Homework 8

Problem 1

Please write down the HCL code for the following signals in the SEQ implementation.

1. need_valC
2. aluB
3. mem_write
4. stat

Problem 2-4

The following problems are based on the following code snippet:

```
0x000:                      min:    #min(int *a, int count)
0x000: a05f                   pushl %ebp
0x002: 2045                   rrmovl %esp, %ebp
0x004: 501508000000           mrmovl 8(%ebp), %ecx      #ecx=a
0x00a: 50250c000000           mrmovl 12(%ebp), %edx     #edx=count
0x010: 30f0ffffff7f           irmovl $2147483647, %eax  #init
0x016: 703c000000             jmp end
0x01b:                      loop:
0x01b: 506100000000           mrmovl (%ecx), %esi       #a[i]
0x021: 2063                   rrmovl %esi, %ebx         #copy
0x023: 6106                   subl %eax, %esi           #compare
0x025: 752c000000             jge  sub
0x02a: 2030                   rrmovl %ebx, %eax         #min=a[i]
0x02c:                      sub:
0x02c: 30f304000000           irmovl $4, %ebx
0x032: 6031                   addl %ebx, %ecx           #next one
0x034: 30f3ffffffff           irmovl $-1, %ebx
0x03a: 6032                   addl %ebx, %edx           #count--
0x03c:                      end:
0x03c: 6222                   andl %edx, %edx
0x03e: 741b000000             jne loop
0x043: 2054                   rrmovl %ebp, %esp
0x045: b05f                   popl %ebp
0x047: 90                     ret
```

Please fill the blanks below.

Problem 2

Stage	Generic	Specific
	`mrmovl D(rB), rA`	`mrmovl 8(%ebp), %ecx`

Fetch	`icode:ifun ← M1[PC]`
	`rA:rB←M1[PC+1]`
	`valC←M4[PC+2]`
	`valP←PC+6`	

Decode	`valB←R[rB]`	
Execute	`valE←valB+valC`	
Memory	`valM←M4[valE]`	
Write Back	`R[rA]←valM`	
Update PC	`PC←valP`	

Important Data:

%ebp	0x1000
0x1000	0x1030
0x1008	0x400
0x100c	0xa

Problem 3

Stage	Generic	Specific
	`OPl rA, rB`	`andl %edx, %edx`

Fetch	`icode:ifun ← M1[PC]`
	`rA:rB←M1[PC+1]`
	`valP←PC+2`	

Decode	`valA←R[rA]` `valB←R[rB]`	
Execute	`valE←valB OP valA` `Set CC`	
Memory		
Write Back	`R[rB]←valE`	
Update PC	`PC←valP`	

Important Data:

%edx	0

Problem 4

Stage	Generic	Specific
	popl rA	popl %ebp
Fetch	icode:ifun←M1[PC] rA:rB←M1[PC+1] valP←PC+2	
Decode	valA←R[%esp] valB←R[%esp]	
Execute	valE←valB+4	
Memory	valM←M4[valA]	
Write Back	R[%esp]←valE R[rA]←valM	
Update PC	PC←valP	

Important Data:

%esp	0x1000
0x1000	0x1030
0x1008	0x400
0x100c	0xa