1. (5 points) MIPS. What is in the specified register at the end of each sequence of code? Give your answer in decimal integers or hex.

```
.data
  x:       .word   0xf8
  y:       .word   0x8ca47

(a) Register $t0
  la $s1, x
  lb $t0, 3($s1)
  ANSWER: -8 or 0xffffffff

(b) Register $t1
  la $s2, y
  lh $t1, -2($s2)
  ANSWER: 248 or 0xf8

(c) Register $t2
  lw $s5, x
  li $t2, 28
  and $t2, $t2, $s5
  ANSWER: 24 or 0x18

(d) Register $t3
  lw $s7, y
  srl $t3, $s7, 5
  sll $t3, $t3, 19
  srl $t3, $t3, 14
  ANSWER: 51,776 or 0xca40

(e) Register $t4
  lw $s4, x
  or $t4, $s4, 170
  ANSWER: 250 or 0xfa
```
2. (10 points) MIPS.

(a) What is the machine language instruction (in hex) for
\text{lw} \quad $7,10($6)
\text{ANSWER: 8cc7000a}

(b) What is the assembly language instruction for the following machine language instruction, given in hex:
\text{018a2820}
\text{ANSWER: add \$5,$12,$10}

3. (5 points) Number Representation. What is -43 in 8bit, 2’s complement arithmetic?
\text{ANSWER: 1101 01001}

4. (5 points) Number Representation. How would 16.875 be represented in IEEE Floating Point Standard format?
\text{ANSWER: 0 10000011 000011100000000000000000}

5. (5 points) Number Representation. What is a floating point underflow?
\text{ANSWER: An underflow occurs when the magnitude of a negative exponent is too large to be represented in 8 bits.}

6. (10 points) ALU. In designing the carry-lookahead adder, we defined
\[ P_1 = p_7 \cdot p_6 \cdot p_5 \cdot p_4 \]
What does this value represent? How is its computation used to speed up the addition?
\text{ANSWER: P1 represents the conditions for a “propagate” through the second set of 4, 1-bit adders. Because we compute it in parallel with setting the values of each of those adders, it is available earlier to determine the carry into the third set of adders (adders for bits 8 through 11) which can then begin adding earlier.}

7. (10 points) Single cycle machine. Explain the purpose of the RegDst control line in the single cycle design. Why is it needed? When is it set to 1 (that is, for which instructions)?
\text{ANSWER: The RegDst line controls the register that is to be written by an instruction. It is set to 1 for R-type instructions because the destination register is in bits 15-11 (for law instructions, it is set to 0 because the destination register is in bits 20-16).}

8. (10 points) Multicycle machine. The AluSrcB line in the multicycle design has four possible values. explain the effect of each value and describe which instructions use that setting.
\text{ANSWER: The ALUSrcB line controls the second input to the ALU. Settings:
- 00 selects the second register read from the register file, used in 3rd cycle of beq and r-type
- 01 selects the constant 4, used by all instructions in the 1st (instruction fetch) cycle
- 10 selects the sign extended bottom half of the instruction, used in sw and}
lw address calculation
• 11 selects the sign-extended, shifted address portion of the instruction, used in 2nd cycle by all instructions but only needed by beq

9. (5 points) Multicycle machine. In the multicycle machine design we discussed, we introduced a register, ALUOut which was not present in our Single Cycle Design. Explain why it is needed.

ANSWER: The ALUOut register is used to hold the value computed in the ALU as input for the next cycle. It wasn’t needed in the single cycle design because all ALU outputs were used immediately, on the same cycle in which they were computed.

10. (12 points) Performance. Compare our single cycle design S with our multicycle design M and our pipelined design P for a program with the following instruction mix. Assume that 50% of the lw instructions are immediately followed by a use of that value and that 70% of branches are taken. Which is the fastest machine? How much faster is it than each of the other two?

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>lw</td>
<td>25%</td>
</tr>
<tr>
<td>sw</td>
<td>10%</td>
</tr>
<tr>
<td>beq</td>
<td>30%</td>
</tr>
<tr>
<td>R-type</td>
<td>35%</td>
</tr>
</tbody>
</table>

ANSWER:
CPI for S is 1, TimeS = I * 1 * 8ns
CPI for M: (.25*5 + .10*4 + .30*3 + .35*4) = 3.95, TimeM = I * 3.95 * 2ns
CPI for P: (.25*(.50*1 + .5*2) + .10*1 + .30*(.70*2 + .30*1) + .35*1) = 1.335, TimeP = I*1.335*2ns
So, P is the fastest. It is 8/(1.335*2) = 2.99 times faster than S and (3.95*2)/(1.335*2) =2.95 times faster than P.

11. (8 points) Performance. A program executes in 5.22 seconds with a CPI of 1.45 on a machine with a cycle rate of 400MHz. At what MIPS is it running?
ANSWER: MIPS is millions of instructions per second.
400 * 10^6 cycles/second
-------------------------------- = 276*10^6 instructions/second = 276 MIPS
1.45 cycles/ instruction

12. (10 points) Pipelining. Shade the diagram appropriately and identify each of the data hazard(s) in the following code sequence. For each, indicate how it will be handled (ie, stall, forwarding, etc.) in our final machine design.

ANSWER:
add  $s3,$s3,$s2
lw  $s2,10($s3) hazard with use of $s3 from add, handled by forwarding
sub $s6,$s3,$s2 hazard with use of $s3 from add, handled by forwarding
          hazard with use of $s2 from lw, needs a stall
or  $s6,$s2,$s7 hazard with use of $s2 from lw, eliminated by stall above