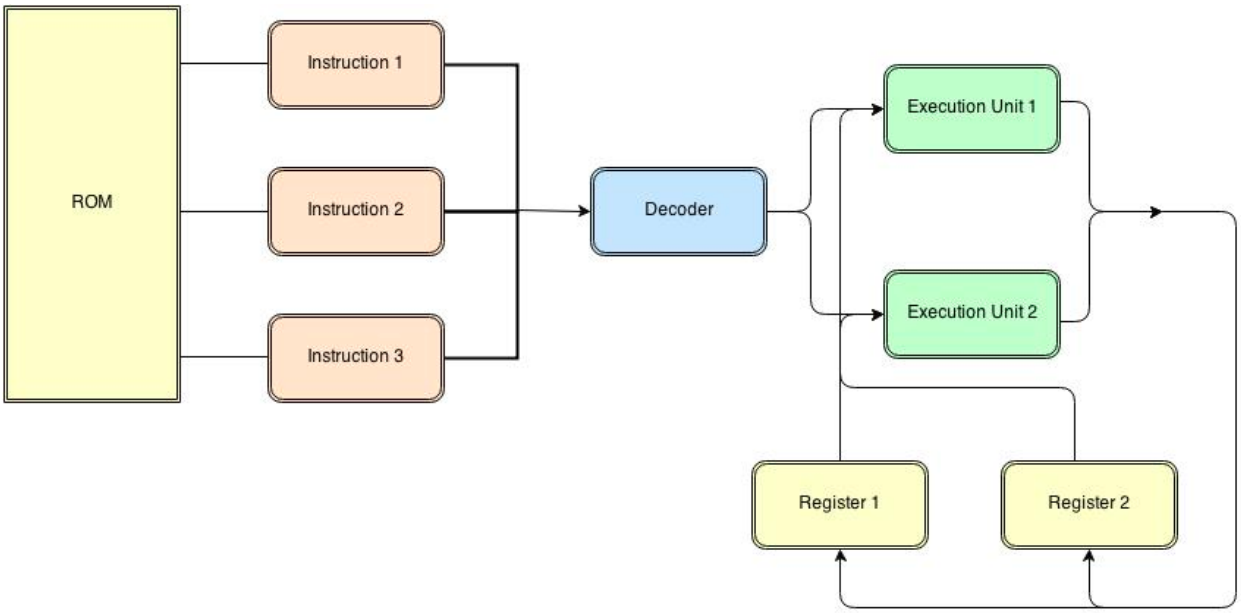
Constructing a Superscalar Processor in Logisim

Aven Bross

General CPU Specifications

- ❏ 8 bit register size
- ❏ Two registers
- ❏ 8 bit instruction size
- ❏ Two integer execution units

Flow Diagram



Procedure Overview:

1) Three instructions read off of ROM line.
2) Decoder looks at first two and decides whether they can be executed together.
3) If so, execute both and read two more from ROM.
   Otherwise execute one and read one more from rom.
4) Write back to registers
5) Repeat.

Instruction Set

Bits	Name	Description
0	S1	Destination Register 0: Reg 0 1: Reg 1
1, 2	S2	Other Register/Constant 00: Reg 0 01: Reg 1 10: low # 11: high #
3, 4, 5	C	Constant Selection low #: 000-111 [00-07] high #: 000-111 [08-0f]
6, 7	Op	Operation Selection 00: Set S1 to S2 01: Add S2 to S1 10: Multiply S1 by S2 11: Left shift S2 by 4 and store in S1

An example programs

0x15 - 00 010 10 1 - Set Reg 1 to 0x02
0x1c - 00 011 10 0 - Set Reg 0 to 0x03
0x83 - 10 000 01 1 - Multiply Reg 1 by itself
0x54 - 01 010 10 0 - Add 2 to Reg 0
0x42 - 01 000 01 0 - Add Reg 1 to Reg 0
0x05 - 00 000 10 1 - Set Reg 1 to 0x00
0x04 - 00 000 10 0 - Set Reg 0 to 0x00

Dealing with data dependencies

Read after Write:

Decoder unit ensures that to run together, the second instruction does not need a write from the first.

Write after Write:

Decoder unit ensures that to run together, two instructions are not writing to the same register.

Discuss and demonstrate Logisim simulation

This will be the majority of the presentation where I will demonstrate and explain how each unit works and what decisions I made in the process of constructing the processor.

Reference Material:

Our class notes
https://www.cs.uaf.edu/courses/cs441/notes/superscalar/

Nice little pdf on superscalar
http://www.cis.upenn.edu/~milom/cis501-Fall11/lectures/07_superscalar.pdf

Research paper on microarchitecture of superscalar
http://minds.wisconsin.edu/bitstream/handle/1793/9476/file_1.pdf?sequence=1