FPGA 101
Field programmable gate arrays in action
About me

• Karsten Becker
• Head of electronics @Part-Time Scientists
• PhD candidate @TUHH
• FPGA Architecture
What is an FPGA

- Programmable Logic
Programmable logic (LUT)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>X</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>Y</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Programmable logic (LUT)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>C</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Programmable logic (LUT)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 25: Simplified Virtex-6 FPGA Slice
FPGA vs CPU

Time

CPU

FPGA

Area/Frequency
Fabric

- Composed of
  - Lookup tables
  - Flip-flops
  - Multiplexer
  - Routing resources
  - Clock Management
  - Carry Logic
  - Shift-register

- Hard blocks
  - DSP Units
  - Multiplier
  - Block RAM
  - High-speed IO
  - CPUs
  - Memory controller
  - ADC/DACs
Cool FPGA Projects

• Borgventilator
• Pixel transformation to polar coordinates
• Transmission as 8b10 coding
• Gamma correction
• Precise timing
• Go to: Das Labor (Hackerspace) somewhere here
Cool FPGA Projects

• Borgventilator
• Pixel transformation to polar coordinates
• Transmission as 8b10 coding
• Gamma correction
• Precise timing

Go to: Das Labor (Hackerspace) somewhere here
Cool FPGA Projects

• HDMI Overlay
  • Detect HDMI Sync signal
  • Encrypt Overlay
  • Select overlay stream vs content stream
  • Does not decrypt content stream!
• See bunnie @28C3
Programming

- High Level tools
  - Code generation
  - Hardware Description Language
    - Synthesis/Mapping
      - Netlist
        - Place&Route
          - Configuration file

- Matlab Simulink, C to HDL generators, OpenCL to HDL generators
- VHDL, Verilog, MyHDL, PSHDL
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity MyFirstModule is
  port (
    clk : in std_logic;
    rst : in std_logic
  );
end;

architecture pshdlGenerated of MyFirstModule is
  signal counter : unsigned(11 downto 0);
begin
  process(clk)
  begin
    if RISING_EDGE(clk) then
      if rst = '1' then
        counter <= (others => '0');
      else
        counter <= (counter + 1);
      end if;
    end if;
  end process;
end;
Pipelining

Calculate \((A+B)(C+D)+10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B)(C+D)+10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B) \times (C+D) + 10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B) \cdot (C+D) + 10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B) \times (C+D) + 10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B)(C+D)+10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B) \times (C+D) + 10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B)\times(C+D)+10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B)(C+D)+10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B)\times(C+D)+10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B) \times (C+D) + 10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B)(C+D)+10\)

Every Clock the data proceeds one step
Pipelining

Calculate \((A+B)*(C+D)+10\)

Every Clock the data proceeds one step
(High-speed) IO

- Tons of GPIO!
- Toggle frequency as high as fabric frequency
- Speeds of up to 400-800 MBit/s with shift registers
- Some dedicated very high speed transceiver
- 1->25 GB/s
- Very detailed control of characteristics
CPUs

- CPUs are good at
  - things that don't repeat very often
  - Large complex pieces of SW like operating system, network stack or dynamic memory allocation
  - quick turnaround times during development
Combining CPU and FPGA

- Connect via bus/memory interface on same PCB
- Soft CPU realized in fabric
- Hard CPU realized in silicon on FPGA die
Accelerating parts of application

- Start with a pure SW implementation
- Measure performance / identify bottle-necks
- Translate expensive part to HDL
Discrete wavelet transformation

- Used by JPEG2000
- Applied to color channels of pictures
- 2D wavelet = 1D wavelet horizontal + 1D wavelet vertical
Horizontal DWT
Horizontal DWT
Horizontal DWT

Cache Hit
Horizontal DWT
Horizontal DWT
Horizontal DWT
Vertical DWT
Vertical DWT
Vertical DWT
Vertical DWT

Cache Miss
Vertical DWT
Vertical DWT
Horizontal DWT

Vertical DWT

FPGA
DWT Summary

- 1D Horizontal
  - CPU perfect cache hits due to prefetch -> very fast
- 1D Vertical
  - CPU near always cache miss -> significant slow down
- FPGA
  - Linear read -> 2 lines latency -> linear write @ full memory speed
## Tech guide

<table>
<thead>
<tr>
<th></th>
<th>x86</th>
<th>Gpu</th>
<th>Dsp</th>
<th>Fpga</th>
<th>µC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embed</td>
<td>Maybe</td>
<td>Hard</td>
<td>Easy</td>
<td>Easy</td>
<td>Easy</td>
</tr>
<tr>
<td>Low power</td>
<td>Unusual</td>
<td>Nope</td>
<td>Sometimes</td>
<td>Sometimes</td>
<td>Yes</td>
</tr>
<tr>
<td>Float op</td>
<td>Good</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Possible</td>
<td>Nope</td>
</tr>
<tr>
<td>Int op</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Mediocre</td>
</tr>
<tr>
<td>Control flow</td>
<td>Excellent</td>
<td>Challenging</td>
<td>Fair</td>
<td>Challenging</td>
<td>Excellent</td>
</tr>
<tr>
<td>IO</td>
<td>Mediocre</td>
<td>Nope</td>
<td>Ok</td>
<td>Ginormous</td>
<td>Ok</td>
</tr>
<tr>
<td>Pipelining</td>
<td>Nope</td>
<td>Nope</td>
<td>Nope</td>
<td>Yes</td>
<td>Nope</td>
</tr>
<tr>
<td>Programmability</td>
<td>Easy</td>
<td>Medium</td>
<td>Medium</td>
<td>Challenging</td>
<td>Easy</td>
</tr>
<tr>
<td>Timing control</td>
<td>Medium</td>
<td>What?</td>
<td>Fair</td>
<td>Excellent</td>
<td>Fair</td>
</tr>
</tbody>
</table>
Vendor Overview

• Xilinx / Altera
  • Kind of the same
  • High Performance
  • High Bandwidth
  • Biggest 2 on the market
  • Decent tooling

• Actel (microsemi)
  • Low Power
  • High reliability

• Lattice
  • Low Power
  • Cheap
Vendor Overview

- Xilinx / Altera
  - Kind of the same
  - High Performance
  - High Bandwidth
  - Biggest 2 on the market
  - Decent tooling
- Actel (microsemi)
  - Low Power
  - High reliability
- Lattice
  - Low Power
  - Cheap

Most expensive Xilinx: 32.000 EUR
Most expensive Altera: 19.755 EUR (3 pieces min order)
## Hacker friendly Boards

<table>
<thead>
<tr>
<th></th>
<th>Actel</th>
<th>Altera</th>
<th>Xilinx</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cheap</strong></td>
<td>PSHDL Board ~3x €</td>
<td>DE0-Nano 70€</td>
<td>Papilio One 47€ (Spartan 3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BEMICRO CV 40€</td>
<td>Mojo V3 54€ (Spartan 6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XuLA2-LX25 103€</td>
</tr>
<tr>
<td><strong>Powerful</strong></td>
<td>Igloo 2 boards</td>
<td>Cyclone 5 Arria</td>
<td>Artix/Kintex/Virtex</td>
</tr>
<tr>
<td><strong>SoC</strong></td>
<td>SmartFusion2 Starter Kit 295€</td>
<td>EBV SoCrates 359€</td>
<td>MicroZedBoard ~200€</td>
</tr>
<tr>
<td></td>
<td>SmartFusion Starter Kit 100€</td>
<td></td>
<td>Parallela 85€</td>
</tr>
<tr>
<td><strong>CPU +FPGA</strong></td>
<td>Datenkrake ~95€</td>
<td></td>
<td>Logi (Kickstarter)</td>
</tr>
</tbody>
</table>
Thanks, questions?

For infos about the PSHDL board, visit: boards.pshdl.org
Join the workshop on Sunday 9pm Hall F