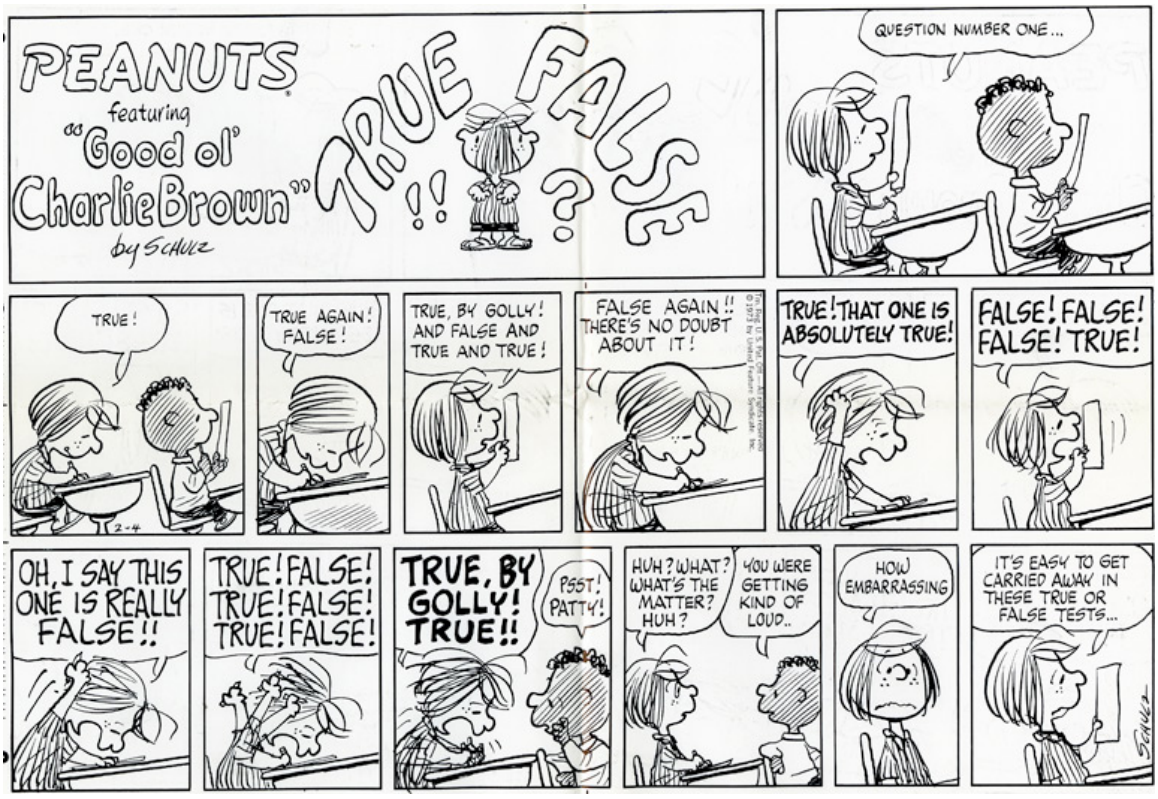
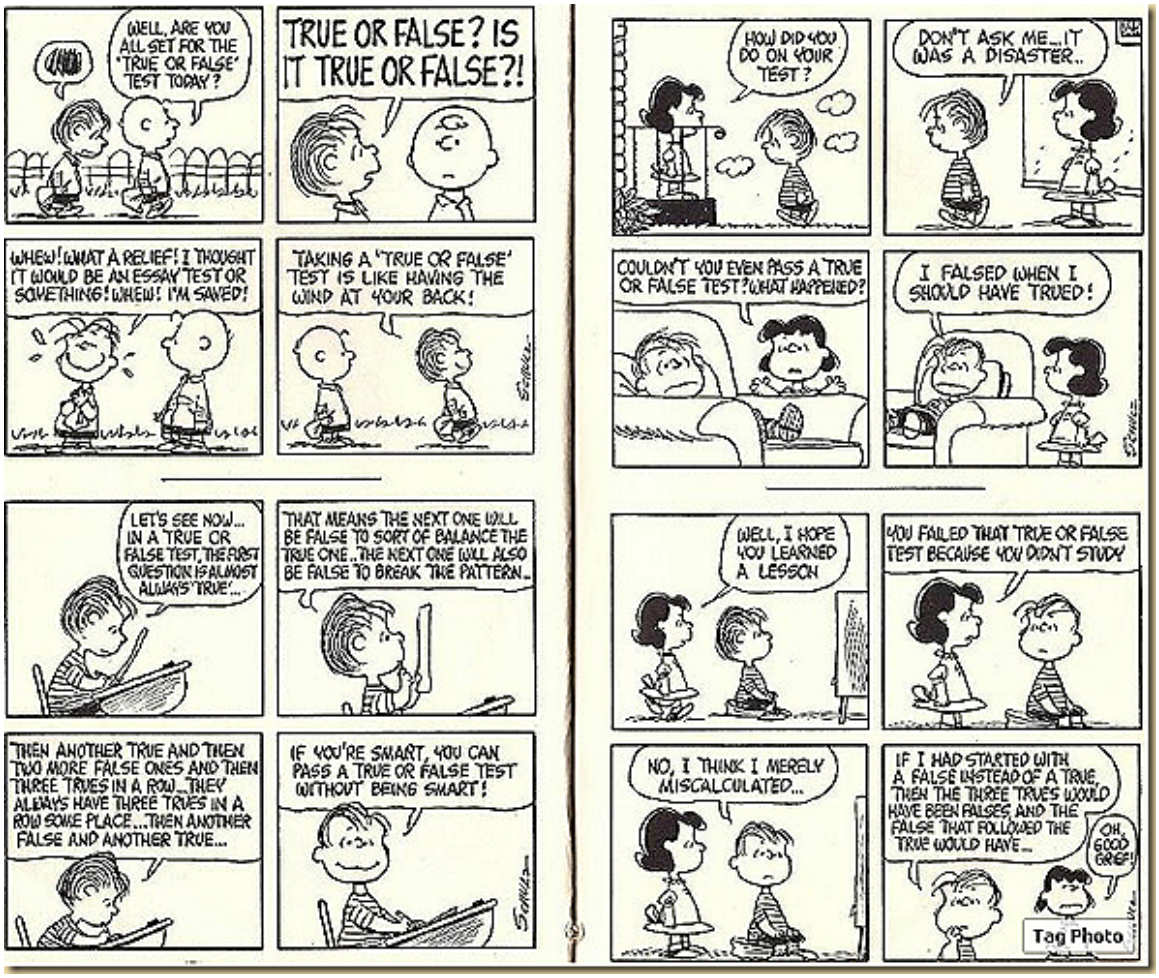
CS61C Fall 2013 Final Instructions

This exam is closed book, closed notes, open two-sided crib sheet. Please put away all phones and calculators -- you won't need them. The exam is worth **50** points and represents 20% of your course grade. It contains **50** multiple-choice questions each worth 1 point on **18** numbered pages, including this cover page. There is no penalty for incorrect answers. Mark your answers on the provided scantron forms in #2 pencil. Erase completely should you wish to change an answer. Be sure to write your name, course login, and lab section on the scantron form. For some questions, more than one answer is correct, and you must mark all of the correct choices to receive credit. The MIPS Green Card is appended at the end.




WELL, ARE YOU ALL SET FOR THE "TRUE OR FALSE" TEST TODAY?
TRUE OR FALSE? IS IT TRUE OR FALSE?!
WHEW! WHAT A RELIEF! I THOUGHT IT WOULD BE AN ESSAY TEST OR SOMETHING! WHEW! I'M SAVED!
TAKING A "TRUE OR FALSE" TEST IS LIKE HAVING THE WIND AT YOUR BACK!
LET'S SEE NOW... IN A TRUE OR FALSE TEST, THE FIRST QUESTION IS ALMOST ALWAYS "TRUE"...
THAT MEANS THE NEXT ONE WILL BE FALSE TO SORT OF BALANCE THE TRUE ONE..THE NEXT ONE WILL ALSO BE FALSE TO BREAK THE PATTERN..
THEN ANOTHER TRUE AND THEN TWO MORE FALSE ONES AND THEN THREE TRUES IN A ROW..THEY ALWAYS HAVE THREE TRUES IN A ROW SOME PLACE...THEN ANOTHER FALSE AND ANOTHER TRUE...
IF YOU'RE SMART, YOU CAN PASS A TRUE OR FALSE TEST WITHOUT BEING SMART!
HOW DID YOU DO ON YOUR TEST?
DON'T ASK ME...IT WAS A DISASTER..
COULDN'T YOU EVEN PASS A TRUE OR FALSE TEST? WHAT HAPPENED?
I FALSED WHEN I SHOULD HAVE TRUED!
WELL, I HOPE YOU LEARNED A LESSON
YOU FAILED THAT TRUE OR FALSE TEST BECAUSE YOU DIDN'T STUDY
NO, I THINK I MERELY MISCALCULATED...
IF I HAD STARTED WITH A FALSE INSTEAD OF A TRUE, THEN THE THREE TRUES WOULD HAVE BEEN FALSES, AND THE FALSE THAT FOLLOWED THE TRUE WOULD HAVE...
OH, GOOD GRIEF!
Tag Photo

Section I: *Cache Aware Programming*
For each of the following common operations, identify the ***best*** caching strategy (i.e., one that maximizes the hit rate). Assume that all caches are the same size:

1. Iterating over a linked list multiple times. Assume that the entire linked list is smaller than the size of the cache.
   (a) Direct mapped cache with blocksize n
   (b) Fully Associative cache with blocksize n and LRU (least recently used) replacement
   (c) Fully Associative cache with blocksize n and MRU (most recently used) replacement
   (d) Both a and c produce equivalent hit rates
   (e) Both a and b produce equivalent hit rates

For Questions 2-4, given the following naïve and three mystery code snippets, match each mystery snippet to a performance improving measure listed in those questions:

```
// NAIVE
… // N and M defined elsewhere, ARR is of size 3*M
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
         ARR[j*2] = ARR[j*2] * ARR[j*2];
    }
}
```

```
// MYSTERY 1
… // N and M defined elsewhere, ARR is of size 3*M
for (int i = 0; i < M; i++) {
    for (int j = 0; j < N; j++) {
         ARR[i*2] = ARR[i*2] * ARR[i*2];
    }
}
```

```
// MYSTERY 2
… // N and M defined elsewhere, ARR is of size 3*M
for (int i = 0; i < M; i++) {
    int x = i * 2;
    for (int j = 0; j < N; j++) {
         ARR[x] = ARR[x] * ARR[x];
    }
}
```

```
// MYSTERY 3
… // N and M defined elsewhere, ARR is of size 3*M
for (int i = 0; i < N; i++) {
    int x = j*2;
    for (int j = 0; j < M; j++) {
         ARR[x] = ARR[x] * ARR[x];
    }
}
```

2. Which code snippet runs correctly and takes advantage of *register blocking*?
   (a) MYSTERY 1
   (b) MYSTERY 2
   (c) MYSTERY 3
   (d) MYSTERY 2 and MYSTERY 3
   (e) None of the above

3. Which code snippet runs correctly and takes advantage of *loop reordering*?
   (a) MYSTERY 1
   (b) MYSTERY 2
   (c) MYSTERY 3
   (d) MYSTERY 1 and MYSTERY 2
   (e) MYSTERY 2 and MYSTERY 3

4. Which code snippet runs correctly and takes advantage of *cache blocking*?
   (a) MYSTERY 1
   (b) MYSTERY 2
   (c) MYSTERY 3
   (d) MYSTERY 2 and MYSTERY 3
   (e) None of the above

5. You're now a CS61C TA. Given the following output from running the benchmark for Project 3, which <u>one</u> of the following should you tell your student to implement next?
   100x200: 1000 Gflop/s
   200x300: 1000 Gflop/s
   300x400: 1000 Gflop/s
   400x500: 500 Gflop/s
   500x600: 400 Gflop/s
   600x700: 210 Gflop/s

   (a) Register blocking
   (b) Cache blocking
   (c) Loop reordering
   (d) SIMD
   (e) None of the above

<u>Section II:</u> *SIMD Programming*

6. We have the following incomplete implementation of a dot product:

```
void dotp(float *c,float *a,float *b,const int length) {
   for (int i=0; i<length; i+=4) {
   _mm_storeu_ps(c+i,_____________________);
   }
}
```

   Which of the following code snippets would correctly complete this method so that it would work for any arrays with length divisible by 8?
   (a) _mm_mul_ps(_mm_loadu_ps(*(a+i)),_mm_loadu_ps(*(b+i)))
   (b) _mm_loadu_ps(a+i) * _mm_loadu_ps(b+i)
   (c) _mm_loadu_ps(*(a+i)) * _mm_loadu_ps(*(b+i))
   (d) _mm_mul_ps(_mm_loadu_ps(a+i),_mm_loadu_ps(b+i))
   (e) _mm_mul_ps(_mm_load1_ps(a[i]),_mm_load1_ps(b[i]))

7. Suppose we vectorize the loop `for (int n=0; n<F_LENGTH; n++) f[n] *= 2;` with __m512 types, and also perform a single loop unrolling by 4 iterations. To handle the edge cases, we need loops with various other types that are not unrolled. If all of the following potential edge case loops were run as many times as possible without doing redundant work, how many times would each run?

I. Looping with __m512 types
II. Looping with __m256 types
III. Looping with __m128 types
IV. Looping with float types

(a) I: 63, II: 15, III: 7, IV: 3
(b) I: 2, II: 2, III: 2, IV: 4
(c) I: 3, II: 0, III: 0, IV: 15
(d) I: 0, II: 0, III: 0, IV: 63
(e) I: 3, II: 1, III: 1, IV: 3

Section III: *Synchronization and Multiprocessor Cache Consistency*

Assume a four processor shared memory multiprocessor. The shared memory is 16 words. Each processor's cache is organized as direct mapped with four blocks, each of four words. The processor implements a simple invalidate protocol. The protocol works as follows: cache block states can be *invalid*, *shared* for reading, or *modified* after writing. Shared copies of a memory word in other caches are invalidated on memory writes; a modified cache block must first be written back to memory in response to reads or writes to other caches. The original copy becomes shared if a read or invalid if a write.

Consider the following sequence of memory accesses, increasing in time and starting at T0. An access is written as Read/Write<***word address, value***>. In the chart below, P0 at time T0 issues a Read to memory location 8 and reads the value 1 from that memory address. Give the BEST answer. To break ties on concurrent write, higher processor numbers write is committed at the end of the time period.

Time	P0	P1	P2	P3
T0	**R<8, 1>**			
T1		R<0, 2>		
T2			R<0, 2>	
T3				W<12,3>
T4	W<8, 4>			
T5		W<0, 5>	W<0, 6>	
T6				R<8, 4>
T7		W<8, 7>		
T8			W<8, 8>	

8. What value is in the first word of P1's cache block 0 immediately after time T3?
(a) Invalid
(b) 1
(c) 2
(d) 3
(e) Insufficient information to answer

9. What value is in the first word of P1's cache block 0 immediately after time T5?
(a) Invalid
(b) 1
(c) 2
(d) 3
(e) Insufficient information to answer

10. What value is in the first word of P2's cache block 0 immediately after time T7?
    (a) Invalid
    (b) 5
    (c) 6
    (d) 7
    (e) Insufficient information to answer

11. What value is in the first word of P3's cache block 0 immediately after time T8?
    (a) Invalid
    (b) 3
    (c) 4
    (d) 8
    (e) Insufficient information to answer

Section IV: *OpenMP Programming*

We are using OpenMP to parallelize array operations, and we would like to compare the implementations with the serial version (i.e., if there were no pragma declarations). Correctness is based on comparing the output to the serial version. Assume NUM_THREADS > 1. For each pair of questions, first determine which statement from *Column 1* best describes the code execution, and then pick the main problem with the code from *Column 2*.

Column 1

A. Always incorrect
B. Sometimes incorrect
C. Always correct, faster than serial
D. Always correct, speed relative to serial depends on cache parameters
E. Always correct, slower than serial

Column 2

A. False sharing
B. Data race on a shared variable
C. Error with scope of variable
D. Other problem
E. No problem with code

Questions 12-14:

```
// Counts the number of elements in array divisible by three
// We know that the result is always greater than zero
int count_threes(int *array, int size) {
  int count = 0;
  #pragma omp parallel for
  for (int i = 0; i < size; i++) {
     if( ! (array[i] % 3) ) {
             count++;
     }
  }
  return count;
}
```

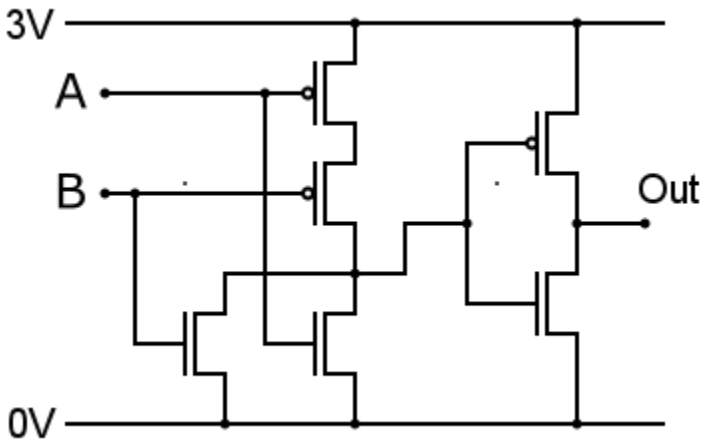
12. Consider the `count_threes()` function. How would the code execute? Choose from *Column 1*.

13. What is the main problem with the code described in the above question? Choose from *Column 2*.

14. Suppose we take the count_threes() function and add private(count) to the pragma statement. How would the code execute? Choose from *Column 1*.

Section V: *Transistors-to-Gates*

15. A and B are the inputs of this gate, while out is the output. High is 3V and low is 0V. What is this gate?



(a) AND gate
(b) OR gate
(c) NAND gate
(d) NOR gate
(e) XOR gate

16. How does the power consumption of a transistor change if its capacitance is doubled, the voltage is quadrupled, and the switch frequency is halved? Select the expression that relates the new power P_{new} to the old power P_{old}.
(a) $P_{new} = 0.5 * P_{old}$
(b) $P_{new} = P_{old}$
(c) $P_{new} = 4 * P_{old}$
(d) $P_{new} = 8 * P_{old}$
(e) $P_{new} = 16 * P_{old}$

Section VI: *Truth Tables to Boolean Algebra*

For the given expression, select the choice to which it is equivalent. Each question will have only one correct answer. Note that $\oplus$ is a 2-input XOR gate, and $\odot$ a 2-input XNOR gate.

17. $(X + Y \cdot \overline{Z}) \cdot (Z + \overline{X} \cdot Z + Y) \cdot \overline{Z \cdot Y}$

(a) $(X + Y \cdot \overline{Z}) \cdot \overline{X \cdot Y \cdot Z}$

(b) $X \cdot (Z + \overline{X} \cdot Z) \cdot \overline{Z}$

(c) $X \cdot (Z \oplus Y)$

(d) $Y \cdot \overline{Z} + X \cdot Z \cdot \overline{Y}$

(e) $(X \cdot Z) \odot (Y \cdot (X + \overline{Z}))$

18. $\overline{X \cdot \overline{Y}} \cdot (Y \oplus \overline{X \cdot Y})$

a. $\overline{X} \cdot Y$

b. $X \cdot Y + (Y \oplus X)$

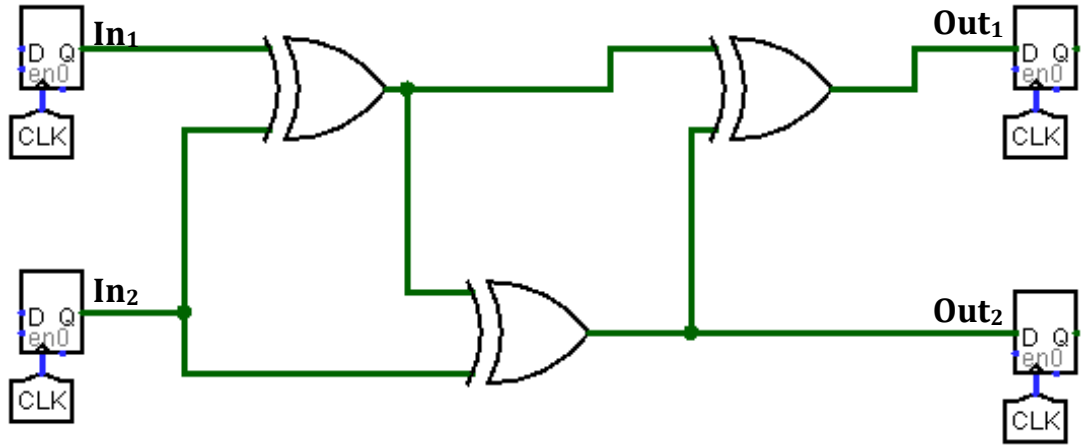
c. $\overline{X} + Y$

d. $X \oplus Y$

e. $\overline{X} \odot \overline{Y}$

Section VII: *Synchronous Logic and Timing Diagrams*

Examine the circuit below:



Determine the simplest possible Boolean expressions for

19. Out_1:

(a) In_1 (b) $In_1 * In_2$ (c) $In_1 * Out_2$ (d) $In_1 + Out_2$ (e) None of the above

20. Out_2:

(a) In_1 (b) $In_1 * In_2$ (c) $In_1 * Out_1$ (d) $In_1 + Out_1$ (e) None of the above

Assume the following for Problems 21-26:

- Negligible setup time
- 5 ns clk-to-q delay
- 10 ns XOR propagation delay
- Hold time may be larger than clk-to-q

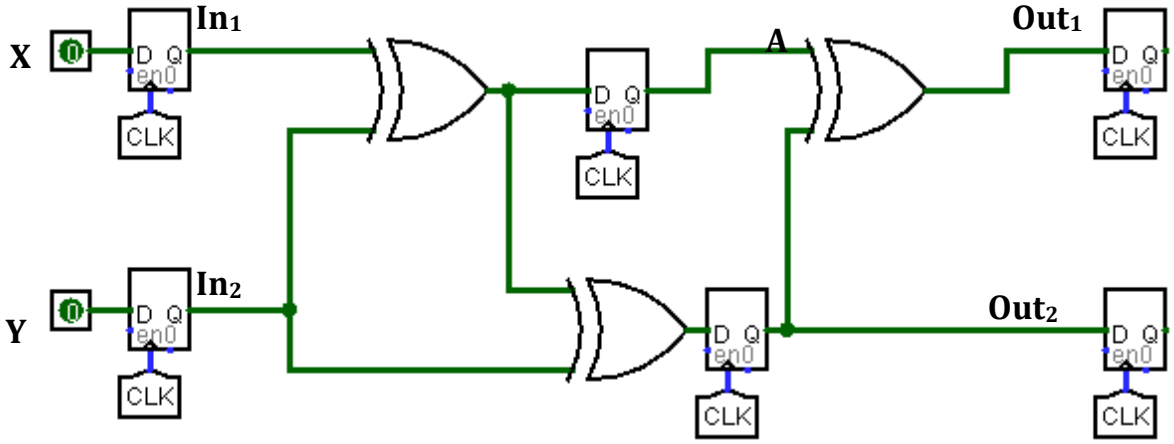
21. What is the maximum hold time we can have for this circuit to satisfy register timing constraints?

(a) 5 ns (b) 10 ns (c) 15 ns (d) 20 ns (e) 30 ns

22. What is the maximum clock rate of this circuit?

(a) 1/(35 ns) (b) 1/(30 ns) (c) 1/(20 ns) (d) 1/(15 ns) (e) 1/(10 ns)

Now, examine the pipelined version of the circuit:



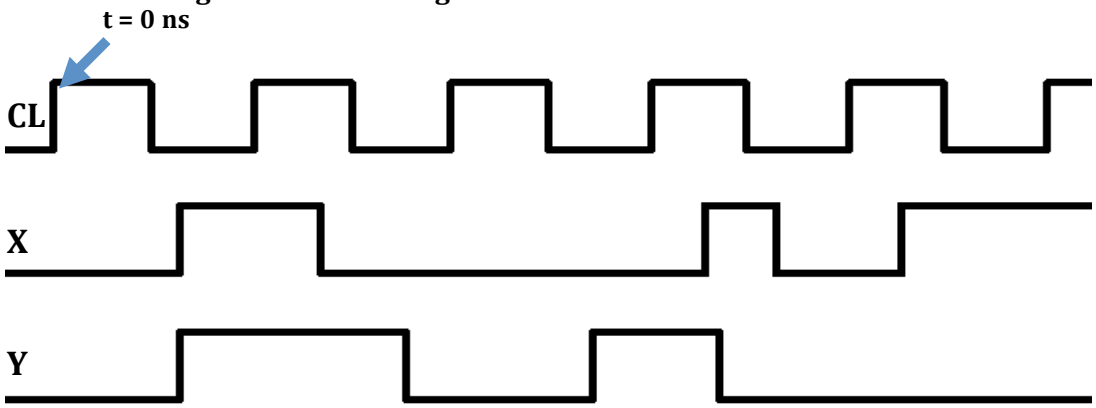
23. Assuming a clock period of 50 ns, what is the maximum theoretical setup time we can have for this circuit to satisfy register-timing constraints and function the same as the non-pipelined version? Assume that X and Y do not ever cause any register timing violations.

(a) 0 ns (b) 5 ns (c) 10 ns (d) 25 ns (e) 40 ns

24. Assuming a clock period of 50 ns, what is the maximum theoretical hold time we can have for this circuit to satisfy register-timing constraints and function the same as the non-pipelined version? Assume that X and Y do not ever cause any register timing violations.

(a) 5 ns (b) 10 ns (c) 25 ns (d) 30 ns (e) 40 ns

You may fill out the following waveform diagram to assist in answering the following questions. Assume a clock period of 50 ns, and all values except X and Y start unknown. **Your waveform diagram will NOT be graded.**



In_1

In_2

You can add more signals here

Out_2

25. At what time does the value at A first change to 1? (unknown to 1 counts as a change)
    (a) 55 ns
    (b) 105 ns
    (c) 155 ns
    (d) 205 ns
    (e) None of the above

26. At what time does the value at Out_2 first change from 1 to 0?
    (a) 55 ns
    (b) 105 ns
    (c) 155 ns
    (d) 205 ns
    (e) None of the above

Section VIII: *State Machines*

For this FSM, we will be implementing a Vending Machine that only takes in Dimes and Nickels. This machine only takes in coins and will not give you back change if you go over,

instead it goes into a miniature "bank" for the next person to be able to use. You are only ever able to either insert nothing, one dime at a time, one nickel at a time, or both a dime and a nickel at the same time. A drink will cost 15 cents, and it will output a drink as soon as you reach pass the 15 cent threshold.

27. What is the minimum number of bits needed to represent the input?
    (a) 0 (b) 1 (c) 2 (d) 3 (e) 4

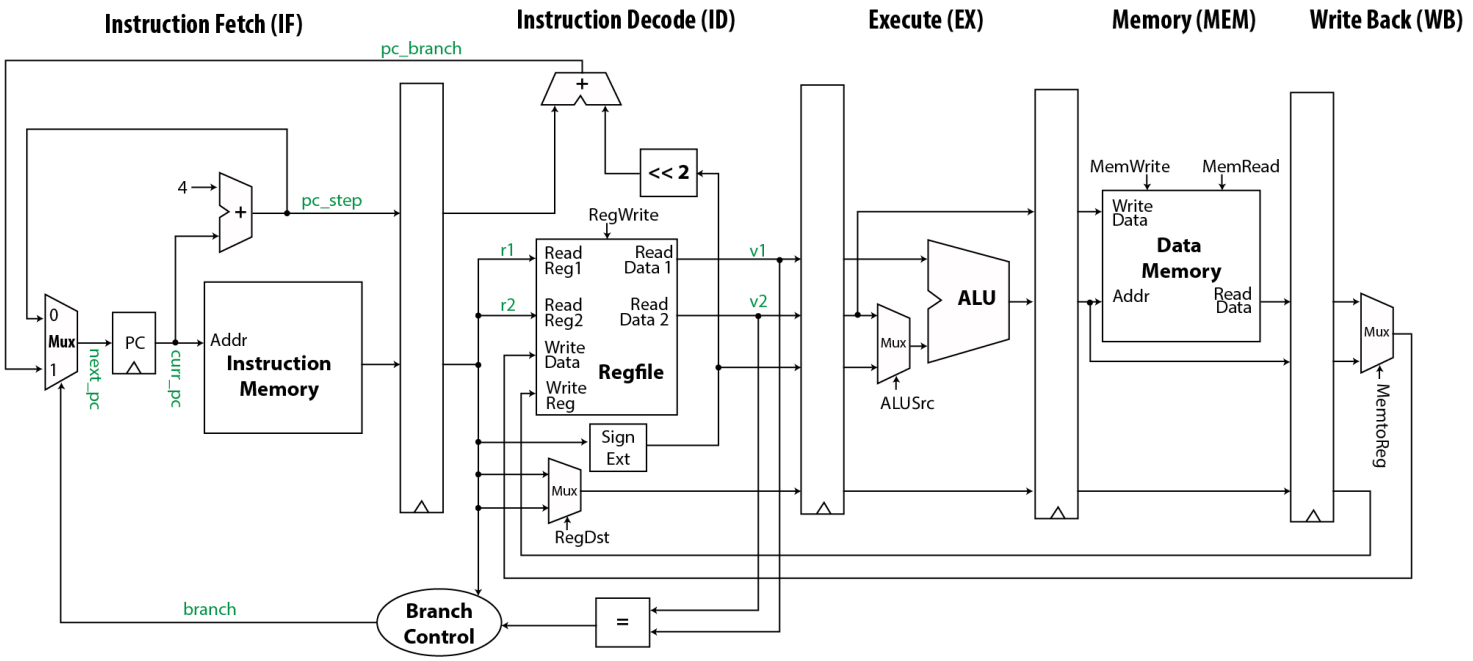
28. What is the minimum number of states needed to represent this?
    (a) 2 (b) 3 (c) 4 (d) 5 (e) 6

29. From how many of the states would you be able to get a drink from?
    (a) 1 (b) 2 (c) 3 (d) 4 (e) 5

Section IX: *Datapath Design*

Normally it is a pretty bad idea to put a `beq` inside of another `beq`'s delay slot. But we're curious so we will investigate the implications of what happens if we do!

We will be running it on the following standard 5-stage pipelined datapath:



Note the following:

- Branch comparison happens in the ID stage
- It has a hazard detection unit capable of correctly performing stalls, which is not shown in the drawing.
- Control signals like RegDst, MemtoReg, etc are also not fully shown, but assume that they are functioning correctly.

- **At the beginning of the first cycle, all the registers are 0, and the PC starts at 0x00.**

Suppose the code we have loaded into instruction memory is as follows:

Address	Encoded Instruction				Written Instruction
	opcode	rs	rt	immediate	
0x00	001000	00000	00010	0x2	`addi $2, $0, 2`
0x04	000100	00000	00000	0x4	`beq $0, $0 FOO`
0x08	000100	00000	00000	0x6	`beq $0, $0 BAR`
0x0c	001000	00000	00011	0x3	`addi $3, $0, 3`
0x10	001000	00000	00100	0x4	`addi $4, $0, 4`
0x14	001000	00000	00101	0x5	`FOO: addi $5, $0, 5`
0x18	001000	00000	00110	0x6	`addi $6, $0, 6`
0x1c	001000	00000	00111	0x7	`addi $7, $0, 7`
0x20	001000	00000	01000	0x8	`BAR: addi $8, $0, 8`
0x24	001000	00000	01001	0x9	`addi $9, $0, 9`

30. As drawn, does this datapath support forwarding for data hazards?
    (a) Yes, register values can be forwarded from MEM
    (b) Yes, register values can be forwarded from EX
    (c) Yes, register values can be forwarded from WB
    (d) No, this datapath does not support forwarding
    (e) Maybe, depending on the details of the regfile

31. Which instruction is in each stage of the pipeline during the 3rd clock cycle?
    (a) IF: addi, ID: beq, EX: beq, M: X, WB: X
    (b) IF: beq, ID: beq, EX: addi, M: X, WB: X
    (c) IF: addi, ID: beq, EX: addi, M: X, WB: X
    (d) IF: X, ID: beq, EX: addi, M: X, WB: X
    (e) IF: beq, ID: beq, EX: beq, M: X, WB: X

32. At the end of the 2nd clock cycle (just before the beginning of the 3rd cycle), what are the values on the wires labeled? (Choose the column with the correct values)

	A	B	C	D	E
curr_pc	0x08	0x08	0x08	0x04	0x04
pc_step	0x0C	0x0C	0x0C	0x08	0x08
r1	00000	00000	00000	00010	00000
r2	00000	00000	00000	00000	00010
v1	00000	00000	00000	00000	00000
v2	00000	00000	00000	00000	00000
branch	1	0	1	1	0
pc_branch	0x14	0x14	0x18	0x06	0x0C
next_pc	0x14	0x0C	0x18	0x0C	0x08

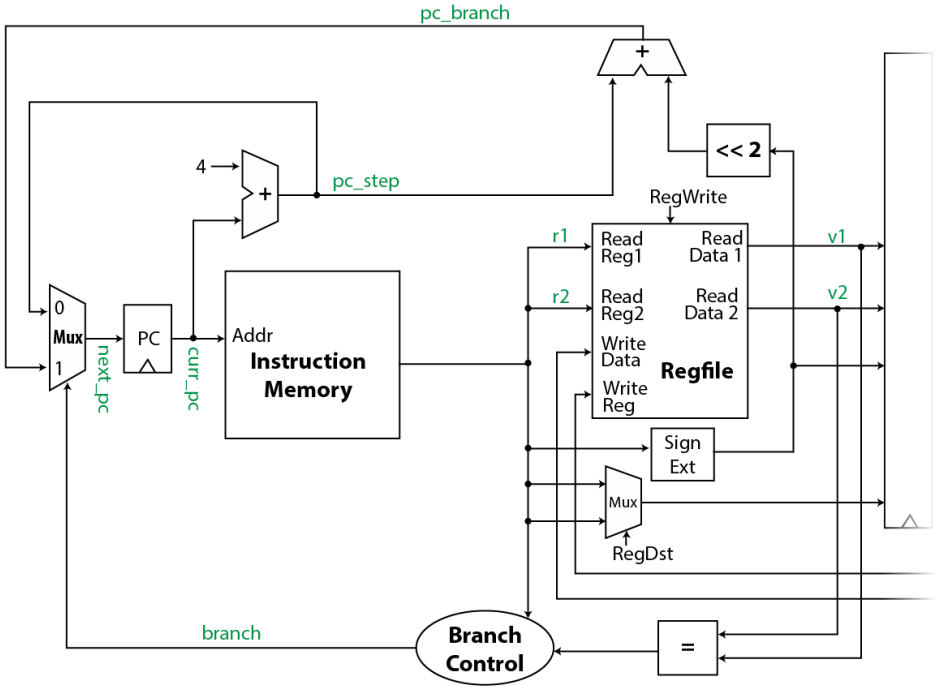
33. What about at the end of the 3rd clock cycle (just before the beginning of the 4th cycle)? (Choose from the above table).
    (a) Column A
    (b) Column B
    (c) Column C
    (d) Column D
    (e) Column E

34. What is the fourth instruction executed?
    (a) **`BAR: addi $8, $0, 8`**
    (b) **`addi $3, $0, 3`**
    (c) **`addi $4, $0, 4`**
    (d) **`FOO: addi $5, $0, 5`**
    (e) **`nop`**

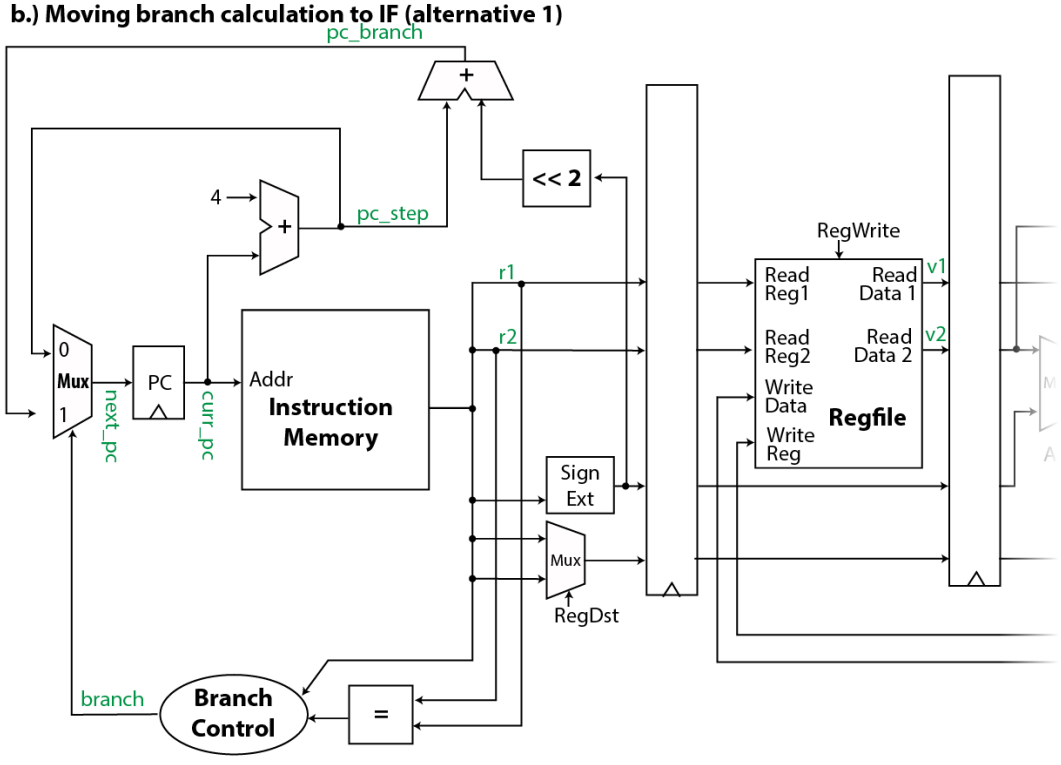
35. Continue to think about the signals on each of the wires in the fifth and sixth cycles. What values are in the registers after **all** of the code executes?
    (a) `$2 = 2, $3 = 3, $4 = 4, $5 = 5, $6 = 6, $7 = 7, $8 = 8, $9 = 9`
    (b) `$2 = 2, $3 = 0, $4 = 0, $5 = 5, $6 = 6, $7 = 7, $8 = 8, $9 = 9`
    (c) `$2 = 2, $3 = 0, $4 = 0, $5 = 5, $6 = 0, $7 = 0, $8 = 8, $9 = 9`
    (d) `$2 = 2, $3 = 0, $4 = 0, $5 = 5, $6 = 0, $7 = 0, $8 = 0, $9 = 0`
    (e) `$2 = 2, $3 = 0, $4 = 0, $5 = 0, $6 = 0, $7 = 7, $8 = 8, $9 = 0`

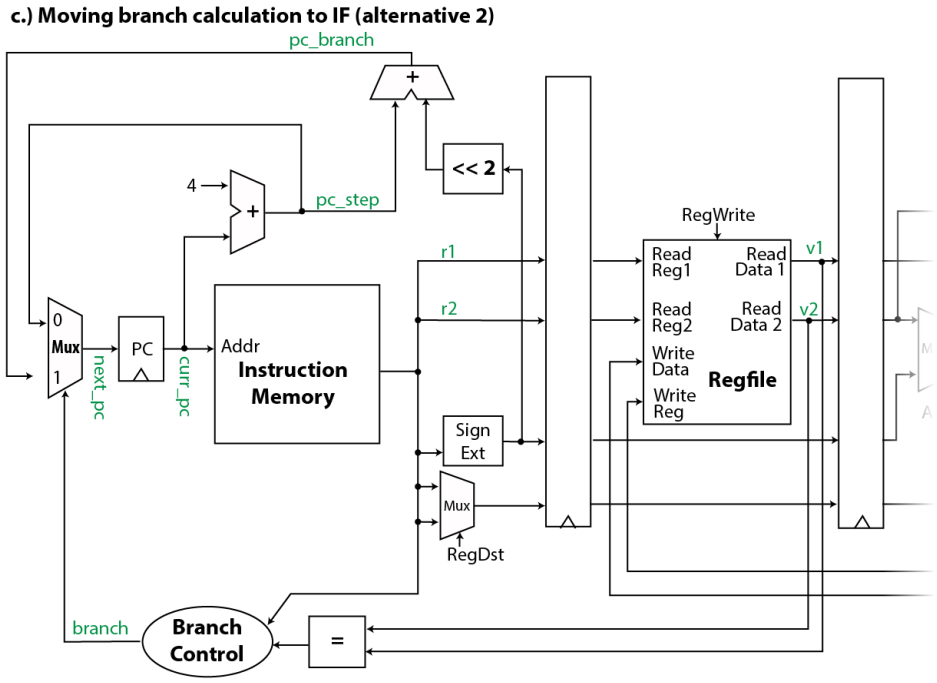
36. Which of the following modifications of the datapath would result in a properly functioning processor that does **not** require a branch delay slot?

a.) Removing a pipeline stage by combining IF and ID



b.) Moving branch calculation to IF (alternative 1)





Section XI: *Instruction Level Parallelism and Pipelining*

37. What is an advantage of increasing the number of pipelines?
    (a) Less complex circuit
    (b) Faster computation on a whole instruction
    (c) Faster clock speed
    (d) Increased clock period
    (e) More efficient bypassing

38. A static issue processor has two ALU instruction units, one load/store unit, one jump unit, and one branch unit. Select the false statement.
    (a) This processor has redundant ALU, therefore it will be less efficient than a processor with just one ALU instruction unit.
    (b) This processor can have IPC less than four.
    (c) This processor will in most cases not make full use of its pipeline.
    (d) This processor will benefit from loop unrolling.
    (e) This processor can have IPC greater than four.

39. Assume you have a dynamic issue processor with one load/store unit and one ALU unit. Instructions are issued in the order of the answers listed below. What would be the second instruction after *re-ordering*? If an ALU instruction and Load/Store instruction occurs at the same time, assume that ALU one will be written first.
    (a) `addi $sp, $sp, -4`
    (b) `sw $s0, 0($sp)`
    (c) `addi $t0, $a0, $a1`
    (d) `lw $t1, 0($t0)`
    (e) `add $v0, $t1, $a2`

40. You have a processor that has two ALU/branch units and two load/store unit with 120 internal registers. You have a loop that is written as follows:

```
LOOP: addi $t1, $a0, $t0
      lw $t2, 0($t1)
      addi $t3, $a1, $t0
      sw $t2 0($t3)
      addi $t0, $t0, -1
      bne $t0, $0, LOOP
```

    What is the max number of times this loop can be unrolled? Choose the closest answer.
    (a) 0
    (b) 30
    (c) 40
    (d) 60
    (e) 120

Section XII: *Architecture Potpourri*

41. Which of the following statement about Virtual Memory is false?
    (a) All memory blocks within the virtual memory will be contiguous blocks in physical memory.
    (b) Virtual memory allows you to use more memory than in Main Memory.
    (c) The biggest price to pay for Virtual Memory is address translation on each memory reference.
    (d) Virtual Memory is only used within User Mode Processes.
    (e) The VPN field can be larger than the PPN field.

42. If we want to construct a PTE (Page Table Entry) where there are flags for Writable, Valid, and Dirty. And we have a total of 1 TB space in Main Memory. Each page also has the size of 8 KBs, what is the minimum number of bits we need to fill up the PTE?
    (a) 28 Bits
    (b) 30 Bits
    (c) 32 Bits
    (d) 33 Bits
    (e) 34 Bits

43. You have two caches:
    - Direct mapped (DM) cache: 2 tag bits, 1 index bit, 1 offset bit
    - 2-way set associative (SA) cache: 3 tag bits, 0 index bits, 1 offset bit.

    Assume MRU (most recently used) replacement policy. Calculate the *miss rate* on the following sequence of accesses: 0, 1, 2, 5, 3.

    (a) **DM**: 0.6 **SA**: 0.8
    (b) **DM**: 1 **SA**: 0.6
    (c) **DM**: 0.4 **SA**: 0.4
    (d) **DM**: 0.6 **SA**: 1
    (e) **DM**: 1 **SA**: 1

44. You have a cache with 8B blocks, and a total size of 64B. However, you forgot the associativity! Given this sequence of accesses on word-addressed memory (4B), what is the associativity?

    0 (MISS), 1 (HIT), 2 (MISS), 15 (MISS), 17 (MISS), 0 (HIT), 32 (MISS), 1 (MISS)

    (a) Direct-mapped
    (b) 2-way
    (c) 4-way
    (d) Fully associative (8-way)
    (e) Cannot be determined from the sequence above

Match the definition on the right that best matches the term on the left:

45. Reliability
46. Service Interruption
47. Availability

(a) MTTF/(MTTF + MTTR)
(b) MTTF + MTTR
(c) MTTR
(d) MTTF
(e) None of the above

48. Given the following 7 bit code string encoded using the Hamming code technique: 1000111_{two}. This code word may or may not have a single bit in error. Numbering the bit positions 1 through 7, from left to right, which statement is true?

    (a) Position 1 is in error: the correct code word is 0000111_{two}
    (b) Position 3 is in error: the correct code word is 1010111_{two}
    (c) Position 5 is in error: the correct code word is 1000011_{two}
    (d) Position 7 is in error: the correct code word is 1000110_{two}
    (e) The code word is correct

49. Consider how to rank the RAID levels using different storage system figures of merit. Using the symbols < (less), = (equal), and > (more), if I tell you that:

    RAID 1 > RAID 2 = RAID 3 < RAID 4 < RAID 5

    (i.e., RAID 1 is more than RAID 2, RAID 2 is equal to RAID 3, RAID 3 is less than RAID 4, and RAID 4 is less than RAID 5), which storage system metric must I be using? Assume the same number of physical disks in each RAID configuration. If **multiple** answers apply, fill them **ALL** in.
    (a) Total storage user storage capacity for a fixed number of disks
    (b) Total amount of data redundancy (redundancy bytes)
    (c) Maximum I/O bytes per second (I/O bandwidth)
    (d) Logical Read Rate (read I/O operations per second)
    (e) Logical Write Rate (write I/O operations per second)

50. In the Big Bang Theory's Rock-Paper-Scissors-Lizard-Spock game, only one of the following statements is false. Which one is it?
    (a) Rock breaks Scissors
    (b) Scissors cut Paper
    (c) Paper disproves Spock
    (d) Spock bites Lizard
    (e) Lizard poisons Spock