Last Time: Timing and Xilinx

From: Xilinx
Spartan 3 data sheet, simplified.

CS 152 L6: Teamwork

UC Regents Spring 2005 © UCB
Recap: Why Xilinx wires are so slow ... 

Wires are slow because (1) each green dot is a transistor switch (2) path may not be shortest length (3) all wires are too long!

The best Xilinx users “write Verilog to the grid”. When Xilinx designs FPGA chips, wiring channels are optimized for (2) & (3).
Reminder: What are the green dots?

Set during configuration.

One flip-flop and a pass gate for each switch point. In order to have enough wires in the channels to wire up CLBs for most circuits, we need a lot of switch points! Thus, “80%+ of FPGA is for wiring”.
Question: You are Xilinx ...

What are your tradeoffs for “wire design” when you design a new part?

Many startups looking at this question ...
Today: Teamwork

The lessons learned from the Fall 04 CS 152 class.

Design Notebook: How to keep a design notebook for your team.

In their own words: The final project presentation from a group whose final project did not make it to board.
CS 152
Computer Architecture and Engineering

What Went Right, What Went Wrong

2004-12-13
Dave Patterson, John Lazzaro
Doug Densmore, Ted Hong, Brandon Ooi

End-of-term presentation to CS hardware faculty ...

www-inst.eecs.berkeley.edu/~cs152/
Successful Start: Lab 2 (Single Cycle Processor) and Lab 3 (Pipelines) went well. Most groups finished on time.

Stressful End: Lab 4 (Caches): 1 group on time, 3 (?) were late, 1 never worked. Lab 5 (Final Project): 1 perfect project, 1 near miss, 2 worked in simulation.

What did we do after Lab 4?
We held a “town meeting” in class ...
Lab 4 “Town Meeting”

Held during one of the last Fall 04 classes ...
Lab 4: Reflections from the TAs

Everyone **worked hard**. Only in retrospect did most students realize they also had to **work smart**.

**Example**: Only one group member knows how to download to board. Once this member falls asleep, the group can’t go on working ...

**Solution**: Actually use the Lab Notebook to document processes. An example of **working smart**.
Lab 4: Reflections from the TAs

Example: Group has a long design meeting at start of project. Little is documented about signal names, state machine semantics. Members design incompatible modules, suffer.

A Better Way: Carry notebooks (silicon or paper) to meetings, and force documentation of the decisions on details.
Lab 4: Reflections from the TAs

Example: Comprehensive test rigs seen as a “checkoff item” for Lab report, done last. Actual debugging proceeds in haphazard, painful way.

A Better Way: One group spent 10 hours up front writing a cache test module. Brandon “The best cache testing I’ve ever seen”. They finished on time. An example of working smart.
Design Notebook
Why should you keep a design notebook?

- Keep track of the design decisions and the reasons behind them
  - Otherwise, it will be hard to debug and/or refine the design
  - Write it down so that you can remember in long project: 2 weeks -> 2 yrs
  - Others can review notebook to see what happened

- Record insights you have on certain aspect of the design as they come up

- Record of the different design & debug experiments
  - Memory can fail when very tired

- Industry practice: learn from others mistakes
Why do we keep it on-line?

° You need to force yourself to take notes
  • Open a window and leave an editor running:
    1) Acts as reminder to take notes
    2) Makes it easy to take notes
  • 1) + 2) => will actually do it

° Take advantage of the window system’s “cut and paste” features

° It is much easier to read typing than writing

° Also, paper log books have problems
  • Limited capacity => end up with many books
  • May not have right book with you.
  • Can use computer to search files.
How to do it? See “Resources” web page

° Keep it simple
  • DON’T make it too elaborate (fonts, layout, ...)

° Separate the entries by dates
  • type “date” command in another window and cut&paste

° Start day with problems going to work on today

° Record output of simulation into log with cut&paste; add date
  • May help sort out which version of simulation did what

° Record key email with cut&paste

° Record of what works & doesn’t helps team decide what went wrong after you left

° Index: write a one-line summary of what you did at end of each day
Goal: Layout the schematic for a 32-bit comparator

I've layed out the schematics and made a symbol for the comparator. I named it comp32. The files are

~/wv/proj1/sch/comp32.sch
~/wv/proj1/sch/comp32.sym
Goal: Test the comparator component

I've written a command file to test comp32. I've placed it in ~/wv/proj1/diagnostics/comp32.cmd.

I ran the command file in viewsim and it looks like the comparator is working fine. I saved the output into a log file called ~/wv/proj1/diagnostics/comp32.log

Notified the rest of the group that the comparator is done.
Goal: Investigate bug discovered in comp32 and hopefully fix it

Bart found a bug in my comparator component. He left the following e-mail.

Hey Bruce,
I think there's a bug in your comparator. The comparator seems to think that ffffffff and ffffffff7 are equal.

Can you take a look at this?
Bart
I verified the bug. here's a viewsim of the bug as it appeared..
(equal should be 0 instead of 1)

------------------
SIM>stepsize 10ns
SIM>v a_in A[31:0]
SIM>v b_in B[31:0]
SIM>w a_in b_in equal
SIM>a a_in ffffffff\h
SIM>a b_in fffffff7\h
SIM>sim
time =   10.0ns  A_IN=FFFFFFFF\H B_IN=FFFFFFF7\H EQUAL=1
Simulation stopped at 10.0ns.
------------------

Ah. I've discovered the bug. I mislabeled the 4th net in
the comp32 schematic.

I corrected the mistake and re-checked all the other
labels, just in case.

I re-ran the old diagnostic test file and tested it against
the bug Bart found. It seems to be working fine. hopefully
there aren't any more bugs:)}
On second inspection of the whole layout, I think I can remove one level of gates in the design and make it go faster. But who cares! the comparator is not in the critical path right now. the delay through the ALU is dominating the critical path. so unless the ALU gets a lot faster, we can live with a less than optimal comparator.

I e-mailed the group that the bug has been fixed

Mon Sep 11 14:03:41 PDT 1995

- ====================================================================

• Perhaps later critical path changes; what was idea to make comparator faster? Check log book!
Administrivia: Upcoming deadlines ...

Friday 2/4: “ModelSim Checkoff”, 12-1, 119 Cory. For 61(c) students, 150 Lab Lecture 3”, 1-2 PM, 125 Cory.

Friday 2/11: “Xilinx Checkoff”, 12-1, 119 Cory. For 61(c) students, 150 Lab Lecture 4”, 1-2 PM, 125 Cory.

Monday 2/14: Lab 2 final report due via the submit program, 11:59 PM.
Back to Fall 04 ...
Successful Start: Lab 2 (Single Cycle Processor) and Lab 3 (Pipelines) went well. Most groups finished on time.

Stressful End: Lab 4 (Caches): 1 group on time, 3 (?) were late, 1 never worked. Lab 5 (Final Project): 1 perfect project, 1 near miss, 2 worked in simulation.

Let's take a look at the final project presentation of a “worked in simulation, not on the board” group.
Other Teamwork Topics
Example: We recommended using the CoreGen multiplier generators to Fall 04. The tool was buggy (my bad). The most successful groups realized this early and switched methods.

Lesson: Most CAD has bugs, and we can’t know them all. Be paranoid -- never blindly trust any CAD tool!
Verilog: Carefully written Verilog will yield **identical semantics** in ModelSim and Synplicity. If you write your code in this way, many “works in ModelSim but not on Xilinx” issues **disappear**.

Always check log files, and inspect output tools produce!

**Backups:** Use CVS, but also make **safety copies off-site** regularly (gmail). New CVS users often lose work as they are learning how to use CVS. Beware of CVS **NT permissions issues**.
Schematics: This schematic uses wires
This schematic uses labels ...

Which is easier to understand?
CAD and Testing: Asset Management

* Agree on where Verilog files will reside in the file directory structure.

* Agree on placement of test bench Verilog and hardware Verilog files.

* Agree on standard way to name files, and standard way to name Verilog modules, variables, parameters, ....

* Don’t copy files -- include them. Each file should exist once in file tree.
Group Dynamics: How to Disagree

Example: 3 members want to do the design one way; member number 4 does not agree.

Solution #1: Voting. “Fair”. But, what if the “loser” was technically correct?

Solution #2: Consensus. Keeping in mind the goal (correctly working CPU on the board on schedule), what option brings the group closer to the goal?

Never lose sight of the goal!
Group Dynamics: Humility is important!

It is certainly of more consequence to a man, that he has learnt to **govern his passions** in spite of temptation, to be **just in his dealings**, to be temperate in his pleasures, to support himself with fortitude under his misfortunes, to behave with prudence in all his affairs and every circumstance of life; I say, it is of much more real advantage to him to be thus qualified, **than to be a master of all the arts and sciences** in the world beside. Virtue alone is sufficient to make a man great, glorious, and happy.

-- Ben Franklin

More at: [http://www.cs.berkeley.edu/~dsw/](http://www.cs.berkeley.edu/~dsw/)

Thanks to Daniel S. Wilkerson
Where we are now, and what is next

<table>
<thead>
<tr>
<th>Date</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th 2/3</td>
<td>Teamwork</td>
</tr>
<tr>
<td>F 2/4</td>
<td></td>
</tr>
<tr>
<td>Sa 2/5</td>
<td></td>
</tr>
<tr>
<td>Su 2/6</td>
<td></td>
</tr>
<tr>
<td>M 2/7</td>
<td></td>
</tr>
<tr>
<td>T 2/8</td>
<td>Performance</td>
</tr>
<tr>
<td>W 2/9</td>
<td></td>
</tr>
<tr>
<td>Th 2/10</td>
<td>Pipelining I</td>
</tr>
</tbody>
</table>

Software for teamwork, group dynamics, etc ...

Pipelining begins ...

CS 152 L6: Teamwork

UC Regents Spring 2005 © UCB