CS 250 Spring 2017 Homework 05
Due 11:58pm Wednesday, February 22, 2017
Submit your typewritten file in PDF format to Blackboard.

1. Why are general purpose registers important?
   General purpose registers are important because they allow the storage of data the program is using, such as the result of an ADD function. These registers make it possible for any kind of program to be run on the processor.
2. Refer to textbook Figure 5.9.
   a. What part of the ISA specifies the type of operands?
      The Instruction name, E.G. add *unsigned*
   b. Name an instruction that shows a register can hold a binary string that is interpreted as a (likely) 2's complement integer.
      add
   c. Name an instruction that shows a register can also hold a meaningless binary string.
      store word
   d. Name an instruction that overrides the work of the circuit in Figure 6.4.
      jump
   e. Name the instruction that corresponds to the C language code *if(a == b){ }*.
      branch equal
   f. How many bits are necessary for the opcode field for the instructions in Figure 5.9?
      32 instructions or 2^5 so 5 bits
   g. Using Section 5.22 of the text and Figure 5.11 as guide, write the single assembly language instruction that implements the test in the C language conditional expression code *(a == 0? b = 4 : b = 5).*
      cmp r1, r0 #letting r1 be the register holding a, r0 is always 0

3. To obtain the highest performance in an instruction execution pipeline what operating condition must be maintained?
   The fetch-execute cycle must constantly be maintained or the processor will waste time waiting for something to do.

4. Modify Figure 6.2 to support a re-design of the processor of Chapter 6 to support 32 general purpose registers. What significant negative impact does this change have on the expressiveness of the ISA?

	operation	reg A	reg B	dst reg	Offset
add	00001	5-bit field	Unused, 5-bit field	5-bit field	Unused 12-bit field
load	00010	5-bit field	Unused, 5-bit field	5-bit field	Offset 12-bit field
store	00011	5-bit field	5-bit field	Unused, 5-bit field	Offset 12-bit field
jump	00100	5-bit field	Unused, 5-bit field	Unused, 5-bit field	Offset 12-bit field

The offset field loses 3bits, or 28,672 positions, which allows a loaded program to jump or store less distances.

5. What is the cardinality of the set of bit strings that are equivalent within the context of Figure 6.3?
   $2^1 + 2^{15} = 32,770$

   add r4, r2, r3 and add r2, r4, r3 are all the equivalent operations since A+B = B+A. The operation field and the dest reg field are fixed since changing them would result in a different instruction. But the offset field, of size 15-bits, is used and may be changed in any way with no effect to the instruction.

6. Imagine the new instruction SUB, meaning subtract, has been added to the ISA of Figure 6.2. The assembly language instruction SUB R1, R2, R3 is defined as R1 ← R2 – R3 where R2 is the minuend (the operand being decreased) and R3 is the subtrahend (the amount of the decrease). Write a descriptor in the format shown in Figure 6.2 for all bit strings that mean SUB R1, R2, R3. For any field in the binary representation of this assembly language instruction that is not a unique bit string, write a simple specification of all acceptable bit strings.

	operation	reg A	reg B	dst reg	unused
sub	00101	4-bit field	4-bit field	4-bit field	15-bit field

7. Is Figure 6.9 of a Von Neumann architecture? Why or why not?
   It does not consist of a Von Neumann architecture because it has separate memory for instruction data and application data. The Von Neumann architecture stores instruction data with application data.

8. Make a table showing for each instruction in Figure 6.2 which input (upper or lower (closer to the figure caption) for each the multiplexers M1, M2, and M3 must be selected. If either multiplexer input is a correct selection, then enter X for don't care in that cell of the table.

Operation	M1	M2	M3
add	Select	X	Select
load	Select	Select	X
store	Select	Select	Select
jump	Select	Select	X

9. How many gates were required for the "bit bucket" in Lab 04?

   1
   A JK flip flop has a max value value of 1, so un-rembering a value means the flip flop should output 0. Tying J to low, the K pin can be set with an AND gate from Q and an un-remember switch.