**Reg.No:** | | | | | | | |

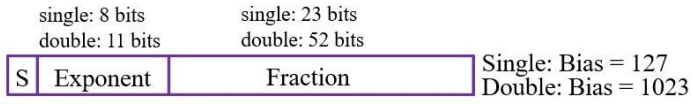
**SNS College of Technology, Coimbatore-35.**
**(Autonomous)**
**B.E/B.Tech- Internal Assessment -II**
**Academic Year 2023-2024(ODD)**
**Third Semester**
**Computer Science and Engineering**

A

**19ITT202 Computer Organization and Architecture**
**[Common to CSE & IT]**

**Time: 1.5 Hours** **Maximum Marks: 50**

**Answer All Questions**

| | **PART - A (5x 2 = 10 Marks)** | **CO** | **Blooms** |
|---|---|---|---|
| 1. | List out the advantages of Booth Algorithm<br>1) It handles both positive and negative multiplier uniformly.<br>2) It achieves efficiency in the number of additions required when the multiplier has afew large blocks of 1's.<br>3) The speed gained by skipping 1's depends on the data. | CO2 | Und |
| 2. | Define floating point number representation and Recall value representation equation for single precision<br>The description of binary numbers in the exponential form is called floating-pointrepresentation.<br>A Single-Precision floating-point number occupies 32-bits, so there is a compromise betweenthe size of the mantissa and the size of the exponent.<br>  <br>S: sign bit<br>0 –non - negative<br>1 - negative<br>Value Represented = ±1.M x 2 $^{E' - 127}$ | CO2 | Rem |
| 3. | Consider the instruction Add (R3),R1 & write the action required to execution of the above mentioned complete instruction | CO2 | Ana |

- Add (R3), R1
- Fetch the instruction
- Fetch the first operand (the contents of the memory location pointed to by R3)
- Perform the addition
- Load the result into R1

| Step | Action |
|---|---|
| 1 | $PC_{out}$ , $MAR_{in}$ , Read, Select4,Add, $Z_{in}$ |
| 2 | $Z_{out}$ , $PC_{in}$ , $Y_{in}$ , WMFC |
| 3 | $MDR_{out}$ , $IR_{in}$ |
| 4 | $R3_{out}$ , $MAR_{in}$ , Read |
| 5 | $R1_{out}$ , $Y_{in}$ , WMFC |
| 6 | $MDR_{out}$ , SelectY, Add, $Z_{in}$ |
| 7 | $Z_{out}$ , $R1_{in}$ , End |

4. What are the 4 different phases of Pipelining — CO3 — Rem

A pipelined processor uses a 4-stage instruction pipeline with the following stages:

- Instruction fetch (IF)
- Instruction decode (ID)
- Execute (EX) and
- Writeback (WB)

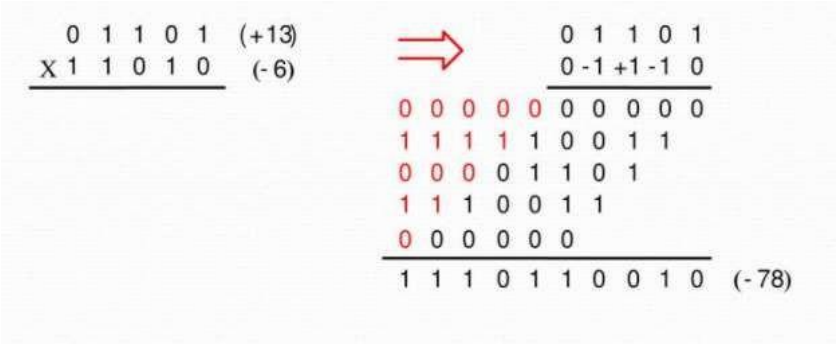
5. Classify the different types of pipeline hazards — CO3 — Und

1. Data hazard
2. Structural Hazard
3. Control Hazard.

**PART – B (13+13+14 = 40 Marks)**

6. (a) Multiply the following pair of signed 2's complement numbers using the Booth algorithm. Assume that A is the multiplicand & B is the multiplier. A=01101 & B=11010. Compare it with normal multiplication and explain the concept of Booth recoding of multiplier — 13 — CO2 — App

| Multiplier | | Version of multiplicand selected by bit |
|---|---|---|
| Bit *i* | Bit *i*-1 | |
| 0 | 0 | 0 X M |
| 0 | 1 | +1 X M |
| 1 | 0 | –1 X M |
| 1 | 1 | 0 X M |

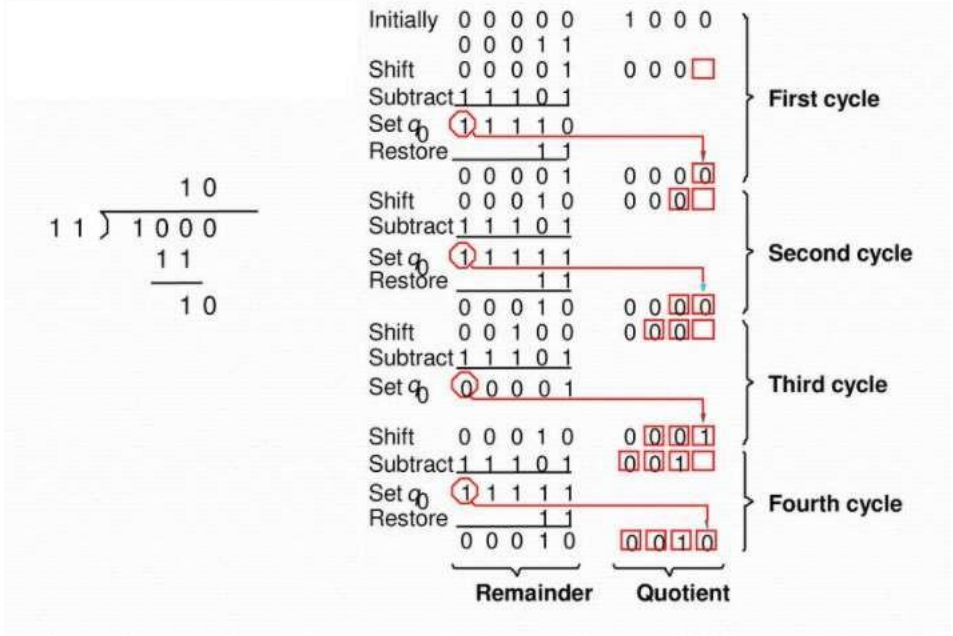



(or)

(b) Elaborate the concept of Integer division. Describe and analyze the difference between restoring and non-restoring division algorithm & apply any one algorithm execution for Dividend value 1000 and Divisor value 11. 13 CO2 App

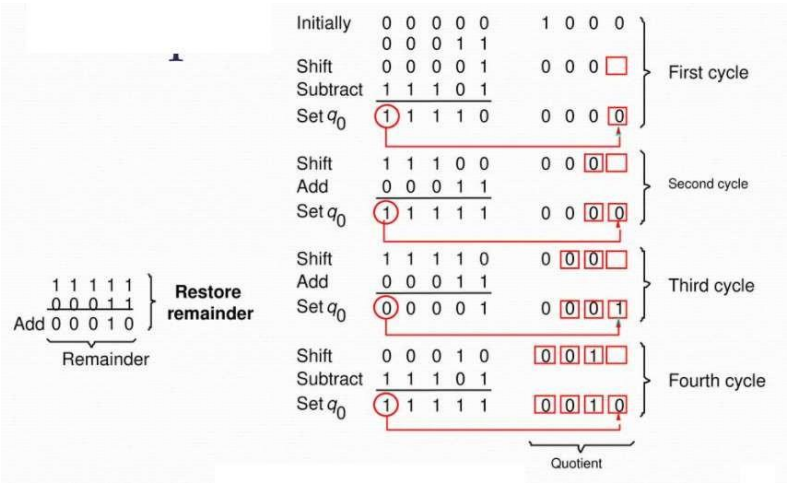
**Restoring Division**

- Shift A and Q left one binary position
- Subtract M from A, and place the answer back in A
- If the sign of A is 1, set $q_0$ to 0 and add M back to A (restore A); otherwise, set $q_0$ to 1
- Repeat these steps $n$ times

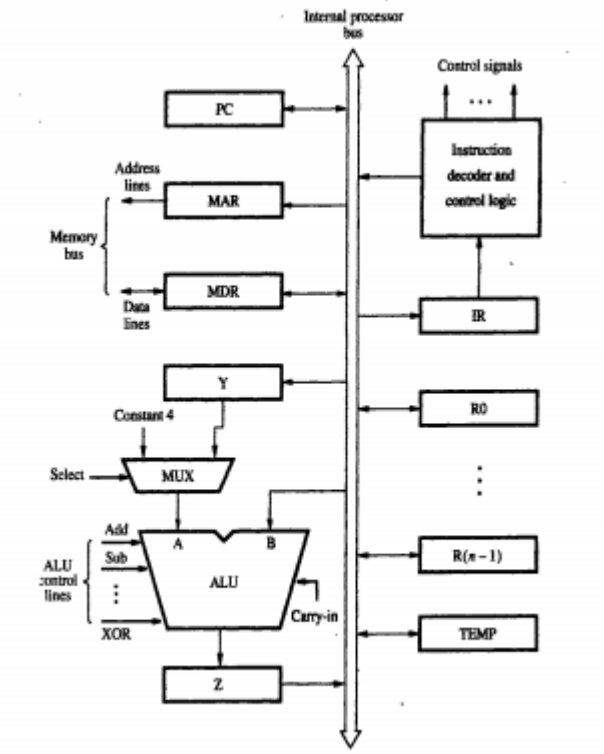


**Non - Restoring Division**

- Avoid the need for restoring A after an unsuccessful subtraction.
- Any idea?
- Step 1: (Repeat *n* times)
  - If the sign of A is 0, shift A and Q left one bit position and subtract M from A; otherwise, shift A and Q left and add M to A.
  - Now, if the sign of A is 0, set $q_0$ to 1; otherwise, set $q_0$ to 0.
- Step2: If the sign of A is 1, add M to A




7. (a) Illustrate the architectural organization of a processor and elaborate the steps needed for execution of a complete instruction. 13 CO3 Und




**Execution of processor Instruction**

- Fetch the contents of the memory location pointed to by the

PC. The contents of thislocation are loaded into the IR (fetch phase).

$$IR \leftarrow [[PC]]$$

- Assuming that the memory is byte addressable, increment the contents of the PC by 4(fetch phase).

$$PC \leftarrow [PC] + 4$$

- Carry out the actions specified by the instruction in the IR (execution phase).

- Transfer a word of data from one processor register to another or to the ALU
- Perform an arithmetic or a logic operation and store the result in a processor register
- Fetch the contents of a given memory location and load them into a processor register
- Store a word of data from a processor register into a given memory location

Add(R3),R1

| Step | Action |
|---|---|
| 1 | $PC_{out}$, $MAR_{in}$, Read, Select4, Add, $Z_{in}$ |
| 2 | $Z_{out}$, $PC_{in}$, $Y_{in}$, WMFC |
| 3 | $MDR_{out}$, $IR_{in}$ |
| 4 | $R3_{out}$, $MAR_{in}$, Read |
| 5 | $R1_{out}$, $Y_{in}$, WMFC |
| 6 | $MDR_{out}$, SelectY, Add, $Z_{in}$ |
| 7 | $Z_{out}$, $R1_{in}$, End |

(or)

(b) Summarize the processor execution of Hard-wired Control and Microprogrammed control instruction execution. Consider executing the instruction Add (R3),R1 by the processor. analyse the actions and control instructions needed for execution. 13 CO3 App

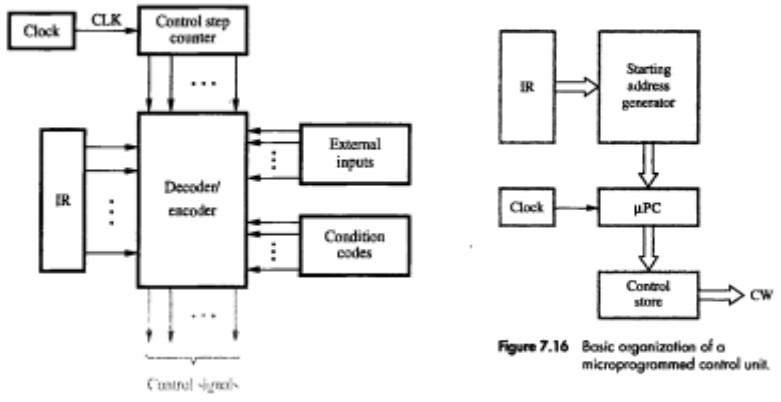
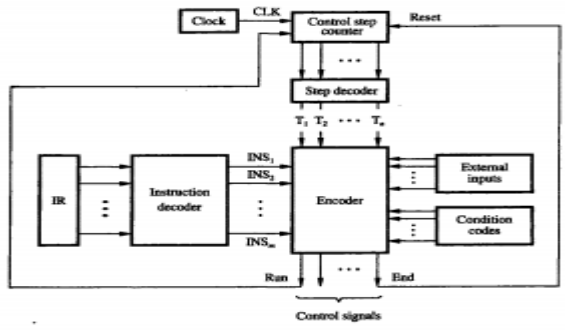



Figure 7.16 Basic organization of a microprogrammed control unit.



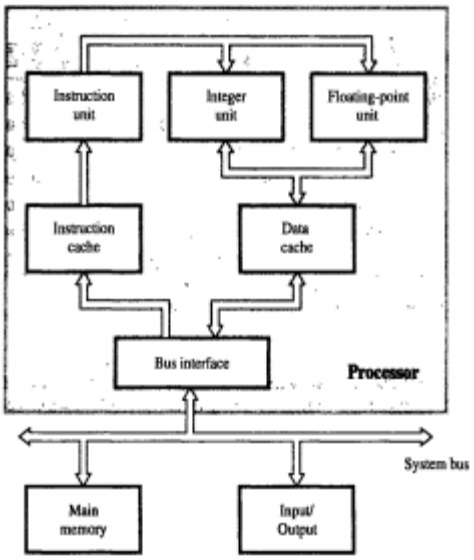


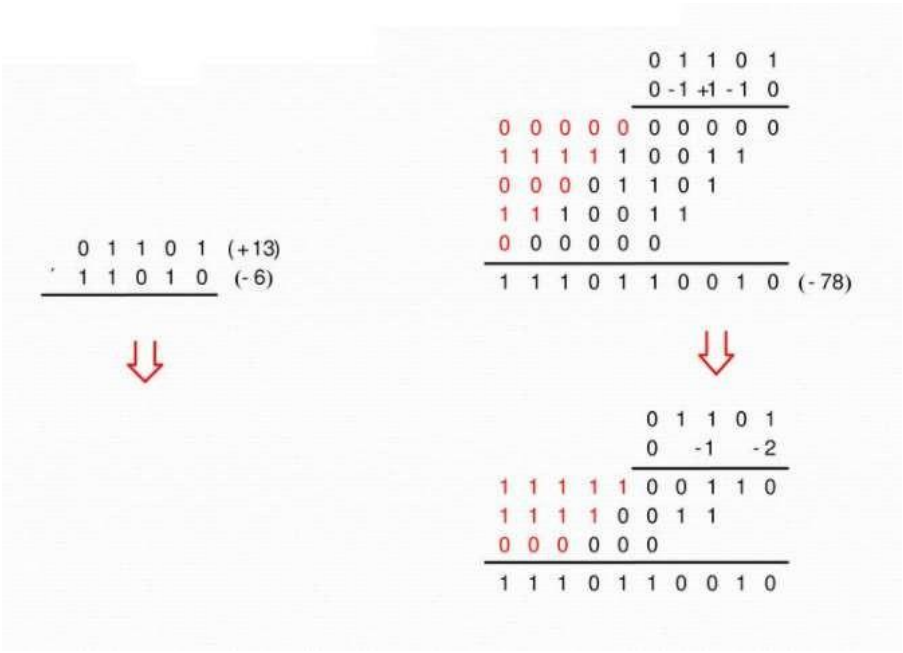

Figure 7.14 Block diagram of a complete processor.

| Step | Action |
|---|---|
| 1 | $PC_{out}$, $MAR_{in}$, Read, Select4, Add, $Z_{in}$ |
| 2 | $Z_{out}$, $PC_{in}$, $Y_{in}$, WMFC |
| 3 | $MDR_{out}$, $IR_{in}$ |
| 4 | $R3_{out}$, $MAR_{in}$, Read |
| 5 | $R1_{out}$, $Y_{in}$, WMFC |
| 6 | $MDR_{out}$, SelectY, Add, $Z_{in}$ |
| 7 | $Z_{out}$, $R1_{in}$, End |

| Micro-instruction | .. | $PC_{in}$ | $PC_{out}$ | $MAR_{in}$ | Read | $MDR_{out}$ | $IR_{in}$ | $Y_{in}$ | Select | Add | $Z_{in}$ | $Z_{out}$ | $R1_{out}$ | $R1_{in}$ | $R3_{out}$ | WMFC | End | .. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 2 | | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| 3 | | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 4 | | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | |
| 5 | | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | |
| 6 | | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 7 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | |

Figure 7.15 An example of microinstructions for Figure 7.6.

8. (a) Illustrate the advantages of fast multiplication over other multiplication algorithms and apply the two types of fast multiplication concept for the following values. Assume that A is the multiplicand & B is the multiplier. A=01101 & B=11010. 14 CO2 App




(or)

(b) Analyze the execution of processor instructions in pipelining and how the performance got affected on the occurrence of Data and Instruction Hazards in pipelining, along with methods of handling its delay — 14 CO3 Ana

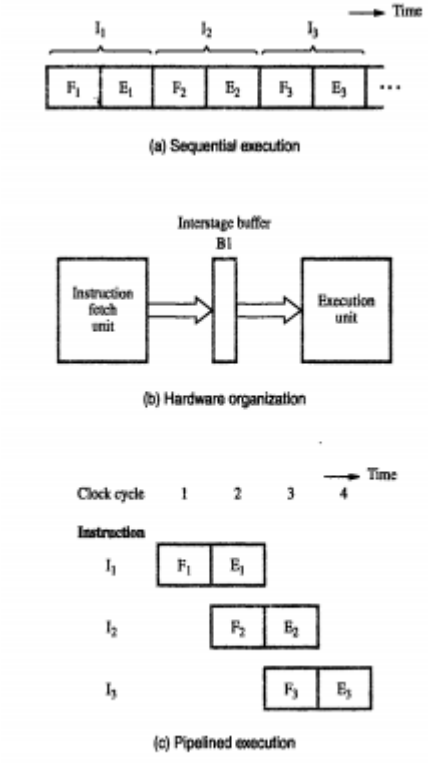



Figure 8.1 Basic idea of instruction pipelining.

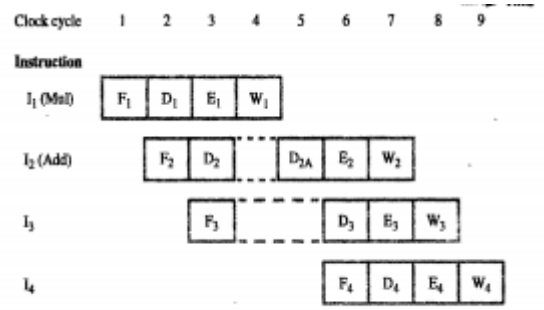
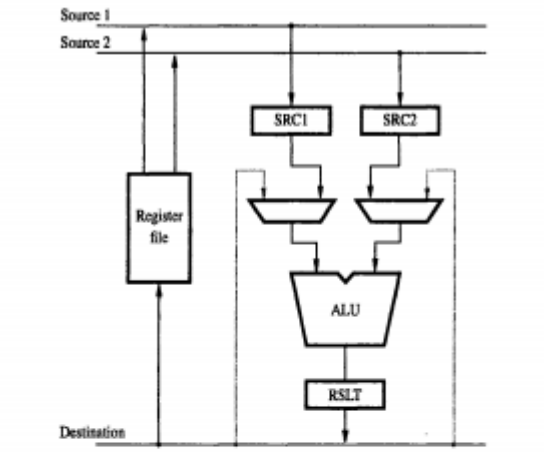
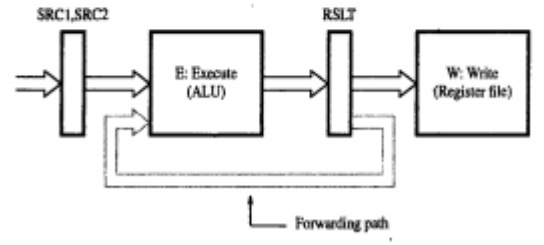



Figure 8.6 Pipeline stalled by data dependency between $D_2$ and $W_1$.




(a) Datapath




(b) Position of the source and result registers in the processor pipeline

Figure 8.7 Operand forwarding in a pipelined processor.

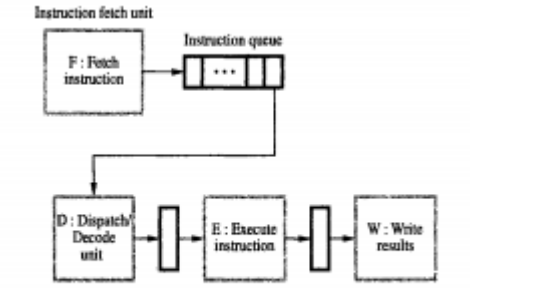



Figure 8.10 Use of an instruction queue in the hardware organization of Figure 8.2b.

**********

**(Note: Und-Understand Rem-Remember Ana-Analyze App-Apply Cre- Create)**

Prepared By Verified By HoD