Spring 2014 Name: Mark F.

Computer Organization Test 2

Question 1. (6 points) Select the best answer to the following true-or-false questions:	Circle the correct answer	
a. The MAR holds the data value written to main memory.	True	(False)
b. The control unit causes the CPU to perform the Fetch-Decode-Execute instruction cycle.	(True)	False
c. During the Fetch cycle, a high-level language instruction (e.g., FOR, IF, WHILE, etc.) is read into the IR register.	True	(False)
d. The IR register in MARIE can be used to hold arbitrary data.	True	(False)
e. The main memory holds both instructions and data values of the program being executed.	(True)	False
f. One MARIE assembly language instruction generally translates to many machine language instruction.	True	(False)

Question 2. (19 points) Translate the following high-level language code segment to MARIE assembly language. Use the variable labels indicated in the code.

```
POS_COUNT = 0
NEG_COUNT = 0
INPUT X
WHILE X != 0 DO
   IF X > 0 THEN
      POS_COUNT = POS_COUNT + 1
   ELSE
      NEG_COUNT = NEG_COUNT + 1
   END IF
   INPUT X
END WHILE
```

```
        CLEAR
        STORE   POS_COUNT
        STORE   NEG_COUNT
        INPUT
        STORE X
WHILE,  SKIPCOND 400
        JUMP  DO
        JUMP  END_WHILE
DO,     SKIPCOND 800
        JUMP  ELSE
        LOAD  POS_COUNT
        ADD   ONE
        STORE POS_COUNT
        JUMP  END_IF
ELSE,   LOAD  NEG_COUNT
        ADD   ONE
        STORE NEG_count
END_IF, INPUT
        STORE X
        JUMP  WHILE
END_WHILE, HALT
```

INPUT

WHILE-loop
- condition - skipcond.
- jumps

IF: condition
jumps

```
POS_COUNT, DEC 0
NEG_count, DEC 0
X,         DEC 0
ONE,       DEC 1
```

25

Question 3. a) (5 points) For the below MARIE program, complete the symbol table.

Symbol	Address of Symbol (in hex.)
ELSE	108
END_IF	10A
IF	102
TEN	10B
X	10C

b) (5 points) Translate the given MARIE assembly language into machine language.

Address	Label	Assembly Language	Machine Language (in hex)
100_{16}		INPUT	5000
101_{16}		STORE X	210C
102_{16}	IF,	SUBT TEN	410B
103_{16}		SKIPCOND 800	8800
104_{16}		JUMP ELSE	9108
105_{16}		LOAD X	110C
106_{16}		OUTPUT	6000
107_{16}		JUMP END_IF	910A
108_{16}	ELSE,	LOAD TEN	110B
109_{16}		OUTPUT	6000
$10A_{16}$	END_IF,	HALT	7000
$10B_{16}$	TEN,	DEC 10	000A
$10C_{16}$	X,	DEC 0	0000

c) (10 points) Translate the above MARIE assembly language into high-level language "pseudo" code.

```
INPUT X
IF X>10 THEN
    OUTPUT X
ELSE
    OUTPUT 10
END_IF
```

Question 4. (10 points) Suppose we want to add a new Assembly Language instruction to MARIE called ADD_IMMEDIATE with opcode F_{16}. For example, the instruction "ADD_IMMEDIATE 12" would add the specified constant 12_{10} to the AC. The corresponding machine language instruction would be "$F00C_{16}$" where "F" is the opcode and "00C" is the constant value 12_{10}. Complete the RTN steps to Fetch-Decode-Execute the new ADD_IMMEDIATE instruction.

MAR ← PC
MBR ← M[MAR]
IR ← MBR
PC ← PC + 1
MBR ← IR[11-0]
AC ← AC + MBR

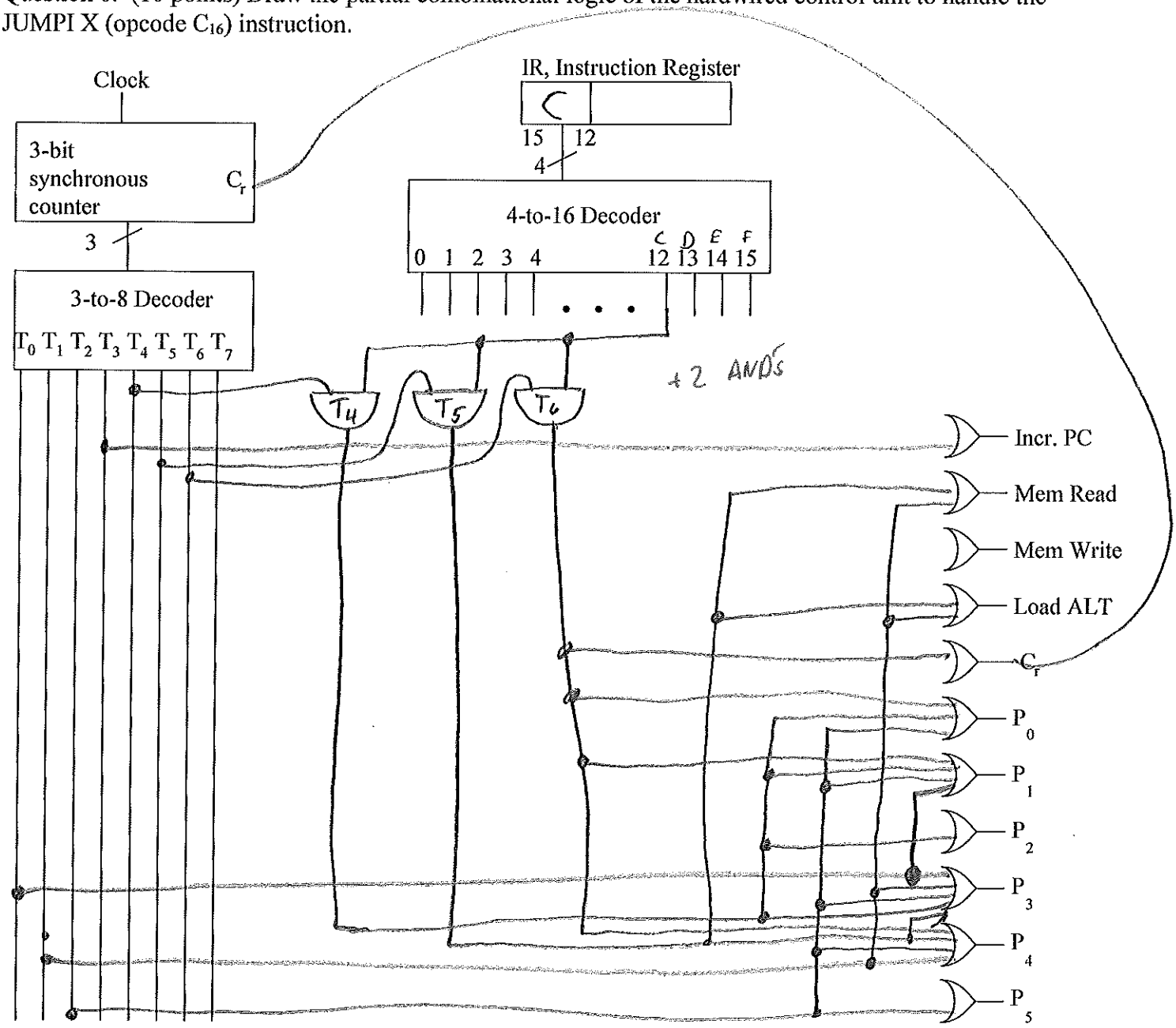
30

Spring 2014 4 2 1 4 2 1 Name: _____

Question 5. (10 points) Which control signals should contain a "1" for each steps in the JUMPI X instruction?

Step	RTN	Step #	P_5	P_4	P_3	P_2	P_1	P_0	C_r	Incr PC	Mem Read	Mem Write	Load ALT
Fetch	MAR ← PC	T_0	0	0	1	0	1	0	0	0	0	0	0
	MBR ← M[MAR]	T_1	0	1	1	–	–	–	0	0	1	0	1
	IR ← MBR	T_2	1	1	1	0	1	1	0	0	0	0	0
Decode IR[15-12]	PC ← PC + 1	T_3	0	0	0	–	–	–	0	1	0	0	0
Get operand	MAR ← IR[11-0]	T_4	0	0	1	1	1	1	0	0	0	0	0
Execute	MBR ← M[MAR]	T_5	0	1	1	–	–	–	0	0	1	0	1
	PC ← MBR	T_6	0	1	0	0	1	1	1	0	0	0	0
		T_7											

Question 6. (10 points) Draw the partial combinational logic of the hardwired control unit to handle the JUMPI X (opcode C_{16}) instruction.



Spring 2014 Name: _____

Question 7. Consider the microprogrammed version of MARIE.

a) (8 points) Explain why the microprogrammed control unit is slower at **decoding** which machine-language instruction was fetched than a hardwired control unit?

In the hardwired control unit one clock cycle (T_3) is needed for the Decoder circuit to "decode" the ML opcode. In the microprogrammed control unit several "If IR[15-12] = MicroOp[4-1]" microoperations (1 to 14) are needed which each taking multiple clock cycles.

b) (17 points) Extend the partial microprogram below to include microoperations to decode and implement the execution of the instructions: JUMPI X and OUTPUT. **(Fill in only the bolded boxes)**

Revised Figure 4.23 Partial Microprogram

Part of Cycle	RTN (of MicroOp1)	μAddr	MicroOp1	MicroOp2	Jump	Dest
Fetch	MAR ← PC	0	01010	00000	0	0
	MBR ← M[MAR]	1	01101	00000	0	0
	IR ← MBR	2	00110	00000	0	0
	PC ← PC + 1	3	10001	00000	0	0
Decode	If ADD, Jump	4	11000	00110	1	19_{10}
("Jump Table")	If LOAD, Jump	5	11000	00010	1	
	If STORE, Jump	6	11000	00100	1	
	If SKIPCOND, Jump	7	11000	10000	1	
	If SUBT, Jump	8	11000	01000	1	
	If JUMP, Jump	9	11000	10010	1	
	If ADDI, Jump	10	11000	10110	1	
	If CLEAR, Jump	11	11000	10100	1	
	If JNS, Jump	12	11000	00000	1	
	If JUMPI, Jump	13	11000	11000	1	22
	If INPUT, Jump	14	11000	01010	1	
	If OUTPUT, Jump	15	11000	01100	1	25
	If LOADI, Jump	16	11000	11010	1	
	If STOREI, Jump	17	11000	11100	1	
	If HALT, Jump	18	11000	01110	1	0
Execute ADD	MAR ← IR[11-0]	19	01000	00000	0	0
	MBR ← M[MAR]	20	01101	00000	0	0
	AC ← AC + MBR	21	00100	00000	1	0
Execute JUMPI	MAR ← IR[11-0]	22	01000	00000	0	0
	MBR ← M[MAR]	23	01101	00000	0	0
	PC ← MBR	24	10000	00000	1	0
Execute OUTPUT	OutREG ← AC	25	01110	00000	1	0
Execute SKIPCOND	...	26				