EECS 361 - HW 2

1. Translate the following loop into C. Assume that the C-level integer `i` is held in register `$t1`, `$s2` holds the C-level integer called `result`, and `$s0` holds the base address of the integer `MemArray`.

```
addi $t1, $0, $0
LOOP: lw $s1, 0($s0)
add $s2, $s2, $s1
addi $s0, $s0, 4
addi $t1, $t1, 1
slti $t2, $t1, 150
bne $t2, $0, LOOP
```

2. Translate function `f` into MIPS assembly language. If you need to use registers `$t0` through `$t7`, use the lower-numbered registers first. Assume the function declaration for `func` is "`int func(int a, int b);`".

The code for function `f` is as follows:

```
int f(int a, int b, int c, int d){
return func(func(a,b),c-d);
}
```

3. Calculate the time necessary to perform a multiply using the approach described in the textbook (31 adders stacked vertically) if an integer is 8 bits wide and an adder takes 5 time units.

4. As discussed in the textbook, one possible performance enhancement is to do a shift and add instead of an actual multiplication. Since 9×6, for example, can be written $(2\times2\times2+1)\times6$, we can calculate 9×6 by shifting 6 to the left 3 times and then adding 6 to that result. Show the best way to calculate 0x23 $\times$ 0x75 using shifts and adds/subtracts. Assume both inputs are 8-bit unsigned integers.

5. The basic single-cycle MIPS implementation in Figure 4.2 can only implement some instructions. New instructions can be added to an existing Instruction Set Architecture (ISA), but the decision whether or not to do that depends, among other things, on the cost and complexity the proposed addition introduces into the processor datapath and control. The first three problems in this exercise refer to the new instruction:
Instruction: `LWI Rt,Rd(Rs)`
Interpretation: `Reg[Rt] = Mem[Reg[Rd]+Reg[Rs]]`
5a. Which existing blocks (if any) can be used for this instruction?
5b. Which new functional blocks (if any) do we need for this instruction?
5c. What new signals do we need (if any) from the control unit to support this instruction?

6. When processor designers consider a possible improvement to the processor datapath, the decision usually depends on the cost/performance trade-off. In the following three problems, assume that we are starting with a datapath from Figure 4.2, where I-Mem, Add, Mux, ALU, Regs, D-Mem, and Control blocks have latencies of 400 ps, 250 ps, 30 ps, 120 ps, 200 ps, 350 ps, and 90 ps, respectively, and costs of 1000, 20, 10, 100, 200, 2000, and 400, respectively. Consider the addition of a multiplier to the ALU. This addition will add 400 ps to the latency of the ALU and will add a cost of 500 to the ALU. The result will be 8% fewer instructions executed since we will no longer need to emulate the MUL instruction.
6a.What is the clock cycle time with and without this improvement?
6b.What is the speedup achieved by adding this improvement?
6c.Compare the cost/performance ratio with and without this improvement.