Nhat Doan\_ CSC137\_ HW4 Chapter 7\_ Chris Grove

**1>For each entry in the F1 table (Table 7-1), translate it into**

**the actual signal names. Include the two multiplexor (name the 2x1**

**mux muxA, and the other one muxB). (1 pt/entry)**

|  |  |  |  |
| --- | --- | --- | --- |
| F1 | Microop. | Symbol | Actual Signals |
| 001 | AC ← AC +DR | ADD | LD(AC), ADD |
| 010 | AC ← O | CLRAC | CLR(AC) |
| 011 | AC ← AC+1 | INCAC | INC(AC) |
| 100 | AC ← DR | DRTAC | LD(AC), DRTAC |
| 101 | AR← DR(0-10) | DRTAR | LD(AR),muxA=1 |
| 110 | AC← PC | PCTAR | LC(AR), muxA=0 |
| 111 | M[AR] ← DR | WRITE | write |

**7>Using the mapping procedure of Fig. 7-3, map out the microcode**

**addresses for these operation code: 0010 1011 1111 (1 pt each)**

* 0010 → 0001000 = 8
* 1011 → 0101100 = 44
* 1111 → 0111100 = 60

**11>With Table 7-1, translate these to control signals (F1, F2,**

**and F3 by number and symbol): (1 pt each)**

Microoperation F1 F2 F3 Symbols

a. AC<-AC+1, DR<-DR+1 011 110 000 INCAC, INCDR

b. PC<-PC+1, DR<-M[AR] 000 100 101 INCPC, READ

c. DR<-AC, AC<-DR 100 101 000 ACTDR, DRTAC

**12>With Table 7-1, convert these microop. symbols to**

**microoperation and binary equivalents: (1 pt each)**

Microop. Symbols Microoperaton F1 F2 F3

a. READ, INCPC DR<-M[AR], PC<-PC+1 000 100 101

b. ACTDR, DRTAC DR<-AC, AC<-DR 100 101 000

c. ARTPC, DRTAC, WRITE PC <-AR, AC<-DR, M[AR] <- DR 111 101 110

**15>With the following microcode (Sec. 7-3):**

**Addr Microcode**

**60 010 000 010 00 00 1000011**

**61 111 100 000 01 01 1000000**

**62 001 001 000 10 10 0111111**

**63 101 110 000 11 11 0111100**

**a. Translate to symbols (Table 7-2). (1 pt each)**

60: CLRAC , COM U JMP INDRCT

61: WRITE , READ I CALL FETCH

62: ADD , SUB S RET 63(NEXT)

62: DRTAC , INCDR Z MAP 60

**b. List things logically wrong in the code. (1 pt each)**

60: cannot increment and complement AC at the same time. Due to JMP to INDRCT, control doesn’t return to 61

61: cannot write and read at the same time. The CALL behaves as a JMP because there is not a return from FETCH

62: cannot add and subtract at the same time. RET will be executed ignoring S

63: MAP is executed regardless of Z or 60

**16>Add the following new commands to Table 7-2. (First 3.) (2 pts each)**

**Command Opcode Microoperation Description**

**AND 0100 AC<-AC^M[AR] bitwise AND**

**SUB 0101 AC<-AC-M[AR] subtraction**

**ADM 0110 M[AR]<-M[AR]+AC add to memory**

AND:

|  |  |  |  |
| --- | --- | --- | --- |
| ORG 16 |  |  |  |
| NOP | I | CALL | INDRCT |
| READ | U | JMP | NEXT |
| ADD | U | JMP | FETCH |

SUB

|  |  |  |  |
| --- | --- | --- | --- |
| ORG 20 |  |  |  |
| NOP | I | CALL | INDRCT |
| READ | U | JMP | NEXT |
| SUB | U | JMP | FETCH |

ADM:

|  |  |  |  |
| --- | --- | --- | --- |
| ORG 24 |  |  |  |
| NOP | I | CALL | INDRCT |
| READ | U | JMP | NEXT |
| DRTAC, ACTDR | U | JMP | NEXT |
| ADD | U | JMP | EXCHANGE+2 |

**17>Write the microop. symbols for the CPU instruction ISZ of**

**Chapter 5 (Table 5-4). Note DR=0 status condition is not**

**available in the CD field in Sec 7-3. However, we can**

**exchange AC and DR check if AC=0. (3 pts)**

ISZ:

|  |  |  |  |
| --- | --- | --- | --- |
| NOP | I | CALL | INDRCT |
| READ | U | JMP | NEXT |
| INCDR | U | JMP | NEXT |
| DRTAC, ACTDR | U | JMP | NEXT |
| DRTAC, ACTDR | Z | JMP | ZERO |
| WRITE | U | JMP | FETCH |

ZERO:

|  |  |  |  |
| --- | --- | --- | --- |
| WRITE, INPC | U | JMP | FETCH |

**18>Write the microop. symbols for the BSA instruction like**

**the previous problem. (2 pts)**

BSA:

|  |  |  |  |
| --- | --- | --- | --- |
| NOP | I | CALL | INDRCT |
| PCTDR,ARTPC | U | JMP | NEXT |
| WRITE,INCPC | U | JMP | FETCH |