CS/EE 3710

Computer Design Lab
Fall 2010

CS/EE 3710

- Computer Design Lab
  - T Th 3:40pm-5:00pm
    Lectures in WEB 110, Labs in MEB 3133 (DSL)
- Instructor: Erik Brunvand
  - MEB 3142
  - Office Hours: After class, when my door is open, or by appointment.
- TA: Michael Kingston
  - Office hours to be determined

CS/EE 3710

- Web Page - all sorts of information!
- http://www.eng.utah.edu/~cs3710
- Contact:
  - 3710@list.eng.utah.edu
    - Goes to everyone in the class
  - teach-3710@list.eng.utah.edu
    - Goes to instructor and Ta
- No textbook – I'll hand out stuff.
  - There's lots of good stuff linked to the web page

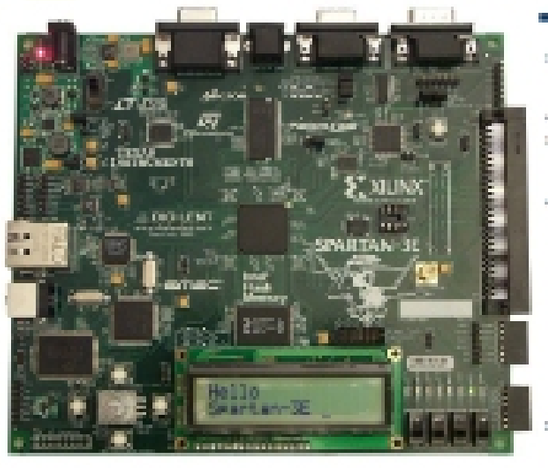
Prerequsites

- Digital Logic
  - CS/EE 3700 or equivalent
- Computer Architecture
  - CS/EE 3810 or equivalent
- First assignment is a review of these subjects!
  - It's on the web page now!
  - It's due on Thursday, September 2 at 5:00pm
    (hand in in class)

Class Goal

- Use skills from both 3700 and 3810 to build a moderately sized project
  - Specifically, a computer processor!
  - Based on a commercial RISC core
- Team projects – groups of 3 or 4
  - Each group will customize their processor for a particular application
    - You choose the application!
    - You choose the customizations!

Hardware Infrastructure

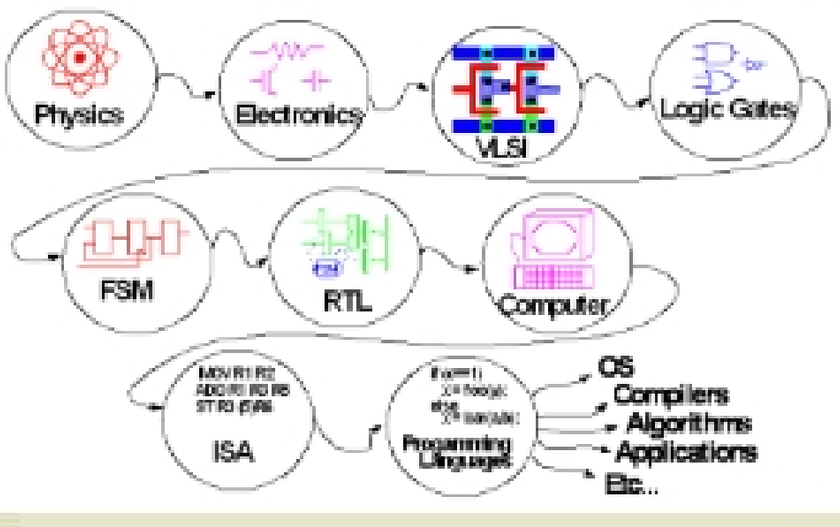


- Spartan-3E "starter" Board from Xilinx

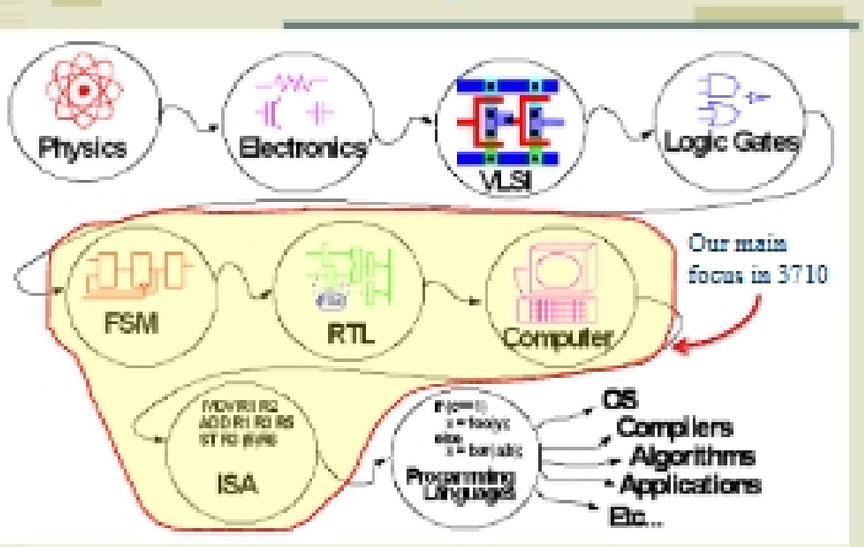
CAD Software

- Xilinx ISE WebPACK 12.2
  - Verilog system definition
  - Schematic capture
  - Verilog/Schematic simulation
  - Synthesis to the Spartan-3E
  - Mapping to the Spartan-3E
- This is installed on the DSL machines, in the CADE PC lab, and is free to install on your own machine
  - It's a BIG download though…

The Big Picture



The Big Picture



The Big Picture

- I'll hand out a Baseline ISA (it's on the web site)
  - Every group must implement these instructions
- There will be labs that require you to design and demonstrate steps along the way
- Each group will customize their processor
  - New instructions
  - New I/O
  - Other features
- End up demonstrating code running on your processor!

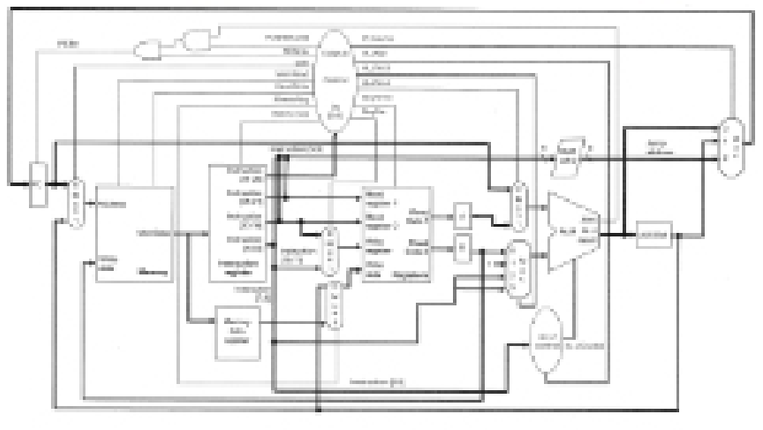
The Big Picture

- Design with a mix of schematics and Verilog
  - Design the datapath
    - ALU, register file, shifter, misc. registers, etc.
  - Design the control FSM
    - Remember Verilog state machine design from 3700?
  - Design the I/O system
    - Memory mapped I/O
    - VGA, PS/2, UART, LCD, etc.
- Use ISE for simulation/synthesis
- Processor runs on the Spartan-3E board

Verilog

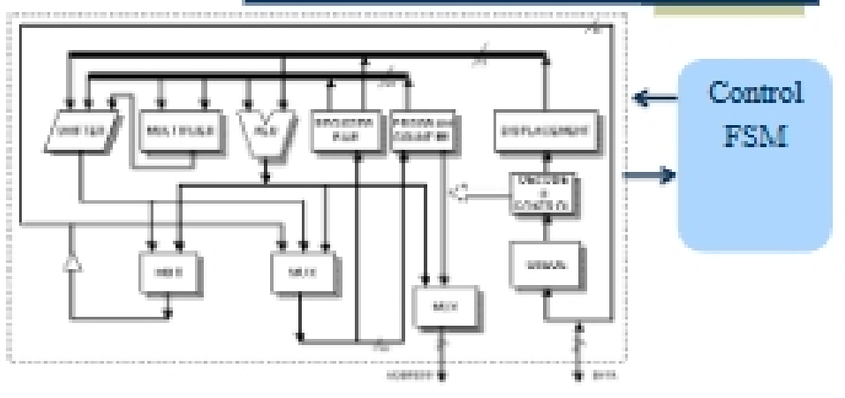
- Plan on good Verilog coding style this semester!
  - Verilog is NOT a programming language!
  - Verilog is a Hardware Description Language
    - A huge number of Verilog errors are related to confusion between combinational and sequential descriptions
    - Think of the HW first, before coding
  - What is "good" Verilog?
    - I like excessive comments in the code
    - I like clear distinctions between seq. and comb. code
    - I like hierarchy
    - I like using a coding style that makes synthesis easy
    - I like using a purely synchronous clocking style in this class

Remember This?

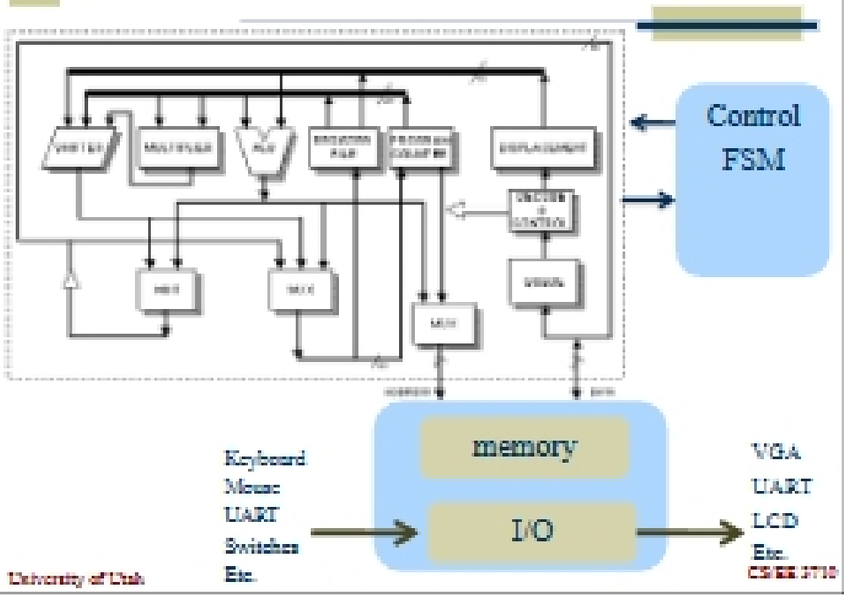


Multicycle MIPS microarchitecture. Reprinted from [Patterson04], with permission from Elsevier.

Generic Architecture



Generic Architecture



The Short-Term Picture

- Start with a review assignment
- Next assignment is a Finite State Machine (FSM) mapped to the Spartan-3E board
  - Thunderbird tail lights...
- Next assignment will be a very small processor
  - I'll hand out mips.v code from Weste/Harris
  - I'll hand out Verilog code for block RAMs
  - I'll hand out sample Fibonacci assembly code
  - You'll augment the processor with ADDI
  - You'll augment the processor with very simple I/O
  - You'll augment the Fibonacci code
- Then a VGA assignment
  - Everyone builds a VGA interface
  - VGA version of the Thunderbird...

The Medium Term Picture

- We'll hand out lab kits on Tuesday next week during class
  - We'll meet in the DSL, MEB 3133
- Be thinking about who to team up with
  - Teams will be 3-4 people
  - Good teams have a mix of complementary skills
- Start thinking about your project
  - Mid-term presentations
    - Present your plans and your design so far
    - All team members must participate and present

The Long Term Picture

- Once teams are formed (Late September)
  - Start working on your project
  - Start with baseline, augment for your application
  - Think about memory and I/O
  - Think about support software (assemblers, compilers, etc.)
  - Think about application software
- Whole thing due at the end of class
  - Demo day at the end of the semester
  - December 9th – 3:40-5:00pm