1. (20 pts) All parts carry equal credit.
   a. Briefly describe Moore’s Law.

   b. What is the difference between signed and unsigned instructions? What are the hardware implications?

   c. Distinguish between dynamic power dissipation and static power dissipation.
d. Describe an example of data parallel computation (in terms of instructions and data). What would be an example application that might be well suited to data parallel computation?
2. (45 pts) Consider the following block of SPIM code. The text segment starts at 
0x00400000 and the data segment starts at 0x10010000.

.data
label: .word 8,16,32,64
L1: .byte 64, 32

.text
.globl main
main: la $a0, label    #pass array base address
     li $a1, 16       #pass array count
     jal func        #function call

     li $v0, 10      #terminate program
     syscall

func: move $v0, $a0     #get array base address
     move $v1, $a1     # get array count
     add $v1, $v1, $v0  #array ending address
     move $t1, $0     #initialize loop sum

loop: lw $s6, 0($v0)      #load first value
     add $t1, $t1, $s6    #update sum
     addi $v0, $v0, 4     #update pointer
     slt $t0, $v0, $v1     #check for termination
     bne $t0, $0, loop   #onto next element
     move $v0, $t1        #pass sum back to main
     jr $ra

a. (10 pts) What are the values of main, func, and loop? Clearly state which 
instructions, if any, have to be translated into more than one native 
instruction.

i. main
   ___________________

ii. func
    _________________

iii. loop
     _______________
b. (5 pts) The second time through the loop, what are the values in registers $ra$, and $t0$.
   i. $ra$ __________
   ii. $t0$ __________

c. (10 pts) If this is a Big Endian machine what value is returned in register $t0$ for the following instruction.
   i. lw $t0, L1($0) ___

d. (10 pts) The following is a block of encoded SPIM code. Decode the sequence to produce the original MIPS instructions.
   0043402a _______________
   1500fffc _______________

e. (10 pts) Imagine you have doubled the number of general-purpose registers in the processor but still need to encode instructions in 32-bits. What is the impact on the encoding of the `beq` and `bne` instructions in terms of the range of the target addresses?
3. **(20 pts)** Consider the function `func` from the SPIM program shown in question 2.
   a. **(10 pts)** Show modifications just to the function `func`, if any, to conform to the MIPS procedure call convention. If none, state why.

b. **(10 pts)** Assume that `func` was independently compiled assuming a starting instruction address of 0x00000000. It is actually loaded into memory starting at location 0x00402000.
   i. Which instructions in `func`, if any, have to be corrected? Explain.

   ii. What would the symbol table contain after assembly is complete?
4. (15 pts) Consider the single cycle SPIM datapath shown overleaf.
   a. (10 pts) Specify the values of the control signals for each of the following instructions. Assume all registers are initialized to 0x44.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>ALUSrc</th>
<th>MemRead</th>
<th>RegDst</th>
<th>RegWrite</th>
<th>ALU Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>addi $8, $9, -4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lw $t2, 8($at)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   b. (5 pts) Suppose that due to device wear out, the RegDst signal becomes permanently stuck at the value 1. What would be the impact of the execution of the addi instruction in part (a).