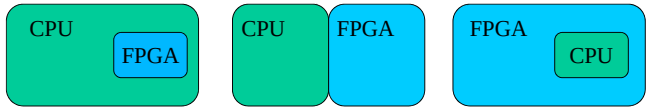
CPU + FPGA

- **CPU + FPGA or FPGA + CPU?**




Altera Excalibur EXPA10 die


Picture: Xilinx



Further Reading

- Maxfield, C., **The Design Warrior's Guide to FPGAs.** April 2004. Newnes.
- Ashenden, P. J., **The Designer's Guide to VHDL.** 2001. Morgan Kaufmann Publishers
- Xilinx: www.xilinx.com
- Altera: www.altera.com
- Dong, C., Chen, D., Haruehanroengra, S., Wang, W., **3-D nFPGA**: **A reconfigurable architecture for 3-D CMOS/nanomaterial hybrid digital circuits**. 2007. IEEE Transactions on Circuits and Systems I: Regular Papers, 54 (11), 2489-2501. http://dx.doi.org/10.1109/TCSI.2007.907844

Q & A