ICS Homework 2

March 13, 2020

1 Organization

1.1 Y86-64 Instructions

Please write down the byte codes of the following Y86-64 instructions.

Y86-64 instructions	**Byte codes (hex value)**
`rrmovq %rbx, %rdx`	0x2032
`jmp 0xabc`	0x70bc0a000000000000
`addq %rbx, %rax`	0x6030
`call 0x1234`	0x803412000000000000
`rmmovl %rcx, 0x12(%rbx)`	0x40131200000000000000
`jle 0x280`	0x718002000000000000
`pushq %rax`	0xa00f

1.2 SEQ Processor

Suppose we are going to implement **crmmovl rA, D(rB)**, which conditionally write rA to memory, in our SEQ Y86_64 processor.

1. How long the `crmmovl` instruction is?

10 bytes.

3. Fill the table below.

Stage	`crmmovl rA, D(rB)`

Fetch	`icode:ifun <- M1[PC]`
	`rA:rB <- M1[PC + 1]`
	`valC <- M8[PC + 2]`
	`valP <- PC + 10`

Decode	`valA <- R[rA]` `valB <- R[rB]`
Execute	`Cnd <- Cond(CC,ifun)` `valE <- valB + valC`
Memory	`Cnd ?  M8[valE] <- valA : -`
Write back	
PC update	`PC <- valP`

1.3 HCL

The register signal *srcB* indicates which register should be read to generate the signal *valB*. The desired value is shown as the second step in the decode stage in Book Figures 4.18 to 4.21. Write HCL code for *srcB*.

```
word srcB = [
icode in { IOPQ, IRMMOVQ, IMRMOVQ } : rB;
icode in { IPUSHQ, IPOPQ, ICALL, IRET } : RRSP;
1 : RNONE; # Do not need register
];
```

2 System Software

2.1 Concurrency

In table below, control flow for a series of processes is shown. A cell with * means the process is executed at current time. Among these processes, which pairs run concurrently and which pairs are sequential? (Suppose all processes finish executing at the end of time.)

Time	A	B	C	D
0	*			
1			*	
2				*
3	*			
4			*	
5		*		
6	*			

SOLUTION:
Concurrent: A&B A&C A&D C&D
Sequantial: B&C B&D