Assign final project

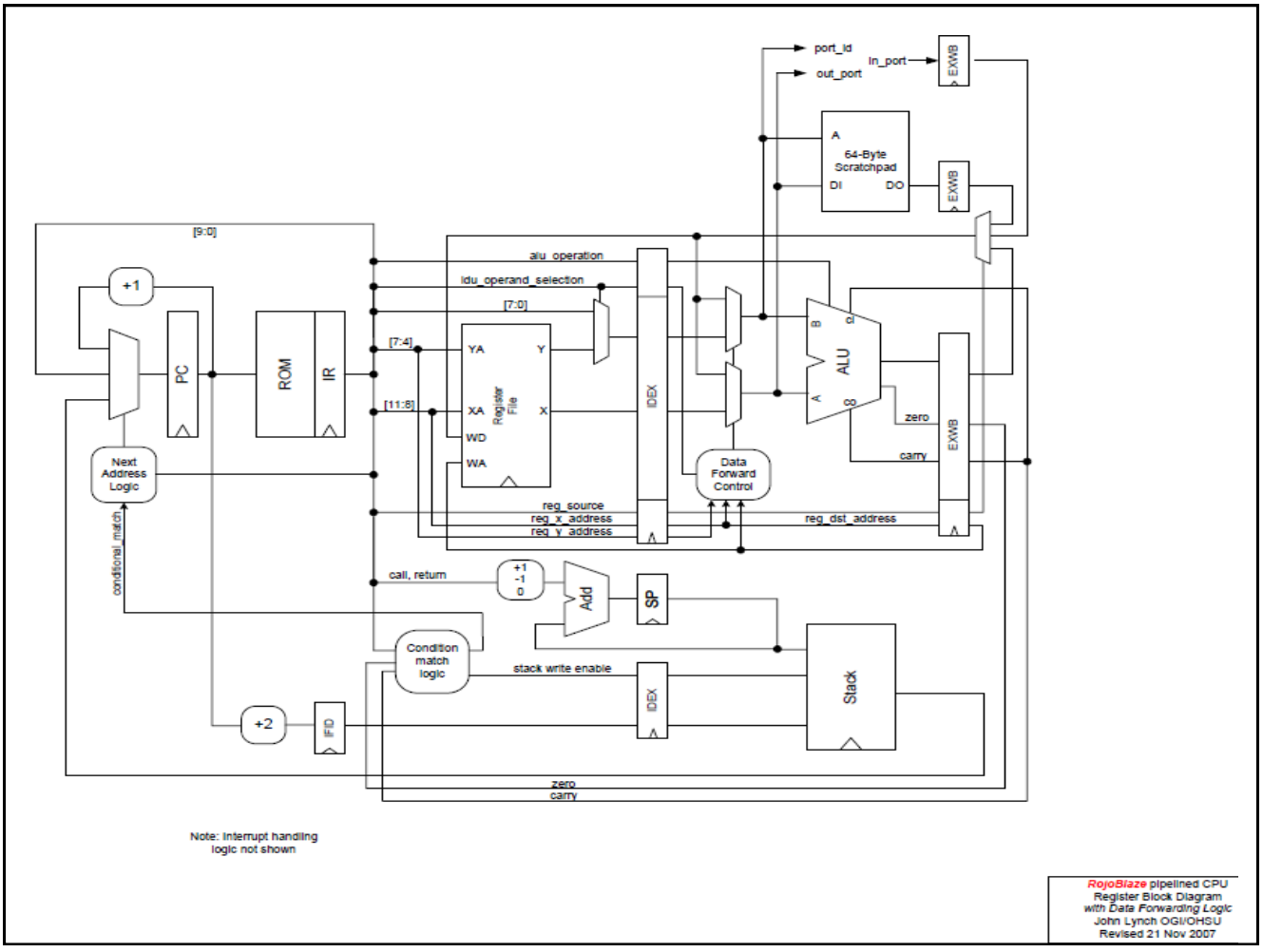
Project will be released by Mon, 08-Nov

Roy's project idea

Project will be released by Sat, 06-Nov

What is the pipelined Picoblaze?

- Picoblaze is an 8-bit microcontroller "soft core" that can be incorporated into an FPGA-based SoC
  - Conceived and implemented by Ken Chapman, a Xilinx FAE, Circa 2010 in its current incarnation…but much older than that
  - Supported on many Xilinx FPGA families including the Series 7 FPGA families we target in ECE 540 and ECE 544
  - Synthesizable but not Verilog as we think of it…specified at the LUT-level
- Pacoblaze is a clone of Picoblaze but written as a synthesizable RTL model
  - Written and (was) supported by Pablo Bleyer Kocik (Last update was in 2007)
- Rojoblaze (Roy-John Picoblaze) is a pipelined version of the Pacoblaze written by John D. Lynch (https://directory.vancouver.wsu.edu/people/john-lynch) and Roy Kravitz, Circa 2007


port_id
out_port
in_port
EXWB
A
64-Byte
Scratchpad
DI
DO
[9:0]
+1
PC
ROM
IR
alu_operation
idu_operand_selection
[7:0]
[7:4]
[11:8]
YA
Y
XA
X
WD
WA
Register
File
IDEX
B
CI
A
CO
ALU
zero
carry
Next
Address
Logic
Data
Forward
Control
reg_source
reg_x_address
reg_y_address
reg_dst_address
conditional_match
call, return
+1
-1
0
Add
SP
Condition
match
logic
stack write enable
Stack
+2
IFID
zero
carry
Note: Interrupt handling
logic not shown
RojoBlaze pipelined CPU
Register Block Diagram
with Data Forwarding Logic
John Lynch OGI/OHSU
Revised 21 Nov 2007

4

Expectations

- Understand the ISA, write test cases, assemble the code using KCPSM and run the test cases on the core
- Understand the core and co-relate it with the ISA
- Write and execute a verification plan
  - Assertions, checkers, randomization, etc.
- Find bugs in a "broken" core.
- Produce a verification report based on your Verification plan
  - Should include coverage statistics from QuestaSim
- Present your findings in a 20-minute presentation on Mon, 06-Dec or Tue, 07-Dec
  - Doodle calendar will be posted as we get closer to Final's week

Self-enrolled teams of 3

Emulation, formal verification "adders"

- A fourth member of the team can be added to work on Veloce emulation, Formal Verification, or other verification tools/processes.
  - Email a proposal for this additional work including the scope of the work, the name of the student being added to the team, and the benefits from adding the additional capabilities to the project
  - Work of the 4th team member must be substantive – I feel that each team member should make a significant contribution to the effort.
- For more details on working on the emulator, please refer to the Veloce Solo user guide, as well as the examples given on the Veloce Solo server (ssh username@velocesolo.ece.pdx.edu)
  - Need to let us know ASAP if you are going to use Veloce so we can get you access to the server and emulator
  - Warning: Neither Apoorva nor I are Veloce users, let alone experts…you will be pretty much on your own

Things to remember

- Teams of 3 - Github Classroom and D2L (Teams of 4 w/ approval)
- *Please make sure all previously leveraged code is compiled and error free before executing your verification plan.*
- Grading:
  - (25 pts) Verification plan
  - (40 pts) Final project presentation
  - (25 pts) The quality of your design report and completed verification plan
  - (10 pts) The quality/readability of your source code
  - (up to 5) extra credit points

Important dates

- Wed, 03-Nov: Final project assigned during class
- Mon, 08-Nov:
  - Pipelined Picoblaze project released to D2L on Sat, 06-Nov
  - Apoorva will introduce her project ideas in class on Mon, 08-Nov (tentative)
- Sun, 28-Nov: Verification plan submitted to D2L by 10:00pm
- Mon, 29-Nov: "Broken" Picoblaze model released
- Wed, 01-Dec: Final Exam (during class time)
- Mon, Tue 06-Dec and 07-Dec: Final Project Presentations
- Wed, 08-Dec: Final project deliverables due to GitHub and D2L by 10:00pm