**18. Given a memory of 2048 bytes consisting of several 64 × 8 RAM chips, and**

**assuming byte-addressable memory, which of the following seven diagrams**

**indicates the correct way to use the address bits? Explain your answer.**

c). 11 bit address

|  |  |
| --- | --- |
| 6 bits chip select | 5 bits for address on chip |

c). because 64 x 8 RAM =23 \*26 = 6+3 =9 that mean we need a least 9 bits 6 for chip select and 3 for address in chip

**19. Explain the steps in the fetch–decode–execute cycle. Your explanation should**

**include what is happening in the various registers.**

Fetch is step that CPU get data from memory

Step 1 MAR 🡨 PC

Step 2 IR 🡨 M[MAR]

Step 3 PC 🡨 PC + 1

Decode is step that CPU decode data

Step 1 MAR 🡨 IR[11 - 0]

Execute is step that CPU calculate task

Step 1 AC 🡨 MBR

**21. Explain why, in MARIE, the MAR is only 12 bits wide and the AC is 16 bits**

**wide. (Hint: Consider the difference between data and addresses.)**

MAR only contain address 000 to FFF that mean it need only 12 bits but

AC need to contain value in memory 0000 to FFFF that mean it need 16 bits.

**22. List the hexadecimal code for the following program (hand assemble it).**

|  |  |  |  |
| --- | --- | --- | --- |
| **HEX Address** | **Label** | **Instruction** | HEX code |
| **100** |  | **LOAD A** | 1108 |
| **101** |  | **ADD ONE** | 3109 |
| **102** |  | **JUMP S1** | 9106 |
| **103** | **S2,** | **ADD ONE** | 3109 |
| **104** |  | **STORE A** | 2108 |
| **105** |  | **Halt** | 7000 |
| **106** | **S1,** | **ADD A** | 3108 |
| **107** |  | **JUMP S2** | 9103 |
| **108** | **A,** | **HEX 0023** |  |
| **109** | **One,** | **HEX 0001** |  |

**23. What are the contents of the symbol table for the preceding program?**

Symbol table is the table that show HEX address where is the variable store.

**24. Consider the MARIE program below.**

**a) List the hexadecimal code for each instruction.**

|  |  |  |  |
| --- | --- | --- | --- |
| **HEX Address** | **Label** | **Instruction** | HEX code |
| **100** | **Start,** | **LOAD A** | 1108 |
| **101** |  | **ADD B** | 3109 |
| **102** |  | **STORE D** | 210B |
| **103** |  | **CLEAR** | A103 |
| **104** |  | **OUTPUT** | 6104 |
| **105** |  | **ADDI D** | B10B |
| **106** |  | **STORE B** | 2109 |
| **107** |  | **HALT** | 7000 |
| **108** | **A,** | **HEX 00FC** |  |
| **109** | **B,** | **DEC 14** |  |
| **10A** | **C,** | **HEX 0108** |  |
| **10B** | **D,** | **HEX 0000** |  |

**b) Draw the symbol table.**

|  |  |
| --- | --- |
| A | 0 x 108 |
| B | 0 x 109 |
| C | 0 x 10A |
| D | 0 x 10B |

**c) What is the value stored in the AC when the program terminates?**

AC =108

**27. Write the assembly language equivalent of the following MARIE machine**

**language instructions:**

1. **0111000000000000**

0111 0000 0000 0000 = 7000 = HAILT

1. **1011001100110000**

1011 0011 0011 0000 = B330 = ADDI 330

1. **0100111101001111**

0100 1111 0100 1111 = 4F8F = Subt F8F

**33. Write the following code segment in MARIE assembly language:**

**X = 1;**

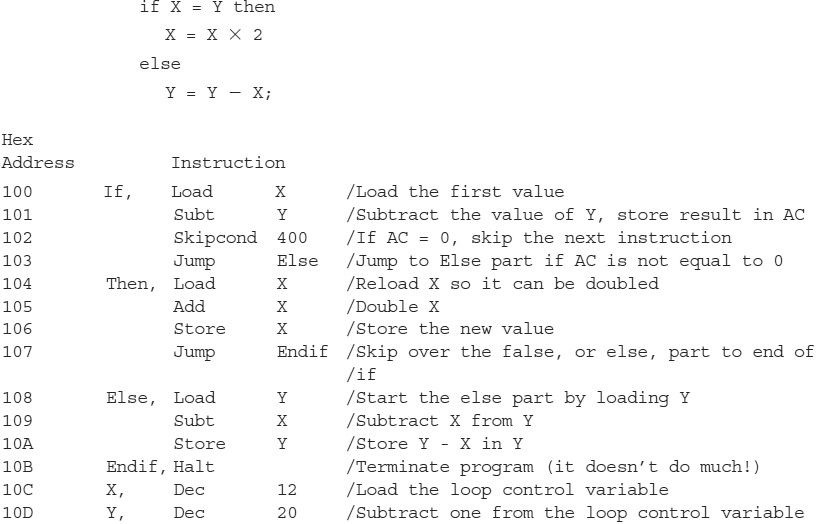
**While X <10 do**

**X = X +1;**

**Endwhile;**

|  |  |  |  |
| --- | --- | --- | --- |
| HEX Address | Label | Instruction | HEX code |
| 100 |  | LOAD X | 1109 |
| 101 | S2 | Subt Y | 410A |
| 102 |  | SKIPS 000 | 8000 |
| 103 |  | JUMP S1 | 9108 |
| 104 |  | LOAD X | 1109 |
| 105 |  | ADD Z | 310B |
| 106 |  | STORE X | 2109 |
| 107 |  | JUMP S2 | 9101 |
| 108 | S1, | Halt | 7000 |
| 109 | X | HEX x 0001 |  |
| 10A | Y | HEX x 000A |  |
| 10B | Z | HEX x 0001 |  |

**41. Provide a trace (similar to the one in Figure 4.14) for Example 4.3. (Provide the trace for the first two instructions at address 100 and 101 only.)**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | RTN | PC | IR | MAR | MBR | AC |
| LOAD 100 |  |  |  |  |  |  |
| (initial values) |  | 100 |  |  |  |  |
| Fetch | MAR 🡨 PC | 100 |  | 101 |  |  |
|  | IR 🡨 M[MAR] | 100 | 110C | 101 |  |  |
|  | PC 🡨 PC + 1 | 101 | 110C | 101 |  |  |
| Decode | MAR 🡨 IR[11 - 0] | 101 | 110C | 10C |  |  |
| Get Operand | MBR 🡨 M [MAR] | 101 | 110C | 10C | C |  |
| Execute | AC 🡨 MBR | 101 | 110C | 10C | C | C |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | RTN | PC | IR | MAR | MBR | AC |
| Subt 101 |  |  |  |  |  |  |
| (initial values) |  | 101 | 110C | 10C | C | C |
| Fetch | MAR 🡨 PC | 101 | 110C | 101 | C | C |
|  | IR 🡨 M[MAR] | 101 | 410D | 101 | C | C |
|  | PC 🡨 PC + 1 | 102 | 410D | 101 | C | C |
| Decode | MAR 🡨 IR[11 - 0] | 102 | 410D | 10D | C | C |
| Get Operand | MBR 🡨 M [MAR] | 102 | 410D | 10D | 14 | C |
| Execute | AC 🡨 MBR | 102 | 410D | 10D | 14 | FFFA |