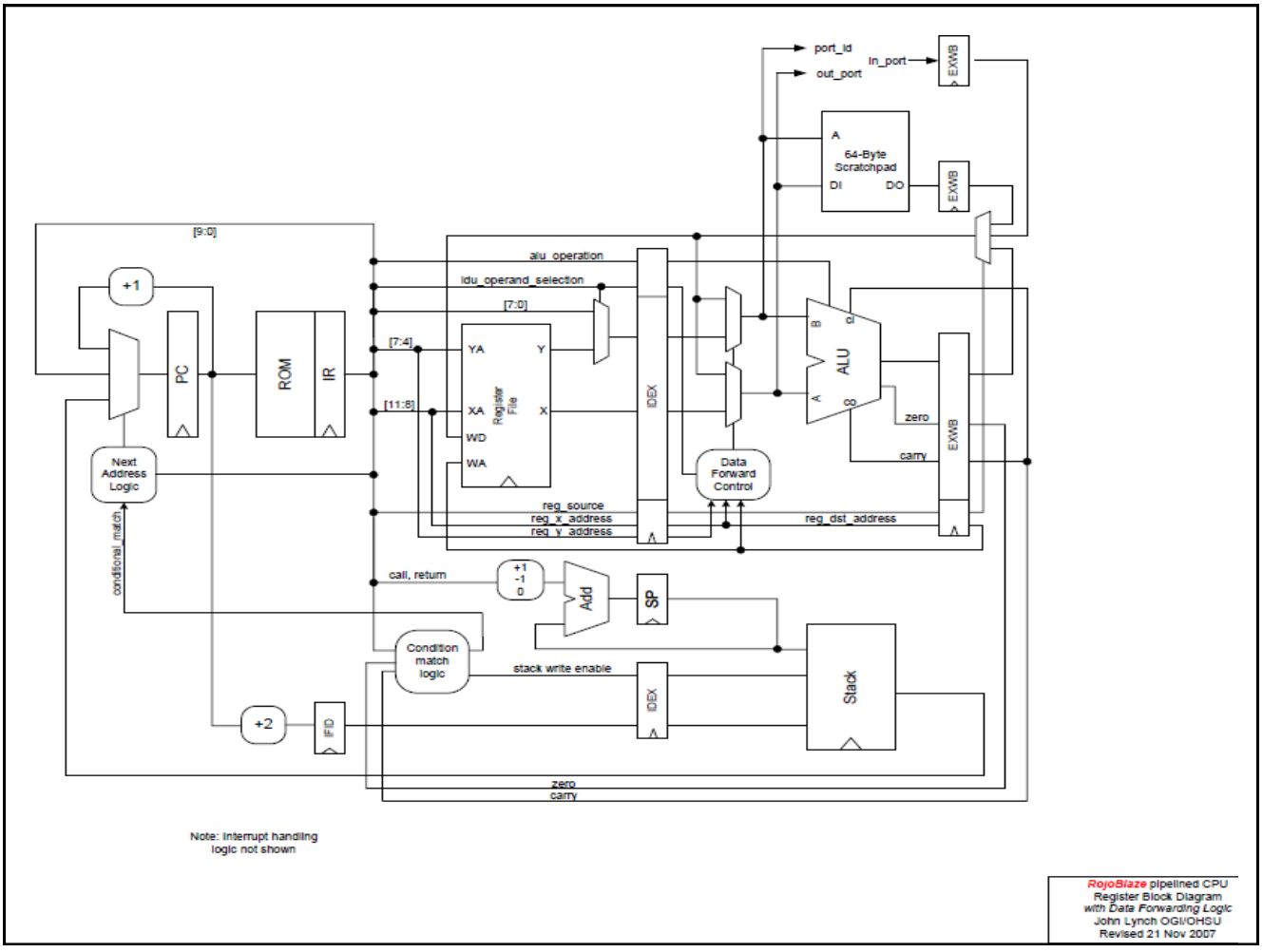
Introduce final project

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

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 Tue, 06-Dec or Wed, 07-Dec
  - Signup calendar will be posted as we get closer to Final's week

Self-enrolled teams of 3

Things to remember

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

Important dates

- Wed, 09-Nov: Final project assigned during class
- Tue, 15-Nov: Pipelined Picoblaze project released to Canvas and GitHub Classroom
- Sat, 26-Nov: Verification plan submitted to Canvas by 10:00pm
- Tue, 29-Nov: "Broken" Picoblaze model released
- Wed, 30-Nov: Final Exam (during class time)
- Tue, 06-Dec, Wed, 07-Dec: Final Project Presentations
- Thu, 08-Dec: Final project deliverables due to GitHub and Canvas by 10:00pm