Name:

There are 13 questions, with the point values as shown below. You have 180 minutes with a total of 150 points. Pace yourself accordingly.

This exam must be individual work. You may not collaborate with your fellow students. You may use 1 sheet of notes you created, but no other external resources.

I certify that the work shown on this exam is my own work, and that I have neither given nor received improper assistance of any form in the completion of this work.

Signature:

<table>
<thead>
<tr>
<th>#</th>
<th>Question</th>
<th>Points Earned</th>
<th>Points Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vocabulary</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Binary Math</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>C Programming</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>MIPS Assembly</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Logic Gates</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>Performance</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>Datapaths</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>Caches I</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>Caches II</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>Virtual Memory</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>11</td>
<td>Branch Prediction</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>Short Answer</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>Multiple-Multiple-Choice</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Percent</td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>
Question 1: Vocabulary [10 pts]
Match each of the following definitions with the appropriate vocab word:

1. Multiple hard disks combined for performance and/or reliability. M. RAID
2. A memory technology which maintains its state as long as the power is on, but loses its contents when power is turned off. R. SRAM
3. An asynchronous notification of an external event, requiring the attention of the OS. H. Interrupt
4. The structure which holds all of the translations from virtual addresses to physical addresses K. Page Table
5. Discarding incorrect instructions from a pipeline F. Flush
6. A piece of logic which selects between two inputs, based on the value of a third input J. Mux
7. The idea that most of a program’s data accesses are likely to be contained within a small range of nearby addresses. Q. Spatial Locality
8. A class of ISAs characterized by simple instructions which are easily implemented in high-performance hardware. O. RISC
9. A type of datapath in which the CPI is always 1.0 (by definition). P. Single-cycle
10. The part of the branch predictor responsible for predicting the taken target of branches (except for returns). B. BTB
Question 2: Binary Math [10 pts]

1. Convert the number -42 to signed, 2’s complement 8-bit binary. 1101 0110

2. Write the binary number 0110 1111 0000 0101 in hexadecimal. 0x6F05

3. Write the hexadecimal representation of -13.75 as an IEEE single-precision floating point number. C15C0000

4. Add the binary numbers 0101 1110 + 0111 0010. 1101 0000

5. State whether the addition you did in part 4 overflows if the operands are treated as signed numbers. Yes

6. State whether the addition you did in part 4 overflows if the operands are treated as unsigned numbers. No
Question 3: C Programming [10 pts]

Given the following linked list node definition:

```c
struct ll_node {
    int data;
    struct ll_node * next;
};
```

Write the `reverseList` function which reverses a linked list, and returns the reversed list.

**Answer:**

```c
struct ll_node * reverseList(struct ll_node * lst) {
    struct ll_node * ans = NULL;
    while (lst != NULL) {
        struct ll_node * temp = lst->next;
        lst->next = ans;
        ans = lst;
        lst = temp;
    }
    return ans;
}
```
Question 4: MIPS Assembly [10 pts]
Translate the strUpper function (written in C below) to MIPS assembly:

void strUpper(char * s) {
    while (*s != \0) {
        *s = toUpper(*s);
        s++;
    }
}

Answer:

strUpper:
    addiu $sp, $sp, 32
    sw $fp, 0($sp)
    sw $ra, 4($sp)
    sw $s0, 8($sp)
    addiu $fp, $sp, 28
    move $s0, $a0
.L_lp:
    lbu $a0, 0($s0)
    beqz $a0, .L_done
    jal toUpper
    sw $v0, 0($s0)
    addiu $s0, $s0, 1
    b .L_lp
.L_done
    lw $fp, 0($sp)
    lw $ra, 4($sp)
    lw $s0, 8($sp)
    addiu $sp, $sp, 32
    jr $ra
Question 5: Logic Gates [10 pts]
Given the following circuit:

1. Write the boolean formula for this circuit: \((A \text{ and not } B) \text{ or } (B \text{ and } C)\)

2. Fill in the truth table for this circuit:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Question 6: Performance [20 pts]
A pipelined processor executes a 100-billion instruction program, and has the following performance related characteristics:

- 20% Loads, 10% Stores, 20% Branches, 50% ALU.
- Clock frequency: 2.0 GHz (0.5ns).
- Branch mis-prediction penalty: 5 cycles. Accuracy: 90%.
- Load-to-use penalty: 50% of loads cause a 1-cycle penalty.
- L1 instruction cache: Thit included in pipeline. %miss: 1%
- L1 data cache: Thit: included in pipeline. %miss: 5%.
- The L1 data cache has a writebuffer and a writeback buffer.
- L2 cache: Thit: 10 cycles. %miss: 5%.
- Memory: Thit: 200 cycles. %miss: 0%.

Compute the following:

1. What is the CPI penalty from load-to-use stalls? 0.1
2. What is the CPI penalty from branch mis-predictions? 0.1
3. What is the CPI penalty from data memory (D-cache misses)? 0.2
4. What is the CPI penalty from instruction memory (I-cache misses)? 0.2
5. What is the total CPI? 1.6

6. How long (in seconds) does the program take to execute? 80 seconds
7. Supposed the pipeline could be re-designed to have more stages and a faster clock. Under this new design, the clock frequency is 4.0GHz (0.25ns). The branch mis-prediction penalty increases to 10 cycles. Now, 50% of loads cause a 2-cycle load-to-use penalty.

(a) What is the new CPI penalty from load-to-use stalls? 0.2

(b) What is the new CPI penalty from branch mis-predictions? 0.2

(c) What is the new CPI penalty from data memory? 0.3

(d) What is the new CPI penalty from instruction memory? 0.3

(e) What is the new total CPI? 2.0

(f) How long (in seconds) does the program take to execute now? 50 seconds
**Question 7: Datapaths [15 pts]**
The following (multi-cycle) datapath has support for J-type absolute jumps (jal, j, etc), but not for I-type relative branches (beq, bne, etc). Recall that these relative branches take the immediate value, shift it left by 2, add it to PC+4, and then use that as the target (if taken). Add support to this datapath for such branches.

![Datapath Diagram](image)
Question 8: Caches I [10 pts]
You are designing the memory hierarchy for a new processor. The access latency of main memory is 200 cycles. You have the following choices for the L2 design:

- 2MB, 16-way set-associative. Thit = 30 cycles. %miss=1%.
- 1MB, 8-way set-associative. Thit = 20 cycles. %miss=5%.
- 512KB, 4-way set-associative. Thit = 15 cycles. %miss=10%.

and the following choices for L1 designs:

- 64KB, 8-way set-associative. Thit = 2 cycles. %miss=4%.
- 32KB, 4-way set-associative. Thit = 1 cycle. %miss=5%.

Which cache designs would you choose and why?

Answer:
The L2 design is independent of the L1 design, so we can/should select it first. For the L2, we can compute Tavg for each design:

- 2MB: 30 + 0.01 * 200 = 32
- 1MB: 20 + 0.05 * 200 = 30
- 512K: 15 + 0.10 * 200 = 35

This means the 1MB (Tavg=30) is the best L2 design. Now we can pick the L1:

- 64KB: 2 + 0.04 * 30 = 2 + 1.2 = 3.2
- 32KB: 1 + 0.05 * 30 = 1 + 1.5 = 2.5

This means 32KB (Tavg = 2.5) is the best L1 design.
Question 9: Caches II [10 pts]

Assume you have an empty 32B cache with 8B blocks. The cache is 2-way set-associative. Addresses are 8 bits. The left column below lists the addresses accessed. For each address, show how it is split into tag, index and offset in the next three columns, show the new state of the cache after the access in the next 4 columns, and state the outcome—whether its a hit or a miss—in the last column. The first row shows the initial state of the cache. The columns for the ways of each set show the tags (only) in those ways. Way 0 is MRU, Way 1 is LRU in each set at any given time. You do not need to worry about data values. All numbers are in hex, as should be all of the numbers in your answer.

<table>
<thead>
<tr>
<th>Address</th>
<th>Tag</th>
<th>Index</th>
<th>Offset</th>
<th>Set 0 Way 0</th>
<th>Set 0 Way 1</th>
<th>Set 1 Way 0</th>
<th>Set 1 Way 1</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(start)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0</td>
<td>F</td>
<td>7</td>
<td>C</td>
<td>—</td>
</tr>
<tr>
<td>F1</td>
<td>F</td>
<td>0</td>
<td>1</td>
<td>F</td>
<td>0</td>
<td>7</td>
<td>C</td>
<td>Hit</td>
</tr>
<tr>
<td>1F</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>F</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>Miss</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>F</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>Hit</td>
</tr>
<tr>
<td>C2</td>
<td>C</td>
<td>0</td>
<td>2</td>
<td>C</td>
<td>F</td>
<td>1</td>
<td>7</td>
<td>Miss</td>
</tr>
<tr>
<td>81</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>C</td>
<td>1</td>
<td>7</td>
<td>Miss</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>7</td>
<td>Miss</td>
</tr>
</tbody>
</table>
Question 10: Virtual Memory [15 pts]
Suppose that a system has a 32-bit (4GB) virtual address space. It has 1GB of physical memory, and uses 1MB pages.

1. How many virtual pages are there in the address space? 4096

2. How many physical pages are there in the address space? 1024

3. How many bits are there in the offset? 20

4. How many bits are there in the virtual page number? 12

5. How many bits are there in the physical page number? 10

6. Some entries of the page table are shown below (all values are in hex, and all entries shown are valid). Translate virtual address 0x410423 to a physical address, using the translations in this page table. 0xDD10423
<table>
<thead>
<tr>
<th>Entry Number</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1F</td>
</tr>
<tr>
<td>1</td>
<td>3C</td>
</tr>
<tr>
<td>2</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>9C</td>
</tr>
<tr>
<td>4</td>
<td>DD</td>
</tr>
<tr>
<td>5</td>
<td>EE</td>
</tr>
<tr>
<td>6</td>
<td>99</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20</td>
<td>2F</td>
</tr>
<tr>
<td>21</td>
<td>4C</td>
</tr>
<tr>
<td>22</td>
<td>65</td>
</tr>
<tr>
<td>23</td>
<td>AC</td>
</tr>
<tr>
<td>24</td>
<td>ED</td>
</tr>
<tr>
<td>25</td>
<td>FE</td>
</tr>
<tr>
<td>26</td>
<td>100</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>40</td>
<td>11F</td>
</tr>
<tr>
<td>41</td>
<td>13C</td>
</tr>
<tr>
<td>42</td>
<td>155</td>
</tr>
<tr>
<td>43</td>
<td>19C</td>
</tr>
<tr>
<td>44</td>
<td>1DD</td>
</tr>
<tr>
<td>45</td>
<td>1EE</td>
</tr>
<tr>
<td>46</td>
<td>199</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
### Question 11: Branch Prediction [10 pts]

One particular branch (i.e., one specific PC) has the following actual outcomes. Show the predictions for both a one-bit counter and a two-bit counter (no history). For the two-bit counter, use $T$ for strongly taken, $t$ for weakly taken, $n$ for weakly not-taken, and $N$ for strongly not taken. Finally fill in the accuracy (percentage of predictions that were correct) at the bottom of the table. The first prediction is done for you.

<table>
<thead>
<tr>
<th>Outcome</th>
<th>1-bit counter</th>
<th>2-bit counter</th>
<th>2-bit counter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prediction</td>
<td>Correct?</td>
<td>Prediction</td>
</tr>
<tr>
<td>T</td>
<td>N</td>
<td>no</td>
<td>n</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>yes</td>
<td>t</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>yes</td>
<td>T</td>
</tr>
<tr>
<td>N</td>
<td>T</td>
<td>no</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>N</td>
<td>no</td>
<td>t</td>
</tr>
<tr>
<td>N</td>
<td>T</td>
<td>no</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>N</td>
<td>no</td>
<td>t</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>yes</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>yes</td>
<td>T</td>
</tr>
<tr>
<td>N</td>
<td>T</td>
<td>no</td>
<td>T</td>
</tr>
</tbody>
</table>

Accuracy: 40% | 60%
Question 12: Short Answer  [10 pts]

1. Explain what a segmentation fault is. In your answer, be sure to address what type of programming errors/program actions cause it, as well as how the hardware detects the situation.

**Answer:**
A segmentation fault occurs when a program attempts to access invalid memory. The canonical example is an attempt to dereference a NULL pointer. The hardware detects this situation by finding no valid translation for the requested address, causing a page fault. The OS then determines that the requested address lies outside of the program’s address space and terminates it with a segmentation fault.

2. Compare and contrast caches and virtual memory. Give at least one similarity and one difference between the two. For the difference, explain *why* this difference exists.

**Answer:**
(Many possible)
**Similarity:** both split memory into fixed sized chunks (blocks/pages), and manipulate memory at this granularity.
**Difference:** In virtual memory, software (the OS) makes a replacement decision, while in caches, the hardware makes the replacement decision. This difference exists because Tmiss is so much larger for memory (which misses to disk) than any other level of the memory hierarchy—this justifies the extra time for software transfer control into a software routine to make a complex decision, in the hopes of reducing %miss.
Question 13: Multiple-Multiple-Choice [10 pts]
For each question, circle all that apply. If none of the selections are appropriate, then choose “e. None of the above”

1. The advantage(s) of virtual memory is/are: a,c
   a. Programmers do not need to worry about the actual memory locations holding their program.
   b. L1 and L2 cache hit rates improve.
   c. Security.
   d. The OS can be written in a less hardware-dependent manner.
   e. None of the above.

2. Which of the following will decrease conflict misses in a cache: e
   a. Increasing block size
   b. Decreasing associativity
   c. Decreasing Tmiss of the next level cache
   d. Increasing the clock frequency
   e. None of the above

3. Which of the following are commonly features of a RISC ISA: b,c
   a. Memory-to-memory operations
   b. 3 operand arithmetic operations
   c. Fixed length instruction encodings
   d. Load-compare-branch instructions
   e. None of the above

4. Which of the following are problems that can arise when pipelining: a,d
   a. Data Hazards
   b. Water Hazards
c. Dukes of Hazards
d. Control Hazards
e. None of the above

5. Which of the following is an advantage of interrupts over polling? b

a. Interrupts are a simpler mechanism for both the hardware and software.
b. Interrupts allow for more efficient CPU utilization.
c. Interrupts are compatible with virtual memory, while polling is not.
d. Polling causes bad hit rates in the L1 instruction cache.
e. None of the above