Lab 4 preview

Hung-Wei Tseng

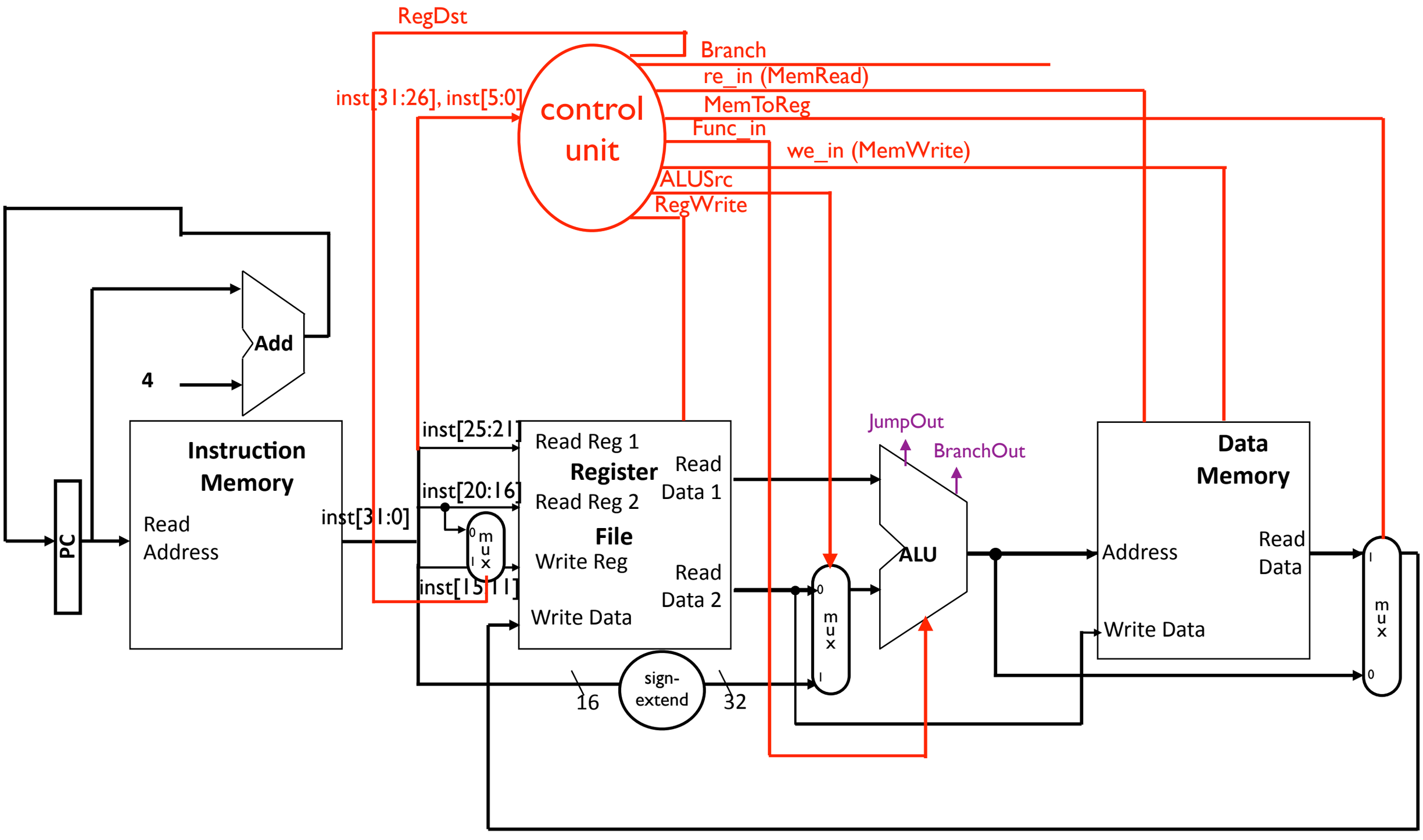
Announcement

- Lab 3 due tomorrow
  - Interview with any of us, and send a copy of your source to the person who you interviewed with
  - All of us will be there between 3:30p-5:00p. Please try finishing interview during that period.
- Lab 4 due next Thursday
  - Please complete the lab as soon as possible

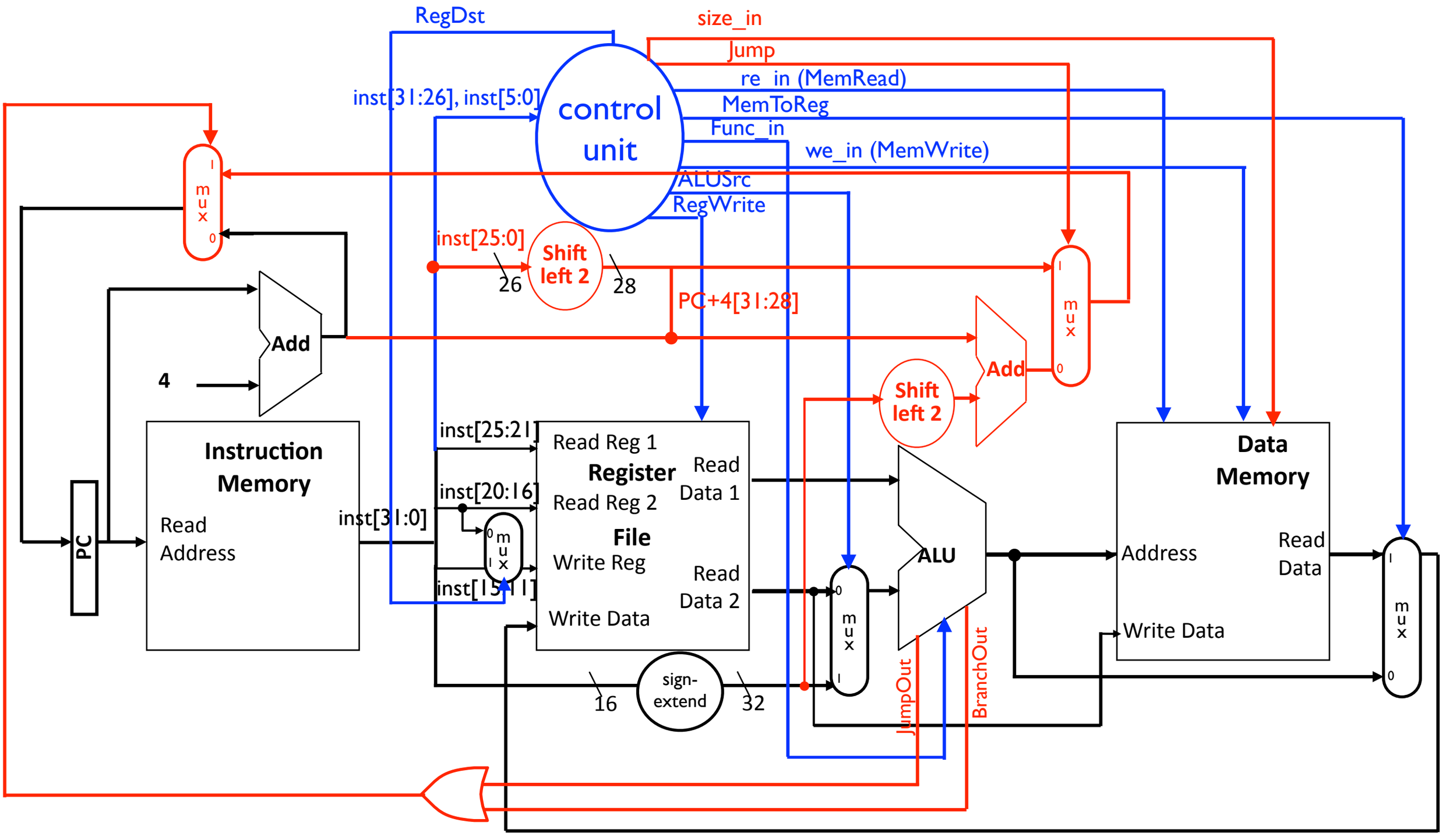
In Lab 3...

- You will be extending the datapath and control unit to support branch instructions!
  - The processor already support lw, sw, add, addi, sub, and, or nor, xor
  - We need to support
  - beq, bne, bltz, bgez, blez, bgtz, jump, jr, jal, jalr
  - lb, lh, sb, sh, lbu, lhu
  - addu, addiu, subu, andi, ori, xori, lui, slt, sltu

Lab 3: adding control to Lab 2!



Lab 4: adding more to Lab 3!



This schematic only support R-type+bne,beq,jump

Control Unit

- We already support lw, sw, add, addi, sub, and, or, nor, xor.

	instruction			control unit output								
	type	opcode inst[31:26]	funct inst[5:0]	func_in	RegDst	ALUSrc	RegWrite	MemRead	MemWrite	MemTo Reg	Jump	size_in
lw	I	0x23		100000	0	1	1	1	0	1	0	11
sw	I	0x2b		100000	X	1	0	0	1	X	0	11
add	R	0	0x20	100000	1	0	1	0	0	0	0	XX
addi	I	0x8		100000	0	1	1	0	0	0	0	XX
sub	R	0	0x22	100010	1	0	1	0	0	0	0	XX
and	R	0	0x24	100100	1	0	1	0	0	0	0	XX
or	R	0	0x25	100101	1	0	1	0	0	0	0	XX
nor	R	0	0x27	100111	1	0	1	0	0	0	0	XX
xor	R	0	0x26	100110	1	0	1	0	0	0	0	XX

Control Unit (extended)

	instruction			control unit output								
	type	opcode inst[31:26]	funct inst[5:0]	func_in	RegDst	ALUSrc	RegWrite	MemRead	MemWrite	MemTo Reg	Jump	size_in
lb	I	0x20		100000	0	1	1	1	0	1	0	00
lh	I	0x21		100000	0	1	1	1	0	1	0	01
sb	I	0x28		100000	X	1	0	0	1	X	0	00
sh	I	0x29		100000	X	1	0	0	1	X	0	01
lbu	I	0x24		100000	0	1	1	1	0	1	0	00
lhu	I	0x25		100000	0	1	1	1	0	1	0	01
beq	I	0x4		111100	X	0	0	0	0	0	0	XX
bne	I	0x5		111101	X	0	0	0	0	0	0	XX
bltz	I	0x1		111000	X	0	0	0	0	0	0	XX
bgez	I	0x1		111001	X	0	0	0	0	0	0	XX
blez	I	0x6		111110	X	0	0	0	0	0	0	XX
bgtz	I	0x7		111111	X	0	0	0	0	0	0	XX

Control Unit (extended)

	instruction			control unit output								
	type	opcode inst[31:26]	funct inst[5:0]	func_in	RegDst	ALUSrc	RegWrite	MemRead	MemWrite	MemTo Reg	Jump	size_in
addu	R	0x0	0x21	100001	1	0	1	0	0	0	0	XX
addiu	I	0x9		100001	0	1	1	0	0	0	0	XX
subu	R	0x0	0x23	100011	1	0	1	0	0	0	0	XX
andi	I	0xC		100100	0	1	1	0	0	0	0	XX
ori	I	0xD		100101	0	1	1	0	0	0	0	XX
xori	I	0xE		100110	0	1	1	0	0	0	0	XX
slt	R	0x0	0x2A	101000	1	0	1	0	0	0	0	XX
sltu	R	0x0	0x2B	101001	1	0	1	0	0	0	0	XX
j	J	0x2		111010	0	0	0	0	0	0	1	XX

We still need to support...

- lui (I-type)
  - $rt = {immediate, 16'b0}
- jr (R-type, func = 0x8)
  - PC = $rs
- jal (J-type)
  - $ra = PC+4
  - PC = {PC+4[31:28], imm << 2}
- jalr (R-type, func = 0x9)
  - $rd = PC+4
  - PC = $rs

Your task

- Modify the schematic to support all the required instructions
- Extend the control unit to support all the required instructions

Benchmarks

- In this lab, we provide three following benchmark programs in http://cseweb.ucsd.edu/classes/su12/cse141L-a/Media/lab4/lab4-files.zip
  - No branch hello world
  - Hello world with branch
  - Fibonacci number
  - Start with PC 0x400000
    - That's why the default PC is 0x3FFFFC
    - But depends on your hardware design, you don't have to make it 0x3FFFFC.

Interview questions

- Show the schematics
- Show the waveforms of three benchmarks until the end
- Measure the IC, total cycles, CPI
- Report the Fmax
  - We can calculate the performance of your processor now!

Q & A