CSE2421 MIDTERM #2 Spring 2013

NAME _________________________________

SCORE ____________________/ 100

FYI: There are no compiler errors in the code presented on this exam. If you think there is, please bring it to my attention.

NOTE: If there is not an answer in the blank provided, the answer is wrong.

Warning: This test is for your eyes only. The information on this exam is NOT to be shared in any way. This exam is 100% your own work. If I see you trying to page through your exam to compare questions with your neighbor, I will move you. If I see you looking at your neighbor’s paper, I will move you. If I suspect that you are talking with your neighbor, I will move you. If I suspect that you have access to written materials during the exam, I will move you and ask you to empty your pockets, etc. The time and attention this takes away from you taking your exam is not available for makeup; that is, you get no extra time for not paying strict attention to your own paper.

NO CALCULATORS ARE ALLOWED
YOU MUST TURN IN EVERY PAGE OF THIS EXAM

Scratch area:
## Y86 Instruction Set

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>halt</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nop</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rrmovl rA, rB</td>
<td>2</td>
<td>0</td>
<td>rA</td>
<td>rB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>irmovl V, rB</td>
<td>3</td>
<td>0</td>
<td>F</td>
<td>rB</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>rrmovl rA, D(rB)</td>
<td>4</td>
<td>0</td>
<td>rA</td>
<td>rB</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>rmovl D(rB), rA</td>
<td>5</td>
<td>0</td>
<td>rA</td>
<td>rB</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>OP1 rA, rB</td>
<td>6</td>
<td>fn</td>
<td>rA</td>
<td>rB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jXX Dest</td>
<td>7</td>
<td>fn</td>
<td></td>
<td>Dest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cmovcx rA, rB</td>
<td>2</td>
<td>fn</td>
<td>rA</td>
<td>rB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>call Dest</td>
<td>8</td>
<td>0</td>
<td></td>
<td>Dest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ret</td>
<td>9</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pushl rA</td>
<td>A</td>
<td>0</td>
<td>rA</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>popl rA</td>
<td>B</td>
<td>0</td>
<td>rA</td>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Operations

- **addl**: 6 0
- **subl**: 6 1
- **andl**: 6 2
- **xorl**: 6 3

### Branches

- **jmp**: 7 0
- **jle**: 7 1
- **jl**: 7 2
- **je**: 7 3
- **ine**: 7 4
- **ige**: 7 5
- **ig**: 7 6

### Moves

- **rrmovl**: 2 0
cmovne 2 4
- **cmovle**: 2 1
cmovge 2 5
- **cmovl**: 2 2
cmovg 2 6
- **cmove**: 2 3
TRUE/FALSE (circle T or F for each statement) – 12 questions; 1 pt each

T  F  There is no difference between 0XF0F0 and 0xF0F0F0F0.
T  F  A byte is the smallest addressable unit of memory.
T  F  Y86 is a load/store architecture.
T  F  (0.011)₂ is equal to (.75)₁₀
T  F  The CPU uses machine language to perform all its operations.
T  F  The conditional move statements have an implied jump designated in the instruction based on whether the condition is true or false.
T  F  Y86 has 3 condition codes and 4 program status designations
T  F  The Y86 rmmovl command writes to memory.
T  F  The ISA serves as the boundary between software and hardware.
T  F  The “fetch” stage of the SEQ hardware structure is NOT blank for any Y86 instruction.
T  F  The number of unique binary values that can be represented in a 4-bit width are the same for all of the following different types of representations: B2U, B2T, B2O and B2S.
T  F  The Y86 assembler directive .align is followed by a number which specifies the number-byte boundary of the next addressable line in the program

MULTIPLE CHOICE (9 questions; 2 points each). Put your answer in the blank provided.

________ A hidden bit of zero for the IEEE floating point representation rules means:
   A. The stored number is zero
   B. The stored number is a denormalized (i.e. subnormal) value
   C. Neither A or B
   D. Both A and B

________ An operating system:
   A. Protects the computer from misuse
   B. Provides an abstraction for using the hardware
   C. Manages the resources to allow for reasonable use by all users
   D. All of the above

________ What is the biggest problem with storing a floating point number into its equivalent binary form (i.e. which of the following statements is true)?
   A. There is no way to represent the decimal point
   B. There isn’t a problem as long as there are enough bits in memory to store the value
   C. There isn’t always a finite and precise representation for a given floating point number
   D. None of the above

________ The Y86 “jle” instruction is followed by a label. A jump to that label occurs when each CC bit listed is set to the specified value:
   A. S=1 or Z=1
   B. S=1 and Z=1
   C. S=1 or Z=0
   D. S=1 and Z=0
When combining the single bit inputs for an add ALU operation using a half-adder, which gate is used to SUM the two bits and which gate is used to determine the carry, respectively?

A. and, xor  B. xor, and  C. and, or  D. xor, or

When multiplying by constants, for example x*8, where x is a positive integer value and the constant is designated to be in the form 2 raised to the k, an equivalent way to express this calculation (x*8) is:

A. 2 raised to the (x<<k)  B. (x<<1)-(x<<3)  C. (x<<k)  D. None of the above

Which of the following is NOT TRUE for fractional (i.e. floating point) binary notation?

A. Can only represent decimal values written in the equivalent form x*2^y  B. Lengthening the binary representation will increase accuracy  C. Can only represent decimal values that have finite-length encodings  D. None of the above

In general, a byte which has the stored hex value 0x59 can represent:

A. An instruction  B. An integer value  C. A character  D. Any of the above

The “Execute” stage for the “rrmovl” Y86 instruction is:

A. valE ← 0 + valA  B. valE ← 0 + valC  C. valE ← valB + valC  D. valE ← valB + 4

SHORT ANSWER  Put your answer in the blanks provided.

(2 pts) What is the binary negation (i.e. the answer is a binary value) of the following 2’s complement signed integer binary value: 0b10000000?

0b10000000

(8 pts; 2 pts each) Consider the number value represented (in simple base 4) as 231_4. NOTE: Use below space for scratch area.

(45) Give its decimal representation.

(0010 1101) Give the 8-bit simple binary representation of this value.

(1101 0011) Give the 8-bit two’s complement encoding of this negative number (i.e. -231_4).

(2D) Give its hexadecimal representation.
(8 pts; 2 pts each) Given an integer representation where \( w \) is the width of the data type, answer the following:

A. What is the smallest signed value? ________________________ \(-2^{w-1}\)
B. What is the smallest unsigned value? ________________________ 0
C. What is the largest signed value? ________________________ \(2^{w-1}-1\)
D. What is the largest unsigned value? ________________________ \(2^w-1\)

(10 pts; 3,1,2,2,2 each) Based on the IEEE floating point representation standard of a single precision 32-bit floating point value which has a bias of 127, answer the following questions given the binary value 1000000.00110011 (repeating 0011):

- Normalize the binary representation __________________________________________
- What value is the sign bit? ________________________
- What is the 8-bit exponent field? ________________________
- What is the 23-bit fractional field? ________________________
- What is the value of the hidden bit? ________________________

1.00000000110011 x 2 raised to the 6; 0; 1000 0101 or 0x85; 00000000110011001100110; 1;

(4 pts; 1 pt each) Match the execution representation in the first column to the Y86 instruction from the right column by putting the correct/matching letter value in the space provided.

<table>
<thead>
<tr>
<th>Execution Representation</th>
<th>Y86 Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>B ( R[rA] \leftarrow M[R[%esp]] ) then ( R[%esp] \leftarrow R[%esp]+4 )</td>
<td>A. push</td>
</tr>
<tr>
<td>D ( PC \leftarrow M[R[%esp]] ) then ( R[%esp] \leftarrow R[%esp]+4 )</td>
<td>B. pop</td>
</tr>
<tr>
<td>A ( R[%esp] \leftarrow R[%esp]-4 ) then ( M[R[%esp]] \leftarrow R[rA] )</td>
<td>C. call</td>
</tr>
<tr>
<td>C ( \text{Dest} \ R[%esp] \leftarrow R[%esp]-4 ) then ( M[R[%esp]] \leftarrow PC ) then ( PC \leftarrow \text{Dest} )</td>
<td>D. return</td>
</tr>
</tbody>
</table>

(2 pts) Give the Y86 instruction format encoding (machine code) in hex notation for the following instruction:

\[ \text{rm movl} \ %ecx, \ 24(\%ebp) \]

\[ \text{401518000000} \]

(2 pts) Decode each of the following bit representations into a Y86 instruction:

\[ 0x2174 \]

\[ \text{cmovle} \ %edi, \ %esp \]

(4 pts; 1 pt each) Determine if overflow occurs for part A and part B below by answering the two questions listed beneath each set of two binary values to be added. Circle either “Yes” or “No” for each question.

A. \[ 0101 0111 \]
\[ 1101 0011 \]
If both values are signed? Yes No
If both values are unsigned? Yes No

B. \[ 0110 0001 \]
\[ 0101 1011 \]
If both values are signed? Yes No
If both values are unsigned? Yes No
**CODE EVALUATION** (18 pts; 2 pts each). Answer the following questions about the Y86 code given below. Put your answers in the blanks provided.

How many of the instructions in the given Y86 program have the ability to change the CC bits? __________ 2

What are the values of the following registers (in hex or decimal but be sure to specify) once the program has finished executing?

%eax _______________ 0x28 or 40 (in decimal) \n%edx _______________ 0x1

Does this program change any value in memory? Yes No

Specify the offset, in hex, in the blanks provided to the left of designated program instructions.

Hex values → 0, 6, c, 12, 18(addl) , 1a, 1c, 21 (on the halt), 24 (Data)

.pos 0x0

0x0 irmovl Data, %ebx

__________ mrmovl (%ebx), %eax

0xc mrmovl 4(%ebx),%ecx

__________ irmovl $1, %edx

loop: __________

addl %eax, %eax

__________ subl %edx, %ecx

0x1c jg loop

__________ halt

.align 4

__________ Data: .long 0x5  # implement left shift operation

Data: .long 0x3  # where 5 is the value and 3 is the shift amount

**SEQ TRACING** (12 pts; 1 pt each) Put the correct value in the 3rd column for each equal sign designation. You are given the following:

The specific addl instruction is located at address 0x012
Register %edx is initialized to 0xc
Register %ebx is initialized to 0x5

<table>
<thead>
<tr>
<th>Stage</th>
<th>Generic: OP1 rA, rB</th>
<th>Specific: addl %edx, %ebx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fetch</td>
<td>icode:ifun &lt;-- M,[PC]</td>
<td>icode = 6 ifun = 0</td>
</tr>
<tr>
<td></td>
<td>rA:rB &lt;-- M,[PC+1]</td>
<td>rA = 2 rB = 3</td>
</tr>
<tr>
<td></td>
<td>valP &lt;-- PC+2</td>
<td>valP = 0x14</td>
</tr>
<tr>
<td>Decode</td>
<td>valA &lt;-- R[rA]</td>
<td>valA = 0xc</td>
</tr>
<tr>
<td></td>
<td>valB &lt;-- R[rB]</td>
<td>valB = 0x5</td>
</tr>
<tr>
<td>Execute</td>
<td>valE &lt;-- valB.OP valA</td>
<td>valE = 0x11</td>
</tr>
<tr>
<td>Memory</td>
<td>set CC</td>
<td>Z = 0 S = 0 O = 0</td>
</tr>
<tr>
<td>Write back</td>
<td>R[rB] &lt;-- valE</td>
<td>R[rB] = 0x11</td>
</tr>
<tr>
<td>PC update</td>
<td>PC &lt;-- valP</td>
<td>PC = 0x14</td>
</tr>
</tbody>
</table>