孟澍 3210101819

5.15

R1 = x3121.

R2 = x4566.

R3 = xABCD.

R4 = xABCD.

5.37

PC -> MAR -> MEMORY -> MDR -> IR -> SEXT[8:0] -> ADDR1MUX -> ADDR2MUX -> ADDER -> MARMUX -> MAR -> MEMORY -> MDR -> MAR -> MEMORY -> MDR -> LOGIC & REGFILE

5.39

PC -> MAR -> MEMORY -> MDR -> IR -> SEXT[8:0] -> ADDR1MUX -> ADDR2MUX -> ADDER -> MARMUX -> REGFILE

5.50

BR: PC.

LEA: R7.

LD: MAR.

6.9

```
0011 0000 0000 0000    ; x3000
0010 000 000000100     ; R0 <- m[x3006]
0010 001 000000100     ; R1 <- m[x3007]
1111 0000 0010 0001    ; TRAP x21
```

```
0001 001 001 1 11111   ; R1 <- R1 - 1
0000 001 111111101     ; BRp #-3
0000 0000 0101 1010    ; ASCII code for ’Z’
0000 0000 0110 0100    ; #100
```

6.10

Only c could be used for NOP. In the excution phase of a, the ALU is used to add 0 to R1, and the CC is setted, so it does something. B will skip the instruction behind it in the memory so the program cannot work correctly. C just do nothing. Therefore, only c could be used for NOP.

Only the ADD instruction uses the ALU and set CC.