

Name: _____

CSE 30321 – Computer Architecture I – Fall 2010
Midterm Exam
October 14, 2010

Test Guidelines:

1. Place your name – or at least your initials! – on *****EACH***** page of the test in the space provided.
2. Answer every question in the space provided. If separate sheets are needed, make sure to include your name and clearly identify the problem being solved.
3. Read each question carefully. Ask questions if anything needs to be clarified.
4. The exam is open book and open notes.
5. All other points of the ND Honor Code apply. By writing your name on the exam, you agree to abide by the ND Honor Code.
6. Upon completion, please turn in the test and any scratch paper that you used.

Suggestion:

- Whenever possible, show your work and your thought process. This will make it easier for us to give you partial credit.

Name: _____

Score Sheet

Question	Possible Points	Your Points
1	15	
2	15	
3	20	
4	10	
5	15	
6	10	
7	15	
Total	100	

Name: _____

Problem 1: (15 points)

This question deals with the 6-instruction ISA that was discussed in Lecture 02 and Lecture 03. As you saw in Lecture 02 and Lecture 03, the instruction encodings for the 6-instruction processor are as shown in the table below:

Instruction	Opcode	16-bit encoding				Function
Mov Ra, d	0000	Opcode (4 bits)	Destination Register (4 bits)	Address (8 bits)		$RF[a] \leftarrow M[d]$
Mov d, Ra	0001	Opcode (4 bits)	Source Register (4 bits)	Address (8 bits)		$M[d] \leftarrow RF[a]$
Add Ra,Rb,Rc	0010	Opcode (4 bits)	Destination Register (4 bits)	Source Register (4 bits)	Source Register (4 bits)	$RF[a] \leftarrow RF[b] + RF[c]$
Mov Ra, #C	0011	Opcode (4 bits)	Destination Register (4 bits)	Constant (8 bits)		$RF[a] \leftarrow c$
Sub Ra,Rb,Rc	0100	Opcode (4 bits)	Destination Register (4 bits)	Source Register (4 bits)	Source Register (4 bits)	$RF[a] \leftarrow RF[b] - RF[c]$
Jumpz Ra, X	0101	Opcode (4 bits)	Source Register (4 bits)	Offset (8 bits)		If $RF[a] == 0$, $PC \leftarrow PC + \text{offset}$

Question A: (5 points)

Assume that you want to augment this ISA to support 20 additional and unique instructions (e.g. Mult, And, Or, etc.), while still keeping the instruction encoding as 16 bits. How will the execution and encoding of the Add instruction be affected? (Other instructions could be affected too, but you just need to comment on how the Add instruction will be impacted.)

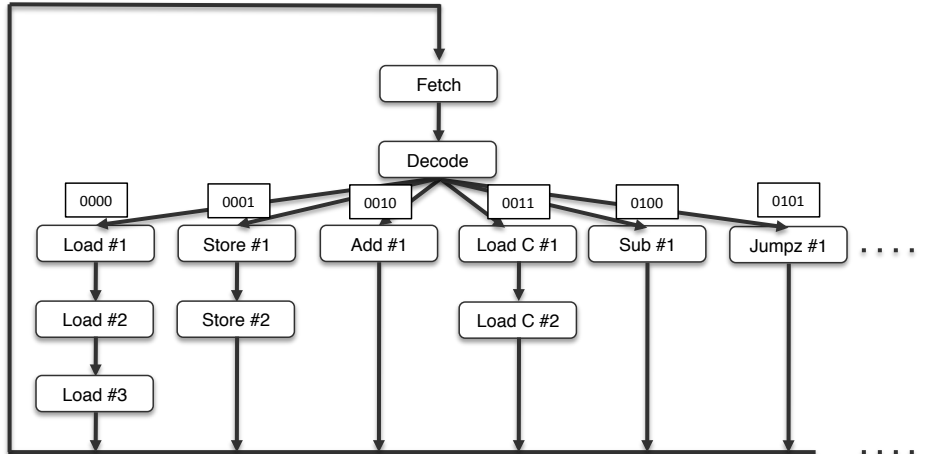
Name: _____

Question B: (5 points)

Now, assume that the 20 new instructions have all been added. Their addition has resulted in changes to the state diagram discussed in lecture. A portion of the new finite state diagram is shown below. Given this new state diagram, how many clock cycles will the code (that is also shown below) take to run?

Code:

```
Mov      R1, #1
Mov      R2, #2
Mov      R4, #4
Sub      R5, R4, R2
Sub      R5, R5, R2
Jumpz   R5, X
Add      R1, R1, R1
X:      Add R1, R2, R4
Mov      10, R1
```



Question C: (5 points)

In what clock cycle(s) does R1 change state? (i.e. when is new data that is put into R1 available?)

Name: _____

Problem 2: (15 points)

Question A: (4 points)

Consider a hypothetical branch-if-equal instruction that is 32 bits long:

- 6 bits are used to encode the opcode
- 6 bits are used to encode one register number
- 6 bits are used to encode another register number
- 14 bits are used to encode an offset that will be added to the program counter (PC) if the branch ends up being taken, and a new instruction address is required.
 - o (The number is not in 2s complement form, and all 14 bits can encode a constant.)

Thus, the instruction syntax might be: BEQ R12, R11, X

- If $R12 == R11$, the PC will be set to $PC + X$ instead of $PC + 4$.

Given this instruction, will the code shown in the table below work? Why or why not?

Address	Instruction
5000	...
5004	BEQ R12, R11, X
5008	Add R1, R2, R3
...	...
X: 21256	Sub R1, R2, R3

Question B: (4 points)

Assuming that the PC has already been incremented by 4 when the comparison for the BEQ instruction at address 5004 is made, how many *instructions* away from the BEQ instruction could we reach?

Name: _____

Question C: (7 points)

Assume that you have 24-bit instructions. A hypothetical “R-type” / ALU instruction (i.e. add, subtract, multiply, etc.) might be encoded as follows:

Opcode	Destination <i>and</i> Source Register	Source Register	Function Code
6 bits	6 bits	6 bits	6 bits

Thus, 1 register serves as both a source *and* a destination:

Add R5, R7 # R5 \leftarrow R5 + R7

Given this type of encoding – i.e. where one register is always both a source and a destination – can the code shown below be translated into assembly with just these types of ALU instructions? If yes, write your code. If no, explain what functionality is missing.

```
...  
y = y + x;  
z = z * q;  
q = y + z;  
z = z * y;  
q = q + z;  
...
```

Name: _____

Problem 3: (20 points)

Assume that to spell check a large file, 820,000,000 instructions are needed. The instructions in the program are broken down into 4 different classes, and each class requires N clock cycles to execute. Specific information is given in the table below. (Here, N is the same as in the MIPS multi-cycle datapath discussed in class.)

Instruction Class	Clock Cycles per Instruction	Number of Instructions
Branch	3	150,000,000
Store	4	185,000,000
Load	5	260,000,000
ALU / R-type	4	225,000,000

Question A: (5 points)

If the total execution time for this program is found to be 1.57 seconds, what is the clock cycle time of the computer on which it was run?

Name: _____

Question B: (10 points)

Assume that as part of the 820,000,000 instruction spell check, 25% of all load instructions are immediately followed by an ALU / R-type instruction that uses the data that was just loaded. To speed up this program, we are contemplating adding a new type of instruction – an ALU instruction where one of the source operands is a value from memory.

- This new instruction will replace the previous, 2 instruction sequence.
- It will take 7 clock cycles.

Will this change offer any speedup over the original design? If so, how much?

You may assume that the clock rate does not change, and your answer to this question does not depend on your answer to Question A.

Question C: (5 points)

Qualitatively, if you see a speedup, where does it come from? If you do not, why not?

Name: _____

Problem 4: (10 points)

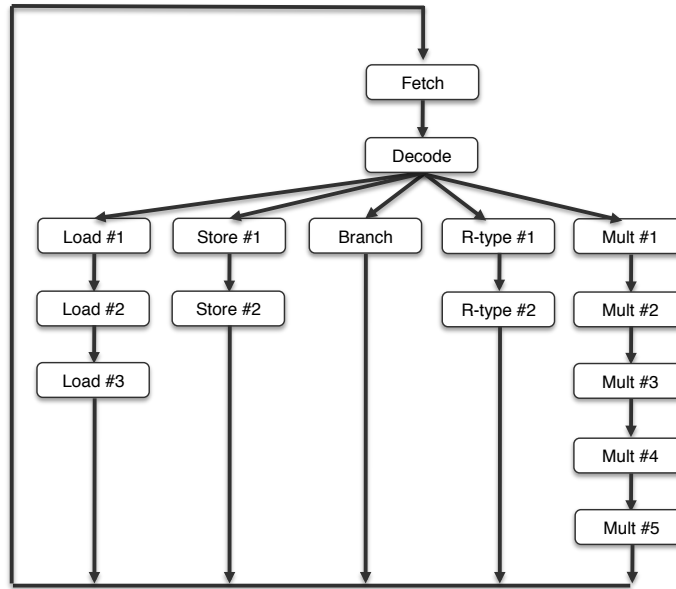
The number of instructions (of a given type) needed to encrypt and decrypt a message are as shown in the table below:

Instruction Class	Cycles Per Instruction	Encrypt	Decrypt
Branch	3	4,000,000	4,000,000
Store	4	10,000,000	9,000,000
Load	5	28,000,000	25,000,000
ALU / R-type	4	23,000,000	22,000,000
Totals:		65,000,000	60,000,000

You are considering changing the datapath that these benchmarks are run on so that a load instruction completes in 4 CCs instead of 5. Because the load instruction has to now do more work in a given clock cycle, that clock cycle will need to get longer. What clock cycle slow down is tolerable such that the performance of the encrypt and decrypt benchmarks is not degraded? You may assume that for every message decrypted, a message is also encrypted (and thus, there is equal use).

Problem 5: (15 points)

Below is a state diagram for a hypothetical processor.



Consider the following for loop: `for (i=0; i<N; i++) {
 x(i) = x(i) * 3;
 }`

Question A: (10 points)

Which MIPS translation – Version 1 or Version 2 – do you think is the most efficient? Why?

Version 1:

X: `addi $1, $0, 1
 addi $2, $0, N
 sll $3, $1, 2
 add $4, $3, $5
 lw $5, 0($4)
 add $5, $5, $5
 add $5, $5, $5
 sw $5, 0($4)
 addi $1, $1, 1
 bneq $1, $2, X`

Version 2:

X: `addi $1, $0, 1
 addi $2, $0, N
 addi $10, $0, 3
 sll $3, $1, 2
 add $4, $3, $5
 lw $5, 0($4)
 mult $5, $5, $10
 sw $5, 0($4)
 addi $1, $1, 1
 bneq $1, $2, X`

Question B: (5 points)

Given your answer, how could you make the above MIPS code even more efficient?

Name: _____

Problem 6: (10 points)

Assume the following:

- Data for 4 arrays – A, B, C, and X – is stored in memory.
- You may assume that:
 - The data elements for array A are stored in sequential memory addresses.
 - The data elements for array B are stored in sequential memory addresses.
 - The data elements for array C are stored in sequential memory addresses.
 - The data elements for array X are stored in sequential memory addresses.
 - However, arrays A, B, C, and X **are not** stored sequentially in memory.
- The starting address of array A is contained in \$1.
- The starting address of array B is contained in \$2.
- The starting address of array C is contained in \$3.
- The starting address of array X is contained in \$4.

Write the MIPS assembly instructions for the following statement:

`X[i] = A[B[i]] + C[B[i+4]];`

To receive full credit, your answer should contain no more than 10 instructions. You may assume `i` maps to \$5.

Please comment your code!

Name: _____

Problem 7: (15 points)

Consider the following C-code:

```
int main(void) {
    int i=0;           # i maps to $s0
    int j=1;           # j maps to $s1
    int k=2;           # k maps to $s2
    int l, m, n, o, p, q; # see below
    int x, y, z;       # see below

    l = i+j+k;        # l maps to $s3
    m = i*j*k;        # m maps to $s4
    n = l-m;          # n maps to $s5
    o = i+j;          # o maps to $s6
    p = m-n;          # p maps to $s7
    q = n+o-p;        # q maps to $t1

    x = function_call_1(o,p,q);

    m = l+x+j;

    y = function_call_2(m,l,x);

    z = x+y;
}

int function_call_1(x,y,z) {
    int a;           # a maps to $s1
    int b;

    a = x+y+z;
    b = function_call_3(a);

    return b;
}

int function_call_2(x,y,z) {
    int a;           # a maps to $s1

    a = x-y-z;

    return a;
}

int function_call_3(x) {
    return x + x;
}
```

Name: _____

Question A: (3 points)

Write the MIPS assembly for the line “return x+x;” in function_call_3.

Question B: (3 points)

What stack operations – if any – take place inside of function_call_1? You can assume that function_call_1 is not a system call.

Question C: (3 points)

What MIPS instruction(s) would you expect to see before function_call_1 in main() – given the MIPS calling convention?

Question D: (3 points)

What register would the variable x map to in the line “x = function_call_1(o,p,q)” in main()?

Question E: (3 points)

What stack operations – if any – must main() perform?