CSEE W3827: Fundamentals of Computer Systems

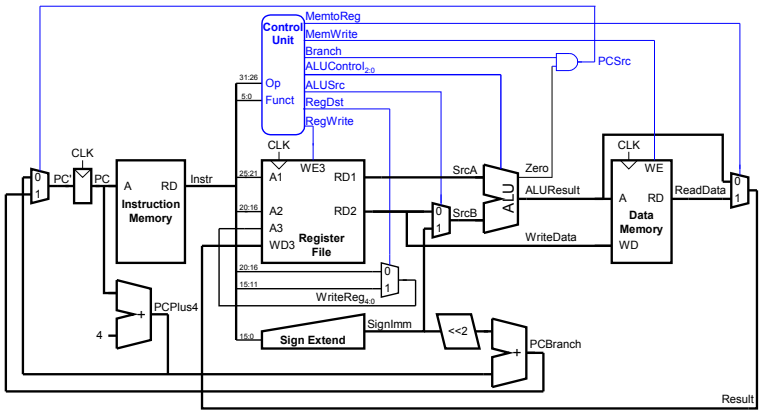
Homework Assignment 4. Stephen Edwards. Columbia University

Due Sunday, Jun 13 at 11:59 PM EDT via Courseworks

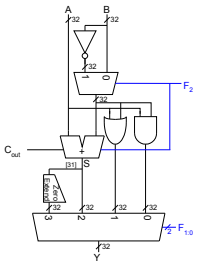
Put your answers in the dashed boxes provided and upload a PDF file to Courseworks. E.g., by editing it in Inkscape (`https://inkscape.org`) or printing and scanning.

Name: Uni:

1. (25 pts.) Extend the single-cycle MIPS processor to support the `jal` instruction (j-type, OP=000011, "`jal` *immediate*") You may modify the datapath, ALU, ALU decoder, and Main Decoder any way you want, but make your modifications clear. In particular, complete the "jal" row for the Main Decoder.



ALU Decoder			
ALU Op	**Funct**	**ALU Ctrl.**	**ALU Func.**
00	—	010	Add
-1	—	110	Sub
1-	100001	010	Add
1-	100011	110	Sub
1-	100100	000	AND
1-	100101	001	OR
1-	101010	111	Slt



F210	Func.
000	A & B
001	A | B
010	A + B
011	
100	A & $\overline{B}$
101	A | $\overline{B}$
110	A − B
111	A < B (slt)

Main Decoder								
Inst.	**OP**	**RegWrite**	**RegDst**	**ALUSrc**	**Branch**	**MemWrite**	**MemToReg**	**ALUOp**
R-type	000000	1	1	0	0	0	0	1-
`lw`	100011	1	0	1	0	0	1	00
`sw`	101011	0	-	1	0	1	-	00
`beq`	000100	0	-	0	1	0	-	01
`jal`								

2. Assuming the following dynamic instruction frequency for a program running on the single-cycle MIPS processor discussed in class,

`addu`	30%
`lw`	20%
`addiu`	15%
`beq`	15%
`sw`	15%
`j`	5%

   (a) (5 pts.) In what fraction of cycles is data memory written?
   %

   (b) (5 pts.) In what fraction of cycles is the ALU used?
   %

   (c) (5 pts.) In what fraction of cycles must ALUSrc = 1?
   %

3. Consider running the following loop on the five-stage MIPS pipeline with full bypassing (i.e., M-to-E, W-to-E, and M-to-D bypasses) and stall logic as described in class.

```
again:
i1: lw   $t1, 4($t0)
i2: lw   $t2, 8($t1)
i3: addu $t3, $t3, $t2
i4: addiu $t4, $t4, 1
i5: addiu $t0, $t0, -8
i6: bne  $t0, $t5, again
```

   (a) (10 pts.) List all control and data dependencies that cause pipeline stalls (e.g., i1 → i2)

   (b) (15 pts.) Below, rewrite the above code by reordering instructions (keep the instruction labels) to eliminate as many stalls as possible. List any remaining stalls.

4. For the caches below with 32-bit addresses, list the number of bits in each field:

   (a) (5 pts.) 65536B total, 4-way set-associative, 64B blocks
   Tag: Set Index: Byte Offset:

   (b) (5 pts.) 262144B total, 4-way set-associative, 64B blocks
   Tag: Set Index: Byte Offset:

5. Consider a computer with a *direct mapped* cache of 32 64-byte blocks backed by 1 megabyte of main memory.

   (a) (5 pts.) How many blocks in main memory?

   (b) (5 pts.) What is the cache set for address `0xDECADE`?

   (c) (15 pts.) Assuming the cache starts empty, what sequence of events would be produced by reading bytes in the following sequences of addresses? Classify each as a compulsory miss, a conflict miss, a spatial locality hit, or a temporal locality hit.

Address	**Event**
`0xDECADE`	
`0xDECAD8`	
`0xDECA08`	
`0xBECADE`	
`0xDECADE`	