VLSI SOC Design ET4351

Rene van Leuken

EWI/MECE/CAS

http://ens.ewi.tudelft.nl/Education/courses/et4351/index.php

http://ens.ewi.tudelft.nl/Education/msc_projects.php
Basic architecture

processor

Instruction program

bus

DSP/Compute block

peripheral
Soc: MBLite+ block schema
Wishbone timing
Assignments i

• All tasks have the goal to design and simulate some hardware function, part of a SoC, and possibly to test functionality on the Spartan 3 board.
• Language: VHDL or SystemC (C++)
• Exercises are grouped around the MBLite+ SoC and the Avnet Spartan board.
• Three themes: ASIC, image and communication.
• Students grouped evenly for the themes!
Assignments ii

- All tasks /themes are the ‘same’
- Start with an algorithm, e.g. C Code or Matlab example
- Design an hardware architecture; in VHDL or SystemC
  - Your datapath
  - Wishbone interface
  - Power saving method?
- All modules will have a Wishbone bus interface
- Verify/simulate your design; Use the bus emulator package, download from website;
- Synthesize your design
- In case IO theme program FPGA and check communication
- Write report.
Theme Image

- Gray scale image operations
- C code template code from “Practical algorithms for images analyses”, Michael Seul.
- Image binarization.
- Integer or fixed-point operations.
<table>
<thead>
<tr>
<th>Modell</th>
<th>Widerstand</th>
<th>Legende</th>
<th>Widerstand</th>
<th>Legende</th>
</tr>
</thead>
<tbody>
<tr>
<td>962-20 500K</td>
<td>0.43</td>
<td>liegend</td>
<td>76-40 50K</td>
<td>0.53</td>
</tr>
<tr>
<td>962-20 1M</td>
<td>0.43</td>
<td>liegend</td>
<td>76-40 100K</td>
<td>0.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stehend</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75H 1,0K</td>
<td>0.61</td>
<td></td>
<td>75T 1,0K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 2,0K</td>
<td>0.61</td>
<td></td>
<td>75T 2,0K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 5,0K</td>
<td>0.61</td>
<td></td>
<td>75T 5,0K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 10K</td>
<td>0.61</td>
<td></td>
<td>75T 10K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 20K</td>
<td>0.61</td>
<td></td>
<td>75T 20K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 50K</td>
<td>0.61</td>
<td></td>
<td>75T 50K</td>
<td>0.72</td>
</tr>
</tbody>
</table>

**Miniatur-Regler Ø 6mm**

<table>
<thead>
<tr>
<th>Modell</th>
<th>Widerstand</th>
<th>Legende</th>
<th>Widerstand</th>
<th>Legende</th>
</tr>
</thead>
<tbody>
<tr>
<td>962-20 500K</td>
<td>0.43</td>
<td>liegend</td>
<td>76-40 50K</td>
<td>0.53</td>
</tr>
<tr>
<td>962-20 1M</td>
<td>0.43</td>
<td>liegend</td>
<td>76-40 100K</td>
<td>0.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stehend</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75H 1,0K</td>
<td>0.61</td>
<td></td>
<td>75T 1,0K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 2,0K</td>
<td>0.61</td>
<td></td>
<td>75T 2,0K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 5,0K</td>
<td>0.61</td>
<td></td>
<td>75T 5,0K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 10K</td>
<td>0.61</td>
<td></td>
<td>75T 10K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 20K</td>
<td>0.61</td>
<td></td>
<td>75T 20K</td>
<td>0.72</td>
</tr>
<tr>
<td>75H 50K</td>
<td>0.61</td>
<td></td>
<td>75T 50K</td>
<td>0.72</td>
</tr>
</tbody>
</table>
Theme Image (2)

- Analyse algorithm
  - Check datatypes => FIXED point (Matlab fi function)
  - Check interface (with Wishbone bus)
  - Make concurrent adjustments
    - VHDL: traditional (FSM) / modern (Loops)
    - VHDL: old RTL style / 2 process method
    - SC: idem
      » Wait statements!!!
Theme Communication

- Communication interface processor.
- Ethernet: DM9000E OR Camera interface
- High level communication chips.
- VHDL and embedded C - and Host processor C code.
FPGA board: Spartan & Virtex
DM9000E board
Theme ASIC

ASIC (semi custom) realization of a chip.
Existing VHDL code of a design: MBLite+
Use Synopsys Design compiler and Cadence SoC Encounter.
The goal is to 'map' the filter design to a semi-custom cell library (Faraday SP90) and by using SOCE place and route the design.

Design steps include: simulation, synsthesis (DC), IO port design, power planning, clocktree generation, P&P and GDS2 (SOCE) generation.

The result is a GDS2 file, which in principle could be fabricated at a silicion foundry.
View of the routed, verified and finalized ASIC.
The power plan of the ASIC. Rings, stripes, block halo and the selected floor planning are visible.
MBLite+ ASIC DESIGN

- Introduction MBLite+
- Architecture
- Memories, etc
- Die size
- Pads and IO bufs
- Goals
- Practical
MBLITE+

- Xilinx Microblaze compatible
- MBLite+ designed
- All Xilinx Microblaze software should run
- Check Xilinx website
Architecture

• MBLite+ core, Embedded memory blocks (RAM/ROM), JTAG, Uart, touch screen controller
Memories

- UMC90 Faraday memories
- Single, Register and dual port

- Info file: /opt/eds/DesignLab/tech/Faraday/L90_SP/Memory
Pads and IO bufs

- IO bufs inferred by VHDL code (techmap)
- PADs instantiated and placed (script: genIO)
- Die size 4x2 150 pads, or 184 pads (staggered)
Die size

- Die size 4 x 2 mm == 1875 x 3750 u
Goals

• die 2x2: MBLite+, max embedded memory, all components;

• die 4x2: MBLite, max embedded memory, all components; optional: ROM

• idem using staggered (184) pads
Practical

- All info on: /opt/eds/DesignLab
- /opt/eds/DesignLab/IP/caslib/..
- /opt/eds/DesignLab/designs/tumble_4wTS
- /opt/eds/DesignLab/tech/Faraday/L90_SP/Memory

- User Guide ET4351; New book Brunvand
How to start

• Run example User guide (ADDSUB and FILTER)
• assemble source code, create dir structure
• Run Modelsim ; get simulation to work (start with behavior memory models)
• Determine Architecture; Memory blocks, IO blocks, no PADS
• create, simulate, debug
• DC Synopsys, simulate netlist, debug
• SOC, IO pads, P&R, simulate netlist, debug
• Write report!