

Homework 1

Problem 1

We have introduced HCL in class to design combinatorial logic circuit. Try to write HCL code for the following circuit: Take A, B, C and D as input, output the secondary biggest value of them.

E.g. Median (1, 3, 4, 2) = 3.

Problem 2~5

The following four problems are based on the following code snippet:

```
# int Sum(int *Start, int Count)
0x042: a05f          Sum:   pushl %ebp
0x044: 2045          rrmovl %esp,%ebp
0x046: 501508000000   mrmovl 8(%ebp),%ecx      # ecx = Start
0x04c: 50250c000000   mrmovl 12(%ebp),%edx     # edx = Count
0x052: 6300          xorl %eax,%eax          # sum = 0
0x054: 6222          andl %edx,%edx           # Set condition codes
0x056: 7378000000     je End
0x05b: 506100000000   Loop: mrmovl (%ecx),%esi   # get *Start
0x061: 6060          addl %esi,%eax           # add to sum
0x063: 30f304000000   irmovl $4,%ebx
0x069: 6031          addl %ebx,%ecx           # Start++
0x06b: 30f3ffffff    irmovl $-1,%ebx
0x071: 6032          addl %ebx,%edx          # Count--
0x073: 745b000000     jne Loop                # Stop when 0
0x078: 2054          End:   rrmovl %ebp,%esp
0x07a: b05f          popl %ebp
0x07c: 90          ret
```

Note: Here are some values of the registers you maybe use before the corresponding instructions. The value stored in **%esp** is **128**, value in **%eax** is **1**, value in **%edx** is **2**.

Fill in the right-hand column of the following table, specialize the function of each stage corresponding to the generic form.

Problem 2

Stage	Generic	Specific
	<code>rrmovl rA, rB</code>	<code>rrmovl %esp, %ebp</code>
Fetch	$\text{icode} : \text{ifun} \leftarrow M1[\text{PC}]$ $\text{rA:rB} \leftarrow M1[\text{PC}+1]$ $\text{valP} \leftarrow \text{PC}+2$	
Decode	$\text{valA} \leftarrow R[\text{rA}]$	
Execute	$\text{valE} \leftarrow 0 + \text{valA}$	
Memory		
Write Back	$R[\text{rB}] \leftarrow \text{valE}$	
Update PC	$\text{PC} \leftarrow \text{valP}$	

Problem 3

Stage	Generic	Specific
	<code>OpI rA, rB</code>	<code>xorl %eax, %eax</code>
Fetch	$\text{icode} : \text{ifun} \leftarrow M1[\text{PC}]$ $\text{rA:rB} \leftarrow M1[\text{PC}+1]$ $\text{valP} \leftarrow \text{PC}+2$	
Decode	$\text{valA} \leftarrow R[\text{rA}]$ $\text{valB} \leftarrow R[\text{rB}]$	
Execute	$\text{valE} \leftarrow \text{valB OP valA}$ Set CC	
Memory		
Write Back	$R[\text{rB}] \leftarrow \text{valE}$	
Update PC	$\text{PC} \leftarrow \text{valP}$	

Problem 4

Stage	Generic	Specific
	<code>irmovl V, rB</code>	<code>irmovl \$-1, %ebx</code>
Fetch	$\text{icode} : \text{ifun} \leftarrow M1[\text{PC}]$ $\text{rA:rB} \leftarrow M1[\text{PC}+1]$ $\text{valC} \leftarrow M4[\text{PC} + 2]$ $\text{valP} \leftarrow \text{PC}+6$	
Decode		
Execute	$\text{valE} \leftarrow 0 + \text{valC}$	
Memory		
Write Back	$R[\text{rB}] \leftarrow \text{valE}$	
Update PC	$\text{PC} \leftarrow \text{valP}$	

Problem 5

Stage	Generic	Specific
	OP1 rA, rB	addl %ebx,%edx
Fetch	icode : ifun $\leftarrow$ M1[PC] rA:rB $\leftarrow$ M1[PC+1] valP $\leftarrow$ PC+2	
Decode	valA $\leftarrow$ R[rA] valB $\leftarrow$ R[rB]	
Execute	valE $\leftarrow$ valB OP valA Set CC	
Memory		
Write Back	R[rB] $\leftarrow$ valE	
Update PC	PC $\leftarrow$ valP	

That's the end of Homework 1. You may find it easy to finish it. Anyway, enjoy the first weekend of this semester o(∩_∩)o