

SAVE TREES USE E-RESOURCES

**ALL COURSES | ALL SEMESTERS | NOTES |
SOLVED QUESTION PAPER | LAB MANUALS**



BU **STUDYMATE**

Connecting Minds...!



/Bustudymate.in



70194 28102

Click here to Join Our WhatsApp Group



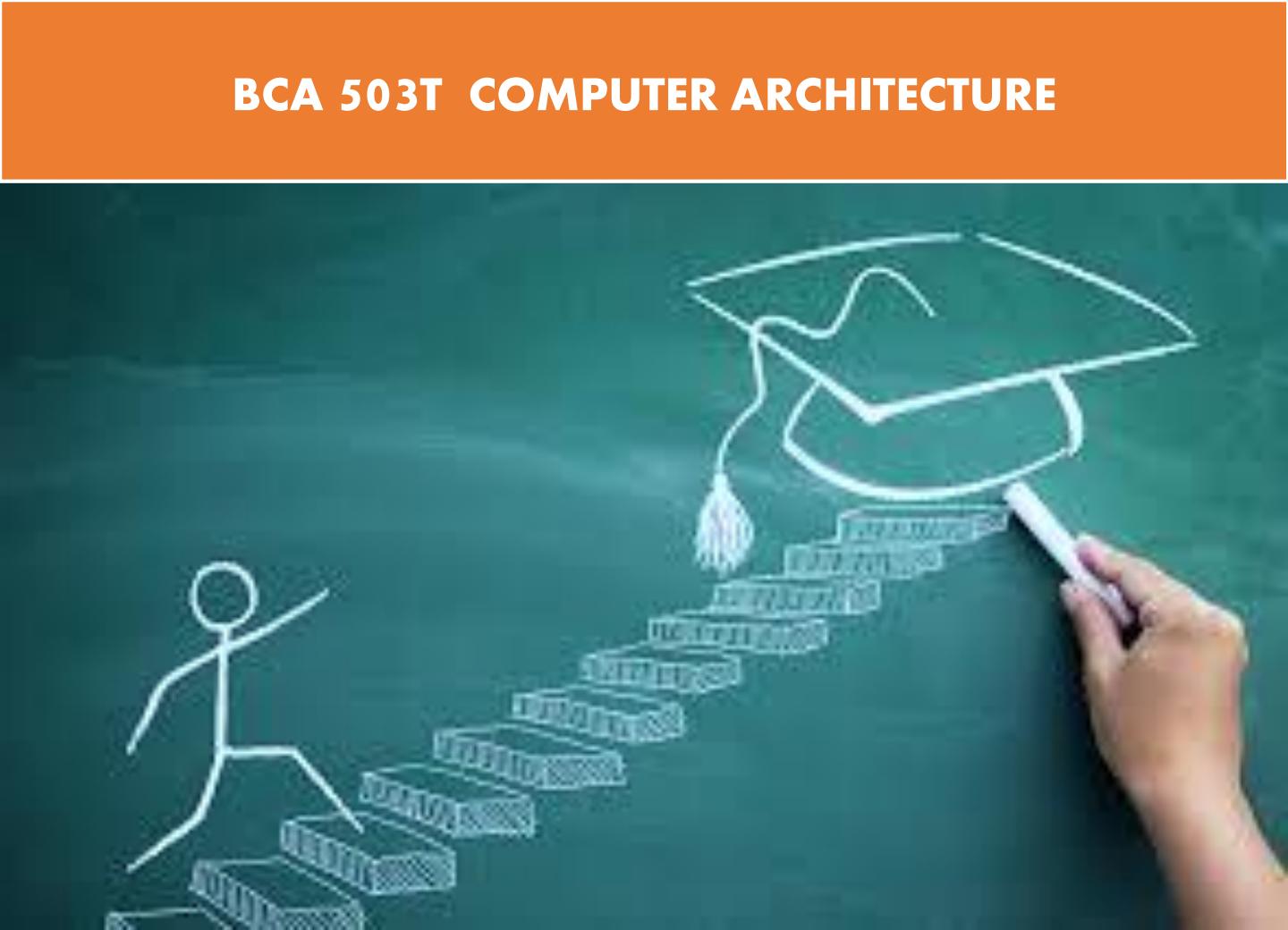
Soundarya Educational Trust

Soundarya Institute of Management & Science
Affiliated to Bangalore University || NAAC Accredited Institution || Approved by Govt.
of Karnataka

Department of Computer Science



BCA 503T COMPUTER ARCHITECTURE



BCA503T: COMPUTER ARCHITECTURE

Total Teaching Hours: 60

No of Hours / Week: 04

Unit - I

DIGITAL LOGIC CIRCUITS: Logic gates Boolean algebra, map simplification, combinational circuits, flip-flop, sequential circuits. **INTEGRATED CIRCUITS AND DIGITAL FUNCTIONS:** Digital integrated circuits, IC flip –flops and registers, decoders and multiplexers, binary counters, shift registers, random –access memories (RAM) read –only memories (ROM). [12 Hours]

Unit - II

DATA REPRESENTATION: Data types, fixed-point representation, floating – point representation, other binary codes, error detection codes. **DATA TRANSFER OPERATIONS:** Register Transfer, Memory Transfer and I/O Transfer. [12 Hours]

Unit – III

BASIC COMPUTER ORGANISATION AND DESIGN: Instruction codes, computer instruction, timing and control, execution and instruction, input-output and interrupt, design of computer. [12 Hours]

Unit - IV

CENTRAL PROCESSOR ORGANIZATION: Processor bus organization, arithmetic logic unit (ALU) instruction formats, addressing modes, data transfer and manipulation, program control, microprocessor organization. [12 Hours]

Unit – V

INPUT-OUTPUT ORGANISATION: Peripheral devices. asynchronous data transfer, direct memory access (DMA), priority interrupt, input –output processor (IOP). **MEMORY ORGANIZATION:** Auxiliary memory, microcomputer memory hierarchy, associative memory, virtual memory, cache memory. [12 Hours]

Text Books:

1. M. Morris Mano, Computer System, Architecture, 2nd Edition Prentice Hall of India.

Reference Books:

1. Heuring and Jordan, Computer systems design and Architecture, Pearson Edition
2. William Stallings, Computer Organization and Architecture, Pearson Education
3. Floyd, Digital Fundamentals,8th Edition, Pearson Education.
4. Andrew S. Tanenbaum, Structured Computer Organization, 3rd Edition; Prentice Hall of India.
5. David Patterson & Hennessy, Computer Organization & Design, Elsevier.

BCA503T: COMPUTER ARCHITECTURE
BLUE PRINT

Question paper pattern for theory has four sections :

Section – A :Contains 12 questions, out of which a student has to answer 10 questions. Each question carries 2 marks ($10 \times 2 = 20$)

Section – B :Contains 8 questions, out of which a student has to answer 5 questions. Each question carries 5 marks ($5 \times 5 = 25$)

Section – C :Contains 5 full questions includes sub-question as (a) & (b). Student has to answer 3 full questions. Each full question carries 15 marks($3 \times 15 = 45$)

Section – D :Contains 2 questions, out of which a student has to answer 1 question. Each question carries 10 marks ($1 \times 10 = 10$)

UNIT	CHAPTER	SECTION A 2 MARKS	SECTION B 5 MARKS	SECTION C 15 MARKS	SECTION D 10 MARKS
I	Digital Logic Circuits	2	1	0.5	0.5
	Integrated Circuits and Digital Functions	2	1	1	0.5
II	Data Representation	2	1	0.5	-
III	Basic Computer Organization and Design	2	2	1	0.5
IV	Central Processor Organization	2	1	1	0.5
V	Input-Output Organization	1	1	0.5	-
	Memory Organization	1	1	0.5	-
	Total Questions	12	8	5	2
		Answer any 10	Answer any 5	Answer any 3	Answer any 1
	Total Marks (100)	20	25	45	10

SECTION – A (2 Marks)

UNIT-I

[Nov / Dec 2015]

1. State and prove De-Morgan's theorem
2. Draw the logic diagram of Boolean function $F = AB + A' B$ using NAND gates only.
3. What is decoder expansion?
4. What is unidirectional and bidirectional shift register

[Nov / Dec 2016]

5. What is computer architecture?
6. State and prove De-Morgan's theorem
7. Mention the different logic families of IC.
8. Distinguish between RAM and ROM.

[Nov / Dec 2017]

9. Write the symbol, logical expression and truth table of NAND gate.
10. Give the classification of integrated circuits.
11. Distinguish between RAM and ROM.
12. Define Multiplexer and Demultiplexer.

[Nov / Dec 2018]

13. Explain full adder.
14. Define universal gates with logic circuit.
15. State and prove De-Morgan's theorem
16. Define Flip-Flop.
17. Why we use shift register?

[TMAQ - Important Tutor Mark Assignment Questions]

18. Define half adder.
19. What is full subtractor.
20. Define parity checker and generator.
21. What is k-map?
22. What are the different types of flipflops.
23. Mention the types of shift registers.
24. Mention the types of RAM and ROM.
25. Mention the different types of computer architecture.

SECTION – A (2 Marks)

UNIT-II

[Nov / Dec 2015]

1. Convert $(736.4)_8$ into decimal and binary
2. What is self-complementing code and weighted code?

[Nov / Dec 2016]

3. What is parity bit?
4. Write the BCD code for decimal number $8745.42_{(10)}$

[Nov / Dec 2017]

5. What are the types of Binary codes?
6. Subtract 24 from 13 using 2's complement method.

[Nov / Dec 2018]

7. Explain Hamming code.
8. Define Parity bit.

[TMAQ - Important Tutor Mark Assignment Questions]

9. What self-complementary code?
10. What is true complement and radix-1 complement?
11. What is alphanumeric code?
12. Define floating point representation.

SECTION – A (2 Marks)

UNIT-III

[Nov / Dec 2015]

1. What are the two types of control organization?
2. How many bits are needed to specify an address for a memory unit of 4096 words.

[Nov / Dec 2016]

3. What are the two types of control organization?
4. Define program counter.

[Nov / Dec 2017]

5. Define opcode and operand.
6. What is BUN instruction?

[Nov / Dec 2018]

7. Explain BSA instruction.
8. Define Indirect Addressing Mode.

[TMAQ - Important Tutor Mark Assignment Questions]

9. What is the difference between BUN and BSA instruction.
10. What is stored program concept?
11. What do you mean by instruction execution?
12. What is timing control?
13. Define Hardwired Control.

SECTION – A (2 Marks)

UNIT-IV

[Nov / Dec 2015]

1. What is PSW?
2. What is an external interrupt? Give an example?

[Nov / Dec 2016]

3. Mention the major components of CPU.
4. What is PSW?

[Nov / Dec 2017]

5. What are the two types of computer architecture based on registers?
6. What are the different types of interrupts?

[Nov / Dec 2018]

No Questions

[TMAQ - Important Tutor Mark Assignment Questions]

7. What is RTL?
8. What do you mean by microoperations.
9. What is the difference between single and stack organization.
10. What is word?
11. What is the expansion of RISC and CISC.

SECTION – A (2 Marks)

UNIT-V

[Nov / Dec 2015]

1. What are peripherals?
2. What is memory management system?

[Nov / Dec 2016]

3. What is polling?
4. What is memory management system?

[Nov / Dec 2017]

5. Define access time and transfer rate.
6. Define Baud rate.

[Nov / Dec 2018]

7. What is meant by Memory-Mapped I/O?
8. Define virtual memory
9. Define types of RAM.

[TMAQ - Important Tutor Mark Assignment Questions]

10. Mention modes of transfer.
11. What is daisy chain priority?
12. What is cache memory?
13. Define Associate memory?

SECTION – B (5 Marks)

UNIT-I

[Nov / Dec 2015]

1. Simplify the Boolean function $F(A,B,C,D)=\sum(0,1,2,5,8,9,10)$ in both SOP and POS .
2. Design 4-to-1 multiplexer.

[Nov / Dec 2016]

3. Prove NAND and NOR gates as universal gates.
4. Explain PIPO shift register with a diagram.

[Nov / Dec 2017]

5. Explain the steps involved in the design of the sequential circuits.
6. Explain synchronous binary counter with logical diagram.

[Nov / Dec 2018]

7. Explain the steps involved in design of combinational circuit.
8. What is a K-map? Explain different types of K-maps.

SECTION – B (5 Marks)

UNIT-II

[Nov / Dec 2015]

1. Define r and (r-1)'s complement. Represent -14 using integer representation stored in an 8 bit register.

[Nov / Dec 2016]

2. Discuss the Parity generator and Parity checker.

[Nov / Dec 2017]

3. Discuss on error detection and correction codes briefly.

[Nov / Dec 2018]

4. NO Questions

SECTION – B (5 Marks)

UNIT-III

[Nov / Dec 2015]

1. List the micro operations of ADD and ISZ instruction.
2. Explain with neat block diagram the input-output configuration.

[Nov / Dec 2016]

3. Explain the operation of interrupt cycle with a flow chart.
4. Explain input-output organization.

[Nov / Dec 2017]

5. Explain any five register reference instructions.
6. With a block diagram, explain how BSA instruction execution.

[Nov / Dec 2018]

7. Write a note on program counter and stack memory.
8. Explain any five register reference instructions
9. Explain timing signals.

SECTION – B (5 Marks)

UNIT-IV

[Nov / Dec 2015]

1. Explain register stack with a neat block diagram.

[Nov / Dec 2016]

2. Explain the three types of CPU organization.

[Nov / Dec 2017]

3. Explain the addressing modes.

[Nov / Dec 2018]

4. Compare CICS and RISC processors.

SECTION – B (5 Marks)

UNIT-V

[Nov / Dec 2015]

1. What is polling? Explain.
2. Explain Associative memory with a neat diagram.

[Nov / Dec 2016]

3. Explain the source initiated data transfer using handshaking with a block diagram and timing diagram.
4. Write a note on memory hierarchy in a computer system.

[Nov / Dec 2017]

5. Explain DMA Controller with a block diagram.
6. Write a note on virtual memory.

[Nov / Dec 2018]

7. Write a note on cache memory.
8. What are the important characteristics of memory?

[TMAQ - Important Tutor Mark Assignment Questions]

NOTE: ALL TMAQ questions for 5 marks and 10 marks will be given at the end please check

SECTION –C(15 Marks)

UNIT-I

[Nov / Dec 2015]

1. Design a octal to binary encoder
2. Explain with a neat diagram a 4-bit bidirectional shift register with parallel load.

[Nov / Dec 2016]

3. Define K map? Simplify the following Boolean function using K-map:
 $F(A, B, C, D) = \sum(0, 2, 4, 6, 10, 11, 12, 13, 14, 15)$
4. Define counter. With a neat diagram explain 4-bit synchronous binary counter.
5. Explain octal to binary encoder with diagram.

[Nov / Dec 2017]

6. Simplify $F(ABCD) = \sum m(1, 3, 7, 11, 15) + \sum d(0, 2, 5)$ using K-map.
7. What is half adder? Design a half adder using only NAND gates.
8. Explain decoder expansion with the neat diagram.

[Nov / Dec 2018]

9. Simplify $F(ABCD) = \sum m(1, 2, 4, 6, 8, 10, 12, 14)$ and draw a circuit diagram.

SECTION –C(15 Marks)

UNIT-II

[Nov / Dec 2015]

1. Design the circuit for a 3-bit parity generator using an odd-parity system.

[Nov / Dec 2016]

2. Explain different binary codes.

[Nov / Dec 2017]

3. Discuss the parity generator and parity checker.

[Nov / Dec 2018]

4. What is parity bit? Explain in brief.

SECTION –C(15 Marks)

UNIT-III

[Nov / Dec 2015]

1. Explain with a neat flowchart the computer operation.

[Nov / Dec 2016]

2. Explain the design of basic computer with flow chart.

[Nov / Dec 2017]

3. No questions

[Nov / Dec 2018]

4. Explain the types of program interrupts.
5. Explain I/O commands.

SECTION –C(15 Marks)

UNIT-IV

[Nov / Dec 2015]

1. What is addressing mode? Explain the different types of Addressing Modes with example.

[Nov / Dec 2016]

2. What is addressing mode? Explain the different types of addressing modes with examples

[Nov / Dec 2017]

3. Explain common bus organization of basic computer with neat diagram.
4. Distinguish between FGI and FGO.
5. What is a sub-routine? Explain CAL and RETURN instructions.

[Nov / Dec 2018]

6. Explain common BUS organization of a Basic Computer
7. Explain different Addressing modes.

SECTION –C(15 Marks)

UNIT-V

[Nov / Dec 2015]

1. Explain source-initiated data transfer using hand shaking.
2. What is virtual memory? Explain address space and memory space in detail.

[Nov / Dec 2016]

3. Explain DMA controller with a block diagram/
4. Explain the working of Associative memory.

[Nov / Dec 2017]

5. Explain I/O interface unit with a neat diagram.
6. Write a note on isolated vs memory mapped I/O.

[Nov / Dec 2018]

7. Explain memory hierarchy.

SECTION –D(10 Marks)

UNIT-I

[Nov / Dec 2015]

1. What is binary counter? Explain a 4-bit synchronous counter with a neat block diagram.

[Nov / Dec 2016]

2. Explain the block diagram of computer with I/O processors.

[Nov / Dec 2017]

3. Explain 4-bit shift register.
4. Explain the working of JK flipflop.
5. List the applications of EEPROM.

[Nov / Dec 2018]

6. Explain the working of RS flip-flop.
7. Explain 8 to 3 encoder.

SECTION -D(10 Marks)

UNIT-II

[Nov / Dec 2015]

No questions

[Nov / Dec 2016]

No questions

[Nov / Dec 2017]

No questions

[Nov / Dec 2018]

1. Discuss error detection and correction codes.

SECTION -D(10 Marks)

UNIT-III

[Nov / Dec 2015]

No questions

[Nov / Dec 2016]

1. Explain the common bus system.

[Nov / Dec 2017]

No questions

[Nov / Dec 2018]

2. Explain direct address and indirect address modes.

SECTION -D(10 Marks)

UNIT-IV

[Nov / Dec 2015]

1. What are the major characteristics of RISC architecture?

[Nov / Dec 2016]

2. Write a note on RISC and CISC.

[Nov / Dec 2017]

3. Explain interrupt cycle with suitable example.

[Nov / Dec 2018]

No questions

SECTION –D(10 Marks)

UNIT-V

[Nov / Dec 2015]

1. Explain the block diagram of computer with I/O processors.

[Nov / Dec 2016]

2. Write a note on modes of data transfer

[Nov / Dec 2017]

No questions

[Nov / Dec 2018]

No questions

[TMAQ - Important Tutor Mark Assignment Questions]

Unit – I

1. Explain any ten Boolean postulates.
2. Write the difference between Von Neuman and Harvard Architecture.
3. Explain JK-Master slave flipflop.
4. Explain the classification of IC Families.
5. Explain SISO in brief.

Unit – II

1. Explain floating point representation in brief.
2. Explain in brief Gray code.
3. What is Excess 3 code? Explain.
4. What is cyclic code? Explain.
5. Explain code conversion with an example.

Unit – III

1. What is stored program organization? Explain.
2. Explain computer registers.
3. Explain instruction execution.
4. Explain computer instruction with an example.
5. Explain the design of the computer with a flowchart.

Unit – IV

1. Explain Bus organization with a neat diagram.
2. Explain the formats of instructions.
3. What is program interrupt? Explain.
4. Distinguish between RISC and CISC.
5. Explain the arithmetic and logic unit.

Unit – V

1. Give the difference between Isolated I/O vs Memory mapped I/O.
2. Explain strobe control with an example.
3. Explain DMA with a neat diagram.
4. Explain in brief Cache memory and Virtual memory.
5. Explain classification of memory. Explain in brief.



V Semester B.C.A. Degree Examination, November/December 2015
(Y2K8 Scheme) (F + R)

Computer Science
BCA – 502 : COMPUTER ARCHITECTURE
(100 – 2013-14 & Onwards) (90 – Prior to 2013-14)

Time : 3 Hours

Max. Marks : 90/100

- Instructions:**
- 1) Section A, B, C is common to all. Section D is applicable to the students of 2011-12 and Onwards.
 - 2) 100 marks for students of 2011-12 and onwards. 90 marks for Repeaters prior to 2011-12.

SECTION – A

- I. Answer **any ten** questions. Each carries two marks. **(10×2=20)**
 - 1) State and prove Demorgan's law.
 - 2) Draw the logic diagram of the Boolean function $F = AB + A' B$ using NAND gates only.
 - 3) What is Decoder Expansion ?
 - 4) What is unidirectional and bidirectional shift register ?
 - 5) Convert $(736.4)_8$ to decimal and binary.
 - 6) What is self complementing code and weighted code ?
 - 7) What are the two types of control organization ?
 - 8) How many bits are needed to specify an address for a memory unit of 4096 words ?
 - 9) What is PSW ?
 - 10) What is an external interrupt ? Give an example.
 - 11) What are peripherals ?
 - 12) What is memory management system ?



SECTION – B

II. Answer any five questions. Each carries five marks. (5x5=25)

- 13) Simplify the Boolean function $F(A, B, C, D) = \sum (0, 1, 2, 5, 8, 9, 10)$ in both sum-of-products and product-of-sums.
- 14) Design a 4-to-1 multiplexer.
- 15) Define r and $(r - 1)$'s complement. Represent -14 using Integer representation stored in an 8 bit register.
- 16) List the micro operations of ADD and ISZ instructions.
- 17) Explain with a neat block diagram the input-output configuration.
- 18) Explain register stack with a neat block diagram.
- 19) What is polling ? Explain.
- 20) Explain Associative memory with a neat block diagram.

SECTION – C

III. Answer any three questions. Each carries fifteen marks. (3x15=45)

21. a) Design a sequential circuit with two JK flip flops A and B and two inputs E and x.
If $E = 0$ the circuit remains in the same state regardless of the value of x.
When $E = 1$ and $x = 1$, the circuit goes through the state transition from 00 to 01 to 10 to 11 back to 00 and repeat.
When $E = 1$ and $x = 0$, the circuit goes through the state transition from 00 to 11 to 10 to 01 back to 00 and repeat.
b) Derive the circuit for a 3-bit parity generator using an odd-parity system. (10+5)
22. a) Design a octal to binary encoder.
b) Explain with a neat block diagram a 4-bit bidirectional shift register with parallel load. (5+10)



- 23) Explain with a neat flowchart the computer operation. 15
- 24) What is Addressing Mode ? Explain the different types of Addressing Modes with example. 15
- 25) a) Explain source-initiated data transfer using hand shaking.
b) What is virtual memory ? Explain address space and memory space in detail. (7+8)

SECTION – D

IV. Answer **any one** questions. **Each** carries **ten** marks. (1×10=10)

- 26) What is binary counter ? Explain a 4-bit synchronous counter with a neat block diagram. 10
- 27) a) What are the major characteristics of RISC architecture ?
b) Explain the block diagram of a computer with I/O processors. (5+5)
-



V Semester B.C.A. Degree Examination, Nov./Dec. 2016
(CBCS – Fresh – 2016 – 17 & Onwards)
BCA – 503 : COMPUTER ARCHITECTURE

Time : 3 Hours

Max. Marks : 100

Instruction : Answer all Sections.

SECTION – A

- I. Answer **any ten** questions. **Each** carries **two** marks. (10×2=20)
- 1) What is Computer Architecture ? 2
 - 2) State and prove DeMorgan's theorem. 2
 - 3) Mention the different logic families of IC. 2
 - 4) Distinguish between RAM and ROM. 2
 - 5) What is Parity bit ? 2
 - 6) Write the BCD code for decimal number $8745.42_{(10)}$. 2
 - 7) What are the two types of control organization ? 2
 - 8) Define program counter. 2
 - 9) Mention the major components of CPU. 2
 - 10) What is PSW ? 2
 - 11) What is Polling ? 2
 - 12) What is memory management system ? 2

SECTION – B

- II. Answer **any five** questions. **Each** carries **five** marks. (5×5=25)
- 13) Prove NAND and NOR gates as universal gates. 5
 - 14) Explain PIPO shift Register with a diagram. 5
 - 15) Discuss the Parity generator and Parity checker. 5
 - 16) Explain the operation of interrupt cycle with a flow chart. 5



- 17) Explain input-output instructions. 5
 18) Explain the three types of CPU organization. 5
 19) Explain the source initiated data transfer using handshaking with a block diagram and timing diagram. 5
 20) Write a note on memory hierarchy in a computer system. 5

SECTION – C

III. Answer **any three** questions. **Each** carries **fifteen** marks. (3×15=45)

- 21) a) Define K-Map ? Simplify the following Boolean function using K-Map : 8
 $F(A, B, C, D) = \Sigma(0, 2, 4, 6, 10, 11, 12, 13, 14, 15)$
 b) Explain different binary codes. 7
- 22) a) Define counter. With a neat diagram explain 4-bit synchronous binary counter. 8
 b) Explain octal to binary encoder with diagram. 7
- 23) Explain the design of basic computer with flow chart. 15
- 24) What is addressing mode ? Explain the different types of addressing modes with examples. 15
- 25) a) Explain DMA controller with a block diagram. 7
 b) Explain the working of associative memory. 8

SECTION – D

IV. Answer **any one** question. **Each** carries **ten** marks. (1×10=10)

- 26) a) Explain the working of full adder. 5
 b) Write a note on modes of data transfer. 5
- 27) a) Explain the common bus system. 5
 b) Write a note on RISC and CISC. 5
-

V Semester B.C.A. Degree Examination, Nov./Dec. 2017**(CBCS) (F + R) (2016-17 and Onwards)****BCA 503 : COMPUTER ARCHITECTURE**

Time : 3 Hours

Max. Marks : 100

Instruction : Answer all Sections.**SECTION – A****I. Answer any ten questions. Each carries two marks. (10×2=20)**

1) Write the symbol, logical expression and truth table of NAND gate.

2) Give the classification of integrated circuits.

3) Distinguish between RAM and ROM.

4) Define Multiplexer and Demultiplexer.

5) What are the types of binary codes ?

6) Subtract $24_{(10)}$ from $13_{(10)}$ using 2's complement method.

7) Define opcode and operand.

8) What is BUN instruction ?

9) What are the two types of computer architecture based on registers ?

10) What are the different types of interrupts ?

11) Define access time and transfer rate.

12) Define Baud rate.

SECTION – B**II. Answer any five questions. Each question carries five marks. (5×5=25)**

13) Explain the steps involved in the design of the sequential circuits.

14) Explain synchronous binary counter with logic diagram.

15) Discuss on error detection and correction codes briefly.

16) Explain any five register reference instructions.

17) With a block diagram, explain how BSA instruction executes.

18) Explain the addressing modes.

19) Explain DMA controller with a block diagram.

20) Write a note on virtual memory.


SECTION – C

III. Answer any three questions. Each question carries fifteen marks. $(3 \times 15 = 45)$

- 21) a) Simplify $F(ABCD) = \sum m(1, 3, 7, 11, 15) + \sum d(0, 2, 5)$ using K-map. 7
b) What is a half adder? Design a half adder using only NAND gates. 8
- 22) a) Explain decoder expansion with neat diagram. 7
b) Discuss the parity generator and parity checker. 8
- 23) a) Explain common bus organization of basic computer with neat diagram. 8
b) Distinguish between FGI and FGO. 7
- 24) a) What is a sub-routine? Explain CALL and RETURN instructions. 8
b) Explain the arithmetic logic shift with a neat diagram. 7
- 25) a) Explain I/O interface unit with a neat diagram. 8
b) Write a note on isolated vs memory mapped I/O. 7

SECTION – D

IV. Answer any one question. Question carries ten marks. $(1 \times 10 = 10)$

- 26) a) Explain 4-bit shift register. 5
b) Explain the working of J-K flip-flop. 5
- 27) a) Explain interrupt cycle with suitable example. 6
b) List the applications of EEPROM. 4

V Semester B.C.A. Degree Examination, Nov./Dec. 2018
(CBCS) (F + R)
(2016-17 and Onwards)
COMPUTER SCIENCE
BCA-503 : Computer Architecture

Time : 3 Hours

Max. Marks : 100

Instruction : Answer all Sections.**SECTION – A**

- I. Answer any ten questions : **(10×2=20)**
- 1) Explain Full adder.
 - 2) Define universal gates with logic circuit.
 - 3) Explain BSA instruction.
 - 4) State De-Morgan's theorem.
 - 5) Define Flip-Flop.
 - 6) Why we use shift register ?
 - 7) Explain Hamming code ?
 - 8) Define Indirect Address Mode.
 - 9) What is meant by Memory-Mapped I/O ?
 - 10) Define virtual memory.
 - 11) What is Parity bit ?
 - 12) Define types of RAM.

SECTION – B

- II. Answer any five questions : **(5×5=25)**
- 13) Explain the steps involved in design of combinational circuit.
 - 14) Write a note on program counter and stack memory.
 - 15) What is a Karnaugh Map ? Explain different types of Karnaugh Maps.
 - 16) Explain any five register reference instructions.



- 17) Write a note on Cache memory.
- 18) Compare CISC and RISC processors.
- 19) What are the important characteristics of memory ?
- 20) Explain timing signals.

SECTION – C

III. Answer **any three** questions. Each question carries **fifteen** marks. **(3x15=45)**

- | | |
|---|----|
| 21) Explain the types of program interrupts. | 10 |
| 22) a) Simplify $F(A, B, C, D) = \sum m(1, 2, 4, 6, 8, 10, 12, 14)$ and draw a circuit diagram. | 5 |
| b) What is a parity Bit ? Explain in brief. | 5 |
| 23) Explain types of CPU organization. | 6 |
| 24) a) Explain I/O commands. | 9 |
| b) Explain common BUS organization of a Basic computer. | 9 |
| 25) a) Explain Memory hierarchy. | 6 |
| b) Explain different Addressing Modes. | 9 |

SECTION – D

IV. Answer **any two** questions. **(1x10=10)**

- | | |
|---|---|
| 26) a) Explain direct Address and Indirect Address Modes. | 5 |
| b) Explain the working of R-S flip-flop. | 5 |
| 27) a) Explain 8 to 3 Encoder. | 5 |
| b) Discuss error detection and correction codes. | 5 |

BCA503T – COMPUTER ARCHITECTURE

2 MARKS QUESTIONS AND ANSWERS

1. State and prove Demorgan's Law.

1. The first theorem states that the compliment of the sum of Boolean expressions is equal to the product of the compliments of the individual expression.
$$(x+y)' = x'y'$$
 2. The second theorem states that the compliment of product of Boolean expressions is equal to sum of the compliment of the individual expressions.
$$(x.y)' = x'+y'$$

2. What is Decoder expansion?

At times we may require decoder of a certain size while only small size decoders are available. Then in order to obtain to require the size decoder, we have to combine two or more available size decoders.

3. Convert $(736.4)_8$ to decimal and binary.

$$\begin{aligned}
 (736.4)_8 &\text{ to decimal} \\
 &= 7*8^2 + 3*8^1 + 6*8^0 + 4*8^{-1} \\
 &= 7*64 + 3*8 + 6*1 + 4*0.125 \\
 &= 448 + 24 + 6 + 0.5 = 478.5_{10}
 \end{aligned}$$

$$(736.4)_8 = 478.5_{10}$$

$$\begin{array}{r}
 (736.4)_8 \text{ to Binary} \\
 = 7 \quad 3 \quad 6 \quad . \quad 4 \\
 111 \quad 011 \quad 110 \quad . \quad 100 \\
 (736.4)_8 \quad \equiv 111011110.100_2
 \end{array}$$

4. What is unidirectional and bidirectional shift register?

Unidirectional: A register capable of shifting in one direction only is called unidirectional shift register.

Bidirectional: A register that can shift in both directions is called Bidirectional shift register.

5. What is self-complementing code and weighted code ?

Weighted code :- Are those codes which obey the positional weighting principles. Each position of number represents the specific weight.

Self-complementing code :- It is an unnatural BCD code. Sum of weights of unnatural BCD codes is equal to 9. It is a self-complementing code. Self-complementing codes provide the 9's complement of a decimal number, just by interchanging 1's and 0's in its equivalent 2421 representation

6. What are the two types of control organization?

1. Hardwired control
 2. Microprogrammed control

7. Explain Full adder?

Full adder: Full adder is an arithmetic circuit block that can be used to add three bits to produce a SUM and CARRY output.

8. Define universal gates with logic circuits?

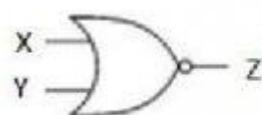
A universal gate is one of the logic gate which uses Boolean function {0,1}. NAND and NOR gates are called universal gates.

The circuit diagram and truth table of NAND gate:



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

The circuit table of NOR gate:



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

diagram and truth

9. Define types of RAM?

Types of RAM are:

1. **Static RAM (SRAM):** SRAM is that it retains data bits in its memory as long as power is being supplied.
2. **Dynamic RAM (DRAM):** DRAM is most common kind of RAM for personal computers and workstations.

10. Define Flip-Flop?

A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems.

11. Why we use shift register?

Shift registers are used for the storage or transfer of data in the form of binary numbers and “shifts” the data out once every clock cycle,hence the name shift register.

12. Explain Hamming code?

Hamming code is a set of error-correction codes that can be used to detect and correct bit errors that can occur when computer data is moved or stored.

13. What is computer architecture?

The architectural design of a computer system is concerned with the specifications of the various functional modules, such as processors and memories and structuring them together into a computer system.

14. What are the two types of computer architecture based on registers?

- 1.Von Neumann architecture
2. Harvard architecture

15. Mention the different families of IC.

- a. Bipolar families
- b. Metal Oxide Semiconductors(MOS) families

16. Distinguish between RAM and ROM.

RAM(Random access memory)	ROM(Read only memory)
In RAM the contents are lost when the computer is switched off.	ROM is a special type of memory which can be read and contents of which it are not lost even when the computer is switched off.
Read and Write operations are performed on RAM and they are classified as :- a. Static RAM b. Dynamic RAM	Only read operation are performed on ROM and they are classified as :- a. Masked programmed b. User programmed

17. What is parity bit?

A parity bit is an extra bit included with a binary message to make the total number of 1's either even or odd.

18. Write the BCD code of decimal number 8745.42₍₁₀₎?

Decimal Number = 8 7 4 5 . 4 2

BCD Code = 1000 0111 0100 0101 . 0100 0010

Therefore, 8745.42 = 1000011101000101.01000010_(BCD).

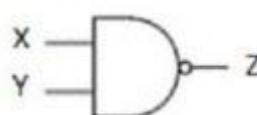
19. Define program counter?

The program counter (PC) holds the address of the next instruction to be read from the memory after the current instruction is executed.

20. Write the symbol, logical expression and truth table of NAND gate?

The Logical symbol and truth table

Logical Expression:
 $Z = (X \cdot Y)'$



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

21. Give a classification of integrated circuits?

1. Small Scale Integration (SSI)
2. Medium Scale Integration(MSI)
3. Large Scale Integration (LSI)
4. Very-Large Scale Integration(VLSI)
5. Super-Large Scale Integration(SLSI)
6. Ultra-Large Scale Integration(ULSI)

22. Define multiplexer and demultiplexer?

A multiplexer (MUX) is a device used to select a single line of input from multiple input lines using control signals. In this diagram, D0 to D3 are input data lines and Y is the output. The S0 and S1 bits tell the mux which one out of the 4 input lines will be selected as output. So, if S0 = 0 and S1=0, then Y= D0,

De-multiplexer(DMUX) is also a device with one input and multiple output lines. It is used to send a signal to one of the many devices. The main difference between a multiplexer and a de-multiplexer is that a multiplexer takes two or more signals and encodes them on a wire, whereas a de-multiplexer does reverse to what the multiplexer does.

23. what are the type of binary code?

1. Weighted Codes
2. Non-Weighted Codes
3. Alphanumeric Codes

24. Subtract 24 from 13 using 2's complement method?

$$\begin{array}{r} 00001101 \quad 13 \\ 11101000 \quad 2\text{'s complement of } (-24) \\ \hline 10001011 \quad \text{Result } (-11) \end{array}$$

25. Explain BSA instruction?

BSA: (Branch and save return address)

This instruction is useful for branching to a position of the program called a subroutine or producer. When executed, it stores the address of the next instruction in sequence into a memory location specified by the effective address.

26. What is BUN instruction?

BUN (Branch unconditionally) this instruction transfers the program to the instruction specifies by the effective address. The program counter PC holds the address of the instruction to be read from memory in the text instruction cycle.

27. Define opcode and operand.

The operation code of an instruction is a group of bits that define operations such as add, subtract, multiply, shift and complement.

Operand: An instruction code must not only specify the operation but also the registers or the memory words.

28. Define indirect Address Mode.

When the bits in the second part of the instruction designate an address of a memory word in which an address of the operand is found, it is called indirect address mode.

29. How many bits are needed to specify an address for a memory until of 4096 words?

For a memory unit with 4096 words, we need 12 bits to specify an address since $2^{12} = 4096$.

30. Mention the major components of CPU

- a. Control Unit
- b. Arithmetic Logical Unit
- c. Immediate Access

31. What is PSW?

PSW in electronics means Program Status Word. A register, which is 32 bits in size and holds all the information about the current state of an operation/program and hence, it helps in proper program execution.

32. What is an external interrupt? Give an example?

It is initiated by an external event. It is asynchronous with program (acts independent of program). It depends on external condition which is independent of program being executed at that point of time.

Eg:- Input /output devices

33. What are peripherals?

Input or output devices attached to the computer are also called Peripherals. Peripherals are Electromechanical and electromagnetic devices of some complexity.

34. What is memory management system?

Memory management is the process of controlling and coordinating the computer memory, assigning portions called blocks to various running programs to optimize overall system performance.

35. What is meant by Memory-mapped I/O?

Memory mapped I/O is a way to exchange data and instructions between a CPU and peripheral devices attached to it. Memory mapped IO is one where the processor and the IO device share the same memory location(memory),i.e.,the processor and IO devices are mapped using the memory address.

36. Define virtual memory?

A virtual memory system provides a mechanism for translating program generated addresses into correct main memory locations.

37. What is Polling?

The software method used to identify the highest priority source is called as Polling.in this there is one common branch address for all interrupts.

38. What are the different types of interrupts

1. External interrupts
2. Internal interrupts
3. Software interrupts

39. Define access time and transfer rate

Access time. The total time it takes the computer to read data from a storage device such as computer memory, hard drive, CD-ROM or other mechanism. Computer access time is commonly measured in nanoseconds or milliseconds and the lower the access the time the better.Data rates are often measured in megabits (million bits) or megabytes (million bytes) per second. These are usually abbreviated as Mbps and MBps,respectively. Another term for data transfer rate is throughput.

40. Define Baud rate

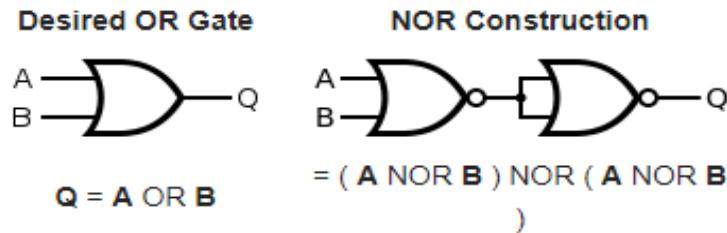
Baud rate represents the number of times per second a signal (changing from zero to one or one to zero) or symbol (the connection's voltage, frequency or phase) in a communications channel changes state or varies. For example, a 2,400 Baud rate means the channel is changing states up to 2,400 times per second.

LONG QUESTIONS WITH ANSWERS.

1. Prove NAND and NOR as universal gates.

NOR gate is a universal gate, meaning that any other gate can be represented as a combination of NOR gates.

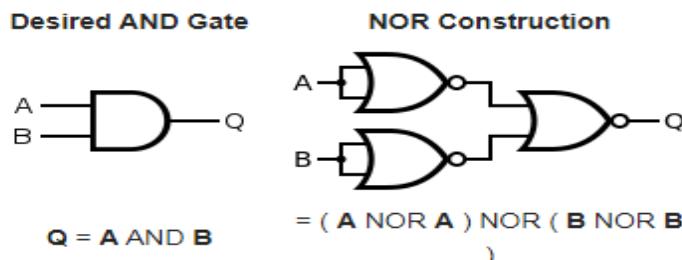
(a) Realizing NOT gate using NOR



Truth Table

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

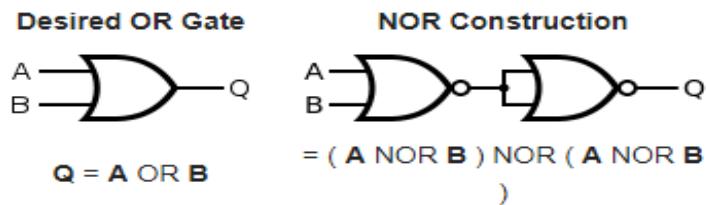
(b) Realizing AND gate using NOR



Truth Table

Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

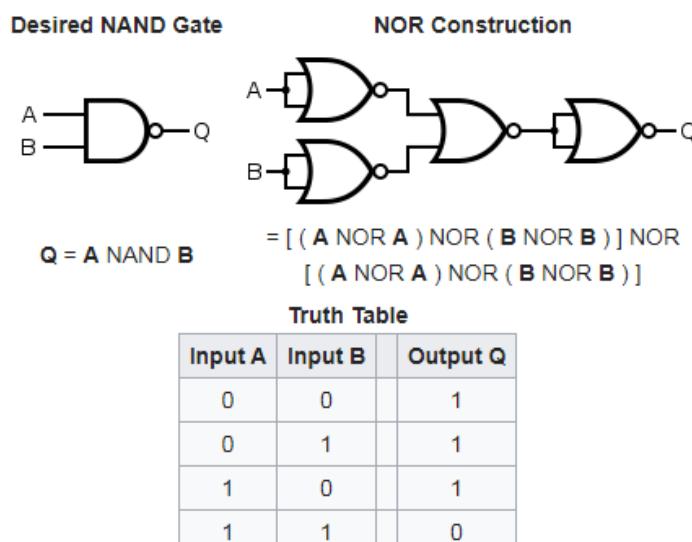
(c) Realizing OR gate using NOR



Truth Table

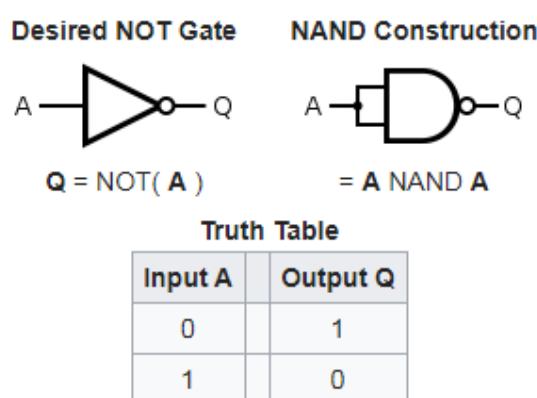
Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

(d) Realizing NAND gate using NOR



A NAND gate is a universal gate, meaning that any other gate can be represented as a combination of NAND gates.

(A) Realizing NOT gate using NAND



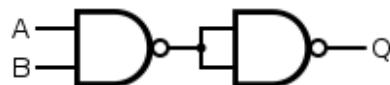
(B) Realizing AND gate using NAND

Desired AND Gate



$$Q = A \text{ AND } B$$

NAND Construction



$$= (\text{A NAND B}) \text{ NAND } (\text{A NAND B})$$

Truth Table

Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

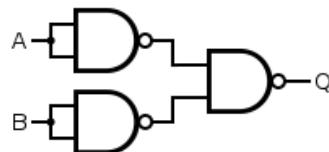
(C) Realizing OR gate using NAND

Desired OR Gate



$$Q = A \text{ OR } B$$

NAND Construction

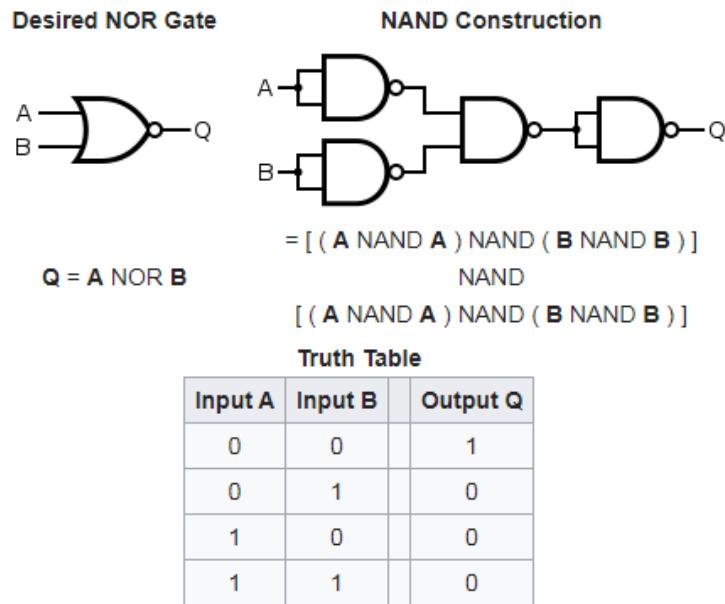


$$= (\text{A NAND A}) \text{ NAND } (\text{B NAND B})$$

Truth Table

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

(D) Realizing NOR gate using NAND



2. Explain the steps involved in design of combinational circuit.

A combinational circuit is a connection of logic gates with a set of inputs and outputs. The design of a combinational circuit starts from the outline of the problem and ends in a logic circuit diagram.

- The problem is stated.
- The input and output variables are assigned letter symbols.
- The truth table that defines the relationship between inputs and outputs is derived.
- The simplified Boolean functions for each output are obtained.
- The logic diagram is drawn.

The design of combination circuits can be demonstrated with two simple examples of arithmetic circuits the Half Adder and the Full Adder.

3. What is K-map? Explain with an example.

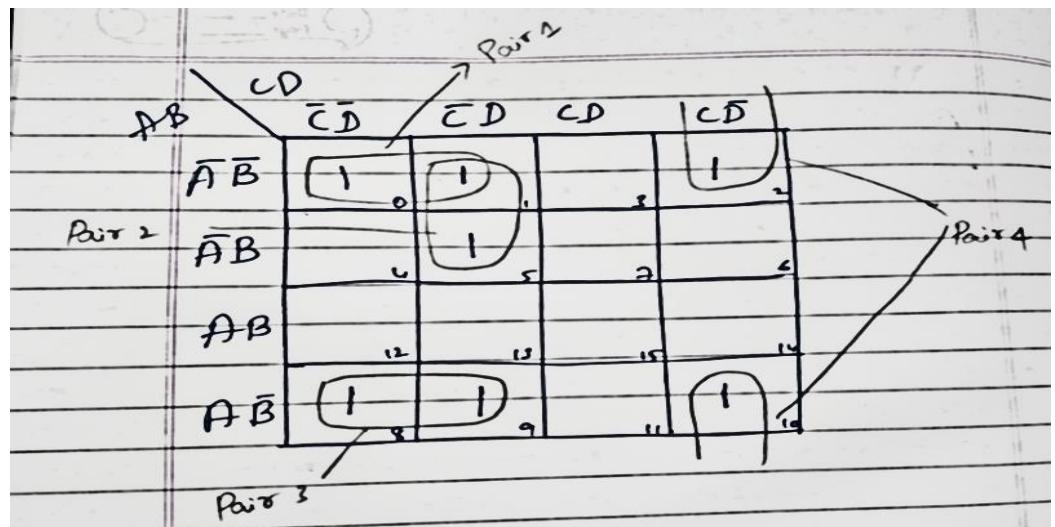
Karnaugh map is also called as K Map. The Karnaugh map is a method of simplifying Boolean algebra expressions. There are three types of K map are

- 2-variables map ($2^2=4$ cells)
- 3-variables map ($2^3=8$ cells)
- 4 variables map ($2^4=16$ cells)

The K-map is used to simplify the complex expression into simplex using different groping. Eg., Pair, Quad, Hexa, Overlapping, Rolling etc.,

Ex:

Boolean function $F(A,B,C,D) = \Sigma (0,1,2,5,8,9,10)$ in sum-of-products.



$$\begin{aligned} \text{Pair 1: } & A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} \\ & \bar{A}\bar{B} \in (\bar{D} + \bar{D}) \\ & = \bar{A}\bar{B}C \end{aligned}$$

$$\begin{aligned} \text{Pair 2: } & \bar{A}\bar{B}\bar{C}D + A\bar{B}C\bar{D} \\ & \bar{A}\bar{C} \in D(B + \bar{B}) \\ & = \bar{A}\bar{C}D \end{aligned}$$

$$\begin{aligned} \text{Pair 3: } & A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D \\ & A\bar{B}\bar{C} \in (\bar{D} + \bar{D}) \\ & = A\bar{B}\bar{C} \end{aligned}$$

$$\begin{aligned} \text{Pair 4: } & \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} \\ & \bar{B}C\bar{D} \in (A + \bar{A}) \\ & = \bar{B}C\bar{D} \end{aligned}$$

$$\Rightarrow \bar{A}\bar{B}C + \bar{A}\bar{C}D + A\bar{B}\bar{C} + \bar{B}C\bar{D}$$

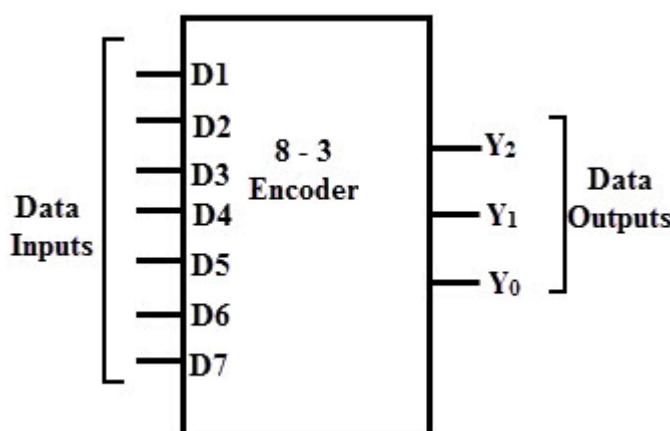


Scanned with
CamScanner

4. Explain octal to binary encoder with a diagram.

An octal to binary encoder consists of eight input lines and three output lines. Each input line corresponds to each octal digit and three outputs generate corresponding binary code.

In encoders, it is to be assumed that only one input is active or has a value 1 at any given time otherwise the circuit has no meaning. The figure below shows the logic symbol of octal to binary encoder along with its truth table.



No	Inputs									Outputs		
	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Y ₂	Y ₁	Y ₀	
0	0	0	0	0	0	0	0	1	0	0	0	
1	0	0	0	0	0	0	1	0	0	0	1	
2	0	0	0	0	0	1	0	0	0	1	0	
3	0	0	0	0	1	0	0	0	0	1	1	
4	0	0	0	1	0	0	0	0	1	0	0	
5	0	0	1	0	0	0	0	0	1	0	1	
6	0	1	0	0	0	0	0	0	0	1	1	
7	1	0	0	0	0	0	0	0	0	1	1	

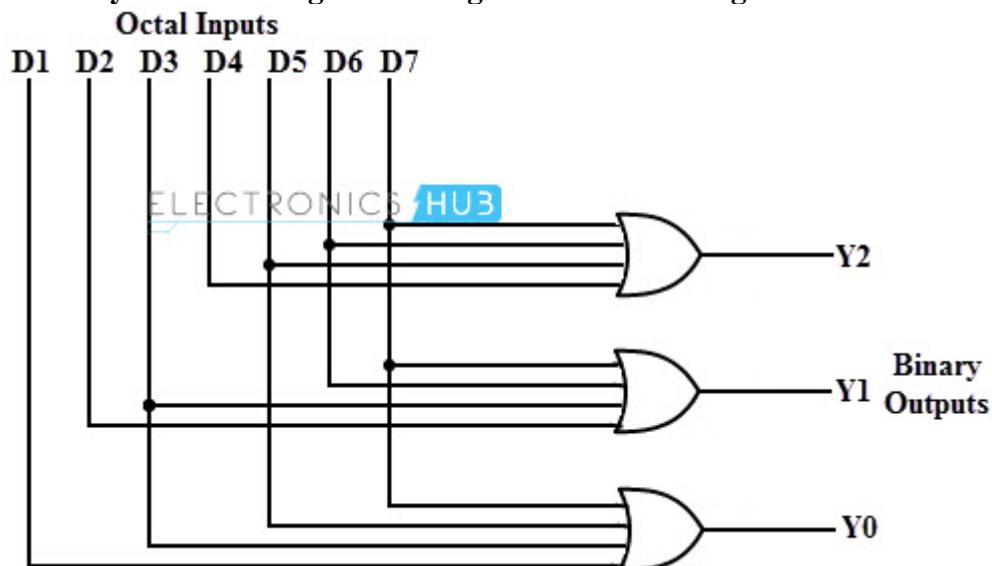
From the above table, the output Y₂ becomes 1 if any of the digits D₄ or D₅ or D₆ or D₇ is one. Thus, we can write its expression as

$$Y_2 = D_4 + D_5 + D_6 + D_7$$

$$\text{Similarly, } Y_1 = D_2 + D_3 + D_6 + D_7 \text{ and}$$

$$Y_0 = D_1 + D_3 + D_5 + D_7$$

Also it is to be observed that D₀ does not exist in any of the expressions so it is considered as don't care. From the above expressions, we can implement the octal to binary encoder using set of OR gates as shown in figure below



There is ambiguity in the octal to binary encoder that when all the inputs are zero, an output with all 0's is generated. Also, when D₀ is 1, the output generated is zero. This is a major problem in this type of encoder. This can be resolved by specifying the condition that none of the inputs are active with an additional output

5. What is half adder? Design a half adder using only NAND gates

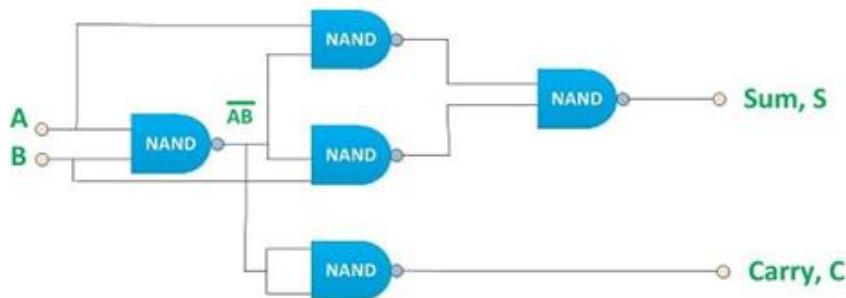
Half Adder is the digital circuit which can generate the result of the addition of two 1-bit numbers. It consists of two input terminals through which 1-bit numbers can

be given for processing. After this, the half adder generates the sum of the numbers and carry if present.



Half Adder using NAND Gates

The half adder can also be designed with the help of NAND gates. NAND gate is considered as a universal gate. A universal gate can be used for designing of any digital circuitry. It is always simple and efficient to use the minimum number of gates in the designing process of our circuit. The minimum number of NAND gates required to design half adder is 5.



The first NAND gate takes the inputs which are the two 1-bit numbers. The resultant NAND operated inputs will be again given as input to 3- NAND gates along with the original input. Out of these 3 NAND gates, 2-NAND gates will generate the output which will be given as input to the NAND gate connected at the end. The gate connected at the end will generate the sum bit. Out of the 3 considered NAND gates, the third NAND gate will generate the carry bit.

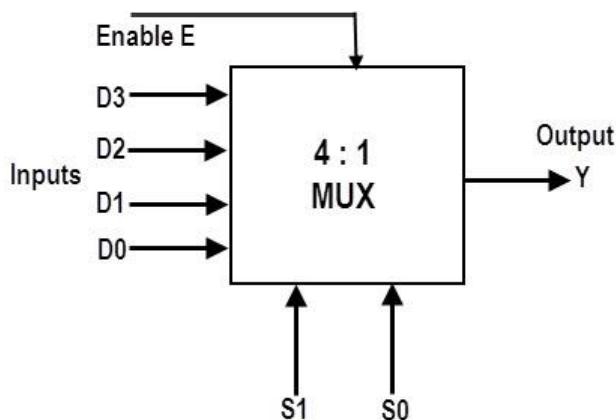
6. Design a 4-to-1 multiplexer

It is a logic circuit that switches digital data from several input lines onto a single output line in a specified time sequence.

A multiplexer has several data input lines and a single output line. It also has data select inputs which permit digital data on only one of the input to be switched to the output lines.

4-to-1 Multiplexer

- A 4-to-1 multiplexer consists four data input lines as D0 to D3, two select lines as S0 and S1 and a single output line Y.
- The select lines S1 and S2 select one of the four input lines to connect the output line.
- The particular input combination on select lines selects one of input (D0 through D3) to the output.
- The figure below shows the block diagram of a 4-to-1 multiplexer in which the multiplexer decodes the input through select line.



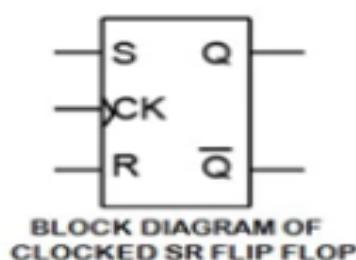
Truth table :

Select Data Inputs		Output
S ₁	S ₀	Y
0	0	D ₀
0	1	D ₁
1	0	D ₂
1	1	D ₃

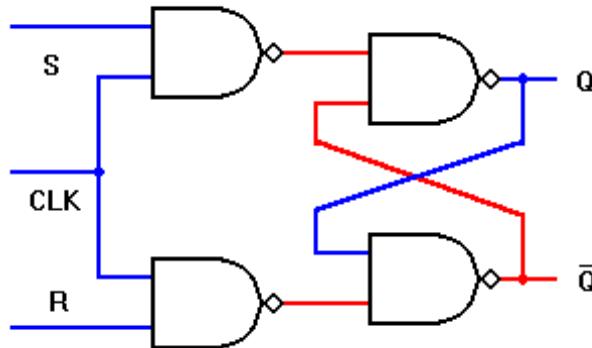
- If S₁=0 and S₀=0 then Y = D₀
 - If S₁= 0 and S₀=1, the Y = D₁
 - If S₁=1 and S₀=0, then Y = D₂
 - If S₁=1 and S₀=1 the Y = D₃
- Therefore, Y = D₃ S₁ S₀

7. Explain the working of R-S flip-flop.

In clocked R-S flip flop the appropriate levels applied to their inputs are blocked till the receipt of pulse from an other source called clock. The flip flop changes state only when clock pulse is applied depending upon the inputs.



The basic circuit is shown below.



INPUTS			OUTPUTS		Mode of Operation	Effect on output X
CLK	S	R	X	\bar{X}		
$\text{--}^{\text{--}}$	0	0	X	\bar{X}	Hold or Idle	No change
$\text{--}^{\text{--}}$	0	1	0	1	Reset	Reset to 0
$\text{--}^{\text{--}}$	1	0	1	0	Set	Set to 1
$\text{--}^{\text{--}}$	1	1	1	1	Forbidden or Prohibited	Should not be used

- With inputs S=0 and R=0, the clock pulse has no effect on output X. The flip flop is in the idle or hold mode.
- With inputs S=0 and R=1, when the clock pulse is applied, the active high signal on R resets or clears the flip flop to 0. Then flip flop is said to be in reset mode.
- With inputs S=1 and R=0, when the clock pulse is applied, the active high signal on S sets the flip flop to 1. Then flip flop is said to be in set mode.
- With inputs S=1 and R=1, when the clock pulse is applied, the flip flop to 0. The flip flop enters the prohibited or forbidden state. This state cannot be used.

8. Explain 8 to 3 Encoder

The 8 to 3 Encoder or octal to Binary encoder consists of 8 inputs : Y7 to Y0 and 3 outputs : A2, A1 & A0. Each input line corresponds to each octal digit and three outputs generate corresponding binary code.

The figure below shows the logic symbol of octal to binary encoder:



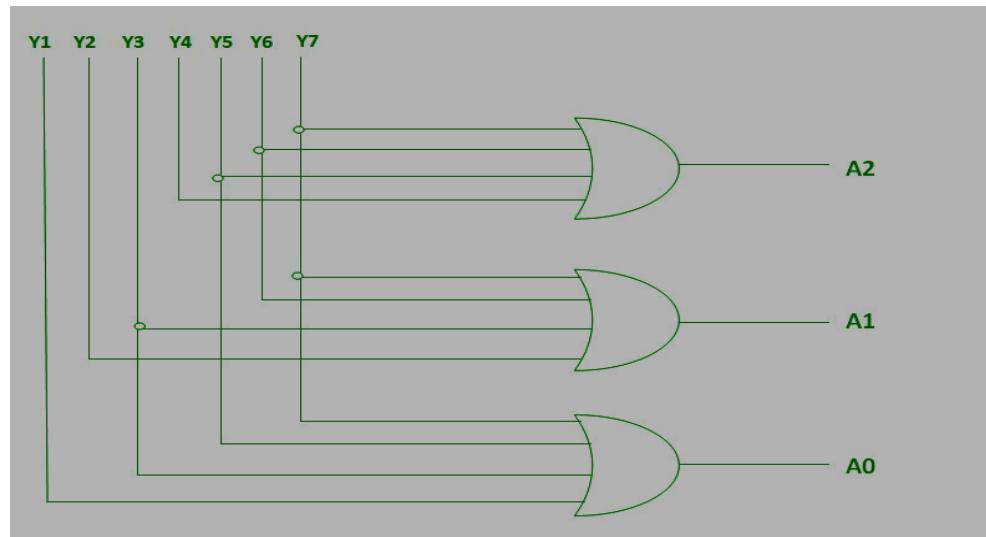
The truth table for 8 to 3 encoder is as follows :

INPUTS								OUTPUTS		
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	A2	A1	A0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Logical expression for A2, A1 and A0 :

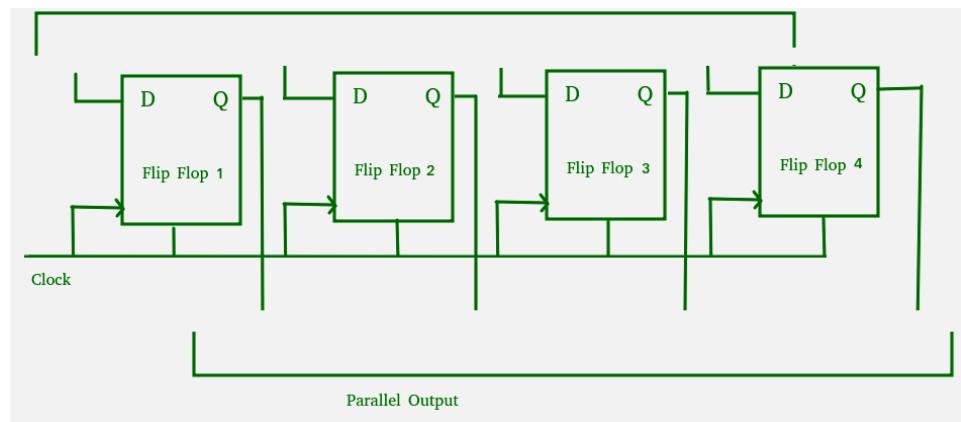
$$\begin{aligned} A2 &= Y7 + Y6 + Y5 + Y4 \\ A1 &= Y7 + Y6 + Y3 + Y2 \\ A0 &= Y7 + Y5 + Y3 + Y1 \end{aligned}$$

The above two Boolean functions A2, A1 and A0 can be implemented using four input OR gates :



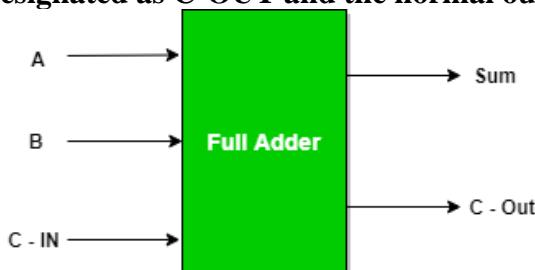
9. Explain PIPO shift register with a diagram.

- The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and also produces a parallel output is known as Parallel-In parallel-Out shift register.
 - The circuit consists of four D flip-flops which are connected. The clear (CLR) signal and clock signals are connected to all the 4 flip flops. In this type of register, there are no interconnections between the individual flip-flops since no serial shifting of the data is required. Data is given as input separately for each flip flop and in the same way, output also collected individually from each flip flop.



10. Explain the working of full adder.

Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.



Truth Table :

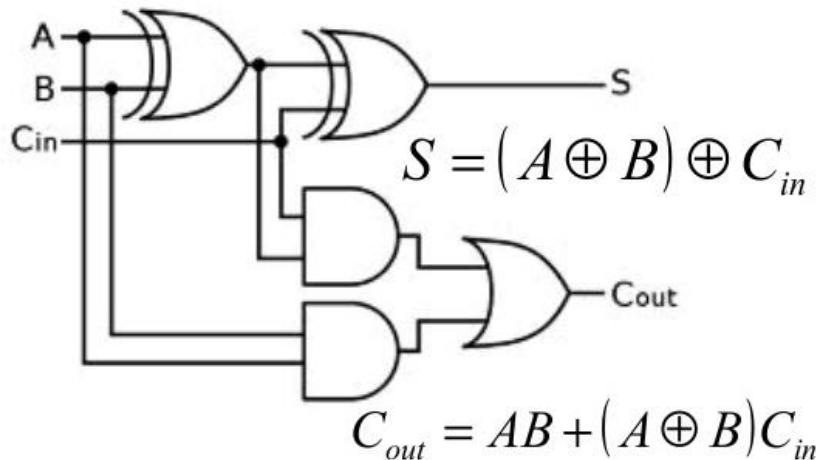
Inputs			Outputs	
A	B	C-IN	Sum	C-OUT
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The logic diagram for Full Adder can be developed from the 2 logical expressions for S (sum) Cout Carry.

$$S = AB'C_{in} + A'BC'_{in} + ABC'_{in} + ABC_{in}$$

$$C_{out} = A'BC_{in} + AB'C_{in} + ABC'_{in} + ABC_{in}$$

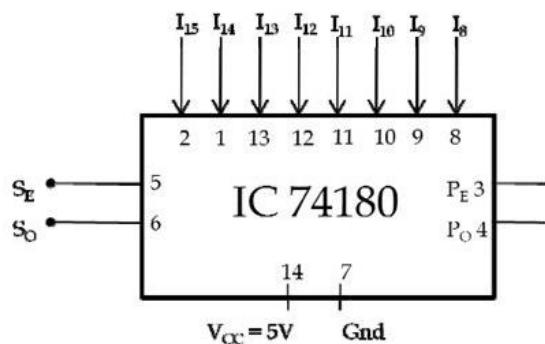
- Full adder logic diagram



11. Explain Parity checker and generator.

PARITY CHECKER

- When this device is used as even parity checker, the number of input bits should always be even. When a parity error occurs, the \sum even output goes low and the \sum odd goes high.
- When it is used as an odd parity checker, the number of input bits should always be odd. When a parity error occurs, the \sum odd output gets high.



Number of Inputs HIGH	OUTPUTS	
	\sum Even	\sum Odd
0,2,4,6,8	HIGH	LOW
1,3,5,7,9	LOW	HIGH

PARITY GENERATOR

- When this device is used as an even parity generator, the parity bit is taken at the odd output because this output is a 0 if there is an even number of input bits, and it is a 1 if there is an odd number. When used an odd parity generator, the parity bit is taken at the a 0 even output because it is a 0 when the number of inputs is odd.

12. Explain error detection and correction code.

- Error detection codes :- are used to detect the error(s) present in the received data (bit stream). These codes contain some bit(s), which are included (appended) to the original bit stream. These codes detect the error, if it is occurred during transmission of the original data (bit stream). Example – Parity code, Hamming code.
- Error correction codes :- are used to correct the error(s) present in the received data (bit stream) so that, we will get the original data. Error correction codes also use the similar strategy of error detection codes. Example – Hamming code.

Therefore, to detect and correct the errors, additional bit(s) are appended to the data bits at the time of transmission.

- **Parity Code:**-It is easy to include (append) one parity bit either to the left of MSB or to the right of LSB of original bit stream. There are two types of parity codes, namely even parity code and odd parity code based on the type of parity being chosen.
- **Even Parity Code:**-The value of even parity bit should be zero, if even number of ones present in the binary code. Otherwise, it should be one. So that, even number of ones present in even parity code. Even parity code contains the data bits and even parity bit.

The following table shows the even parity codes corresponding to each 3-bit binary code. Here, the even parity bit is included to the right of LSB of binary code.

Binary Code	Even Parity bit	Even Parity Code
000	0	0000
001	1	0011
010	1	0101
011	0	0110
100	1	1001
101	0	1010
110	0	1100
111	1	1111

- **Odd Parity Code** The value of odd parity bit should be zero, if odd number of ones present in the binary code. Otherwise, it should be one. So that, odd number of ones present in odd parity code. Odd parity code contains the data bits and odd parity bit.

The following table shows the odd parity codes corresponding to each 3-bit binary code. Here, the odd parity bit is included to the right of LSB of binary code.

13. What is a parity Bit? Explain in brief.

A parity bit is an extra bit included with a binary message to make the total number of 1s either odd or even.

EVEN	Parity	ODD	Parity
P	BCD	P	BCD
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001

1. The parity bit can be attached to the code at the beginning or the end, depending on how the system is designed.
2. At the sending end, the message is applied to a parity generator, where the required parity bit is generated.
3. The message, including the parity bit, is transmitted to its destination.
4. At the receiving end, all the incoming bits are applied to a parity checker that checks the proper parity adopted (odd or even).
5. If the checked parity does not conform to the adopted parity, an error is detected.

14. Explain the Common Bus system.

- The basic computer has eight registers, a memory unit, and a control unit . Paths must be provided to transfer information from one register to another and between memory and registers
- The number of wires will be excessive if connections are made between the outputs of each register and the inputs of the other registers.
- A more efficient scheme for transferring information in a system with many registers is to use a common bus.
- The connection of the registers and memory of the basic computer to a common bus system is shown in Fig. below. The outputs of seven registers and memory are connected to the common bus.

The specific output that is selected for the bus lines at any given time is determined from the binary value of the selection variables S_2 , S_1 , and S_0 .

- The number along each output shows the decimal equivalent of the required binary selection. For example, the number along the output of DR is 3.
- The 16-bit outputs of DR are placed on the bus lines when $S_2S_1S_0 = 011$ since this is the binary value of decimal 3.
- The lines from the common bus are connected to the inputs of each register and the data inputs of the memory. The particular register whose LD (load) input is enabled receives the data from the bus during the next clock pulse transition.

- The memory receives the contents of the bus when its write input is activated. The memory places its 16-bit output onto the bus when the read input is activated and $S_2S_1S_0 = 111$.
- Four registers, DR, AC, IR, and TR, have 16 bits each. Two registers, AR

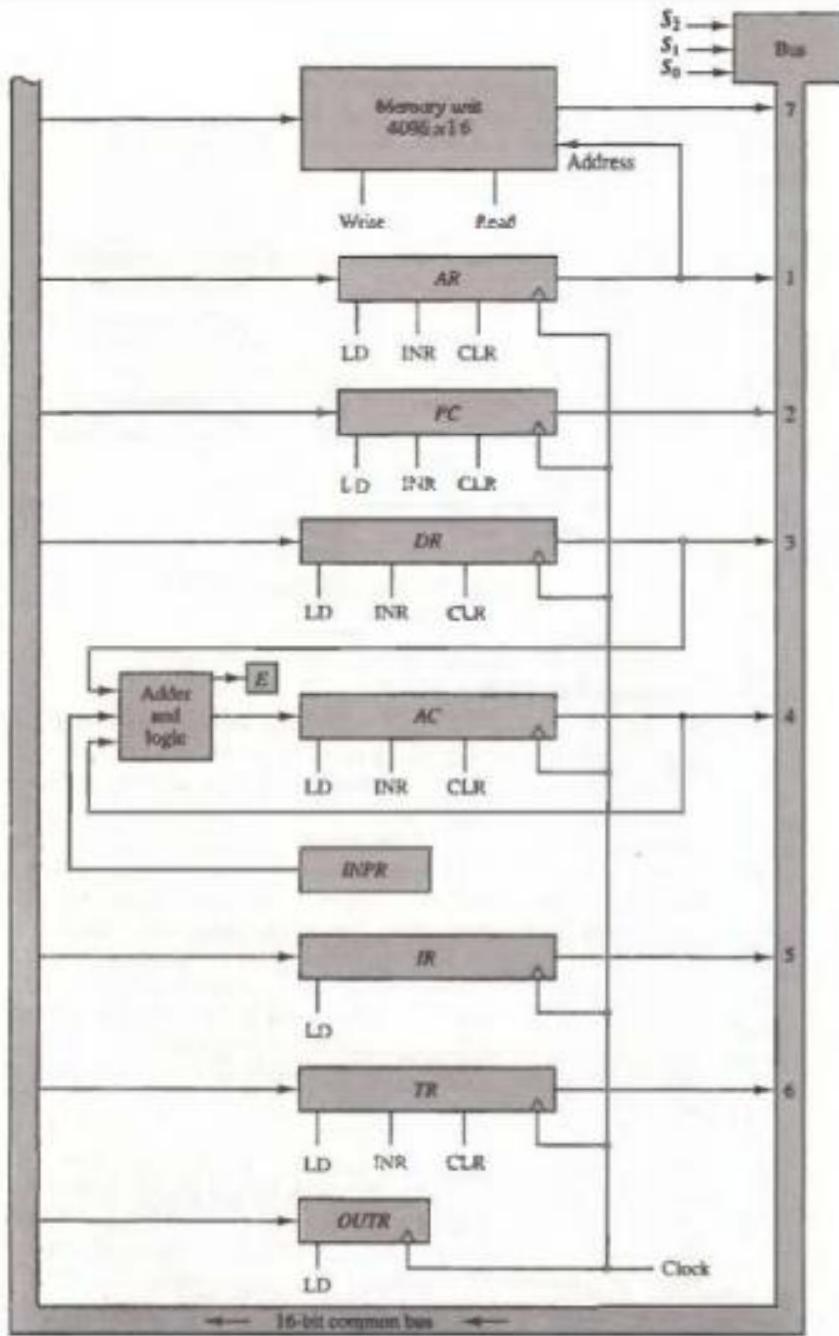


Figure 5-4 Basic computer registers connected to a common bus.

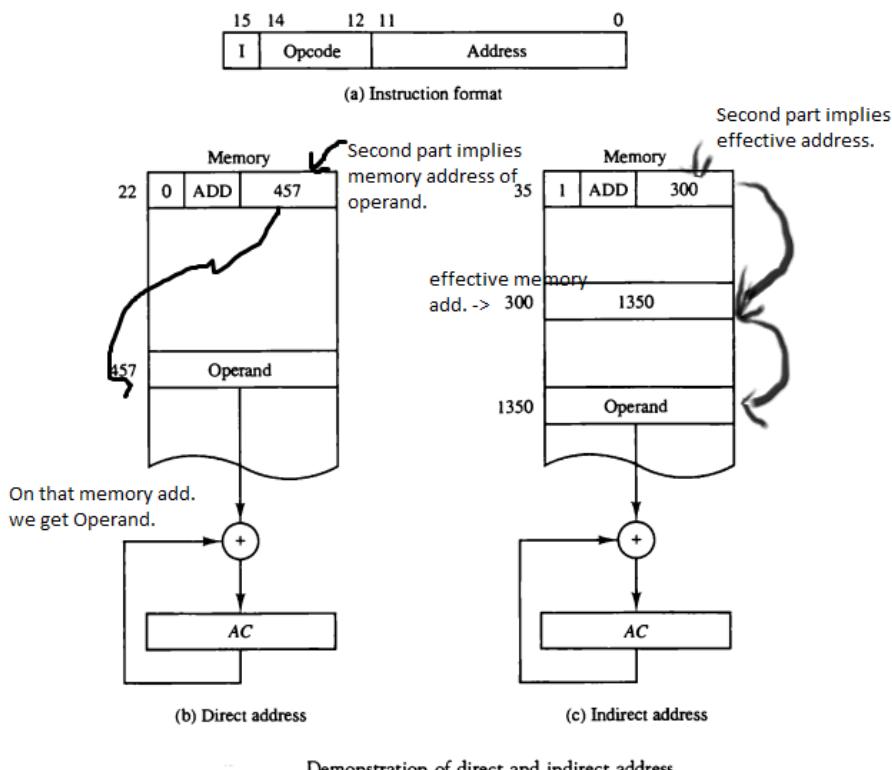
- and PC, have 12 bits each since they hold a memory address. When the contents of AR or PC are applied to the 16-bit common bus, the four most significant bits are set to 0's.
- When AR or PC receive information from the bus, only the 12 least significant bits are transferred into the register. The input register INPR and the output register OUTR have 8 bits each and communicate with the eight least significant bits in the bus.

- INPR is connected to provide information to the bus but OUTR can only receive information from the bus.
- This is because INPR receives a character from an input device which is then transferred to AC. OUTR receives a character from AC and delivers it to an output device. There is no transfer from OUTR to any of the other registers.
- The 16 lines of the common bus receive information from six registers and the memory unit. The bus lines are connected to the inputs of six registers and the memory. Five registers have three control inputs: LD (load), INR (increment), and CLR (clear).
- This type of register is equivalent to a binary counter with parallel load and synchronous clear. The increment operation is achieved by enabling the count input of the counter. Two registers have only a LD input.
- The input data and output data of the memory are connected to the common bus, but the memory address is connected to AR. Therefore, AR must always be used to specify a memory address.
- By using a single register for the address, we eliminate the need for an address bus that would have been needed otherwise. The content of any register can be specified for the memory data input during a write operation. Similarly, any register can receive the data from memory after a read operation except AC .
- The 16 inputs of AC come from an adder and logic circuit. This circuit has three sets of inputs. One set of 16-bit inputs come from the outputs of AC . They are used to implement register micro operations such as complement AC and shift AC .
- Another set of 16-bit inputs come from the data register DR. The inputs from DR and AC are used for arithmetic and logic rnlcro operations, such as add DR to AC or AND DR to AC.
- The result of an addition is transferred to AC and the end carry-out of the addition is transferred to flip-flop E (extended AC bit). A third set of 8-bit inputs come from the input register INPR.
- Note that the content of any register can be applied onto the bus and an operation can be performed in the adder and logic circuit during the same clock cycle. The clock transition at the end of the cycle transfers the content of the bus into the designated destination register and the output of the adder and logic circuit into AC.

15. Explain direct and indirect addressing mode with an example.

Direct address mode:

The effective address is equal to the address part of the instruction. The operand resides in memory and its address is given directly by the address field of the instruction. In a branch type instruction, the address field specifies the actual branch address.



It consists of a 3bit opcode, a 12 bit address and a mode bit 1 which is 0 for direct address. A direct address instruction is placed is address 22 in memory.

Indirect Address mode:

The address field of the instruction gives the address where the effective address is stored in memory. Control fetches the instruction from memory and uses its address part to access memory again to read the effective address. One bit of the instruction code can be used to distinguish between a direct and an indirect address.

The instruction is placed in address 35. The mode bit 1 and so is indirect address. The address part is binary of 300. The control goes to address 300 to find the address of the operand. The operand found in address 1350 is then added to the content of AC.

16. Explain any five Memory reference instruction.

Memory Reference – These instructions refer to memory address as an operand. The other operand is always accumulator. Specifies 12-bit address, 3-bit opcode (other than 111) and 1-bit addressing mode for direct and indirect addressing.



The set of instructions incorporated in 16 bit IR register are:

1. Arithmetic, logical and shift instructions (and, add, complement, circulate left, right, etc)
2. To move information to and from memory (store the accumulator, load the accumulator)
3. Program control instructions with status conditions (branch, skip)
4. Input output instructions (input character, output character)

Symbol	Hexadecimal Code		Description
AND	0xxx	8xxx	And memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load memory word to AC
STA	3xxx	Bxxx	Store AC content in memory
BUN	4xxx	Cxxx	Branch Unconditionally
BSA	5xxx	Dxxx	Add memory word to AC
ISZ	6xxx	Exxx	Increment and skip if 0

17. What is addressing mode? Explain the different types of addressing modes.

The addressing mode gives or indicates a rule to identify the operands location. Computers use addressing mode techniques for the purpose of accommodating the following provisions.

- To give programming versatility to the user.
- To reduce the number of bits in the address field of the instruction.
- To provide flexibility for writing programs.

The various addressing modes available are:

- **Implied mode:**

In this mode the operands are specified implicitly in the definition of the instruction. All register reference instructions that use an accumulator are implied mode instruction.

Ex: CMA

- **Immediate mode:**

The purpose of an address is to identify an operand value to be used in executing the instruction. Sometimes the operand values are contained in the instruction itself, this mode of operand specification is called immediate addressing mode.

Ex: MVI A, 45

- **Register mode:**

In this mode the operand are in registers which reside within the CPU. The register is selected from the register field in the instruction.

Ex: MOV AX, BX

- **Register indirect mode:**

In this instruction, the address field specifies a processor register in the CPU whose contents give the address of the operand in memory.

Ex: LXI H E000 ; memory address placed in processor register.

18. Explain three types of CPU organization.

There are three types of CPU organizations are:

- Single accumulator organization.
- General register organization.
- Stack organization.

- a. Single accumulator organization: All operations are performed with an implied accumulator register. The instruction format uses one address field.

ADD X

Where X is the address of the operand. This causes
 $AC \leftarrow AC + M[X]$.

The accumulator contents are added to the memory location content whose address is X and the result is stored in the accumulator.

- b. General register organization: General register type computers employ two or three address fields in their instruction format. Each address field may specify a processor register or a memory word.

ADD R1, X

This operation specifies $R1 \leftarrow R1 + M[X]$

It has two address fields, one for register R1 and the other for the memory address X.

- c. Stack organization: Computers with stack organization have PUSH and POP instructions which require an address field.

PUSH X

The instruction will push the word at address X to the top of the stack.

19. Differentiate between CISC and RISC.

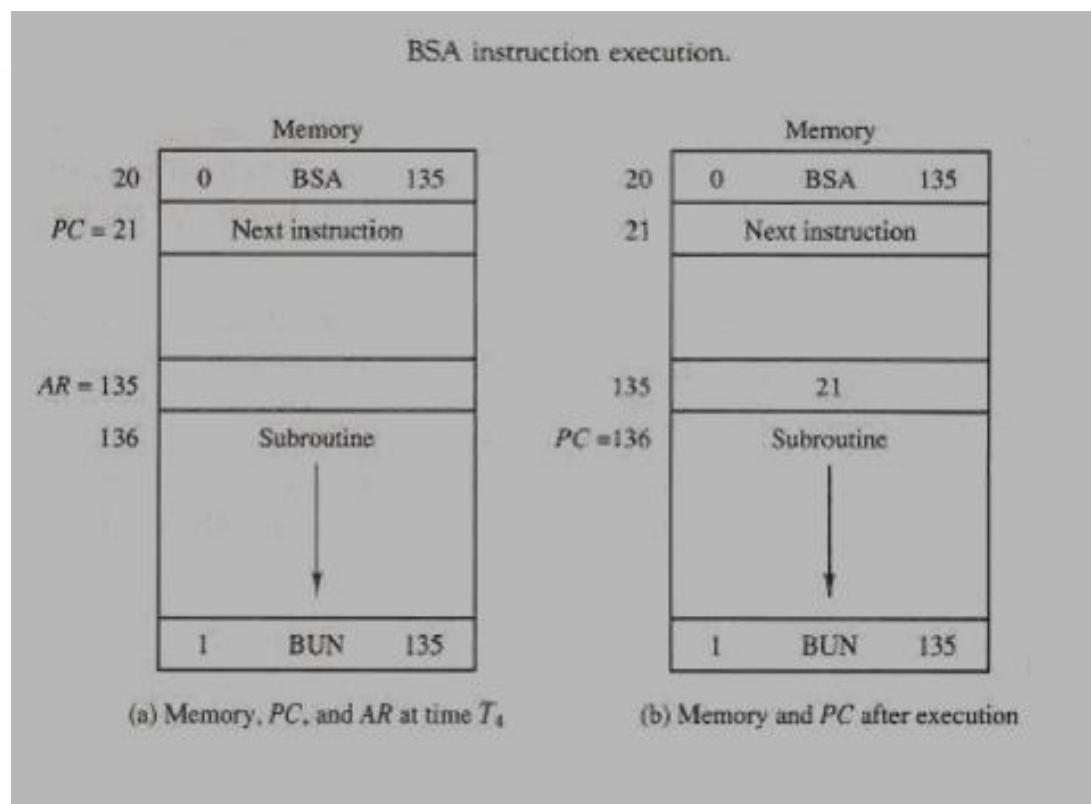
CISC	RISC
1. Large number of instructions	1. Fewer instructions
2. Emphasis is on hardware	2. Emphasis is on software
3. It includes multi-clock complex instructions	3. It includes single-clock, reduced instruction only
4. Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	4. Register to register: "LOAD" and "STORE" are independent instructions
5. Code size is small but complex. High cycles per second	5. Code size is large but simple. Low cycles per second
6. Variable length instruction format	6. Fixed length instruction format
7. Large variety of addressing modes	7. Few addressing modes

20. With a block diagram explain how BSA instruction executes .

Branch and Save Return Address

- This instruction is useful for branching to a portion of the program called a subroutine or procedure.
- When executed, the BSA instruction stores the address of the next instruction in sequence (which is available in PC) into a memory location specified by the effective address.
- The effective address plus one is then transferred to *PC* to serve as the address of the first instruction in the subroutine.
- This operation was specified with the following register transfer:
A numerical example that demonstrates how this instruction is used with subroutine is shown in Fig.

$$M[AR] \leftarrow PC, \quad PC \leftarrow AR + 1$$



- The BSA instruction is assumed to be in memory at address 20.
- The I bit is 0 and the address part of the instruction has the binary equivalent of 135.
- After the fetch and decode phases, *PC* contains 21, which is the address of the next instruction in the program (referred to as the return *address*). *AR* holds the effective address 135.

- This is shown in part (a) of the figure.
- The BSA instruction performs the following numerical operation:
- The result of this operation is shown in part (b) of the figure.
- The return address 21 is stored in memory location 135 and control continues with the subroutine program starting from address 136.
- The return to the original program (at address 21) is accomplished by means of an indirect BUN instruction placed at the end of the subroutine.

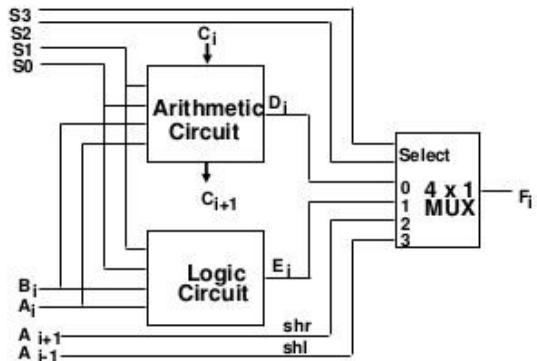
21. Explain sub-routine? Explain CALL and RETURN instructions?

- A set of Instructions which are used repeatedly in a program can be referred to as Subroutine. Only one copy of this Instruction is stored in the memory. When a Subroutine is required it can be called many times during the Execution of a Particular program. A *call Subroutine Instruction* calls the Subroutine. Care Should be taken while returning a Subroutine as Subroutine can be called from a different place from the memory.
- **Unconditional Return instruction:** RET is the instruction used to mark the end of sub-routine. It has no parameter. After execution of this instruction program control is transferred back to main program from where it had stopped. Value of PC (Program Counter) is retrieved from the memory stack and value of SP (Stack Pointer) is incremented by 2.
- **Conditional Return instruction** –By these instructions program control is transferred back to main program and value of PC is popped from stack only if condition is satisfied. There is no parameter for return instruction.
Unconditional Call instruction –CALL address is the format for unconditional call instruction. After execution of this instruction program control is transferred to a sub-routine whose starting address is specified in the instruction. Value of PC (Program Counter) is transferred to the memory stack and value of SP (Stack Pointer) is decremented by 2.
- **Conditional Call instruction** –In these instructions program control is transferred to subroutine and value of PC is pushed into stack only if condition is satisfied.

22. Explain the arithmetic logic shift with neat diagram

The arithmetic, logic, and shift circuits can be combined into one ALU with common selection variables. One stage of an arithmetic logic shift unit is shown in Fig. 4-13. The subscript i designates a typical stage. Inputs A1 and B1 are applied to both the arithmetic and logic units.

ARITHMETIC LOGIC SHIFT UNIT



S_3	S_2	S_1	S_0	Cin	Operation	Function
0	0	0	0	0	$F = A$	Transfer A
0	0	0	0	1	$F = A + 1$	Increment A
0	0	0	1	0	$F = A + B$	Addition
0	0	0	1	1	$F = A + B + 1$	Add with carry
0	0	1	0	0	$F = A + B'$	Subtract with borrow
0	0	1	0	1	$F = A + B' + 1$	Subtraction
0	0	1	1	0	$F = A - 1$	Decrement A
0	0	1	1	1	$F = A$	TransferA
0	1	0	0	X	$F = A \wedge B$	AND
0	1	0	1	X	$F = A \vee B$	OR
0	1	1	0	X	$F = A \oplus B$	XOR
0	1	1	1	X	$F = A'$	Complement A
1	0	X	X	X	$F = shr A$	Shift right A into F
1	1	X	X	X	$F = shl A$	Shift left A into F

- A particular microoperation is selected with inputs S_1 and S_0 . A 4×1 multiplexer at the output chooses between an arithmetic output in E_i and a logic output in H_i . The data in the multiplexer are selected with inputs S_3 and S_2 . The other two data inputs to the multiplexer receive inputs A_{i-1} for the shift-right operation and A_{i+1} for the shift-left operation. Note that the diagram shows just one typical stage. The circuit of Fig. 4-13 must be repeated n times for an n -bit ALU. The output carry C_{i+1} of a given arithmetic stage must be connected to the input carry C_i of the next stage in sequence. The input carry to the first stage is the input carry C_{in} , which provides a selection variable for the arithmetic operations.
- The circuit whose one stage is specified in Fig. 4-13 provides eight arithmetic operations, four logic operations, and two shift operations. Each operation is selected with the five variables S_3, S_2, S_1, S_0 , and C_{in} . The input carry C_{in} is used for selecting an arithmetic operation only.
- Table 4-B lists the 14 operations of the ALU. The first eight are arithmetic operations and are selected with $S_3S_2 = 00$. The next four are logic operations and are selected with $S_3S_2 = 01$. The input carry has no effect during the logic operations and is marked with don't-care x's. The last two operations are shift operations and are selected with $S_3S_2 = 10$ and 11 . The other three selection inputs have no effect on the shift.

23. With a block diagram explain how BSA instruction executes

Branch and Save Return Address

This instruction is useful for branching to a portion of the program called a subroutine or procedure.

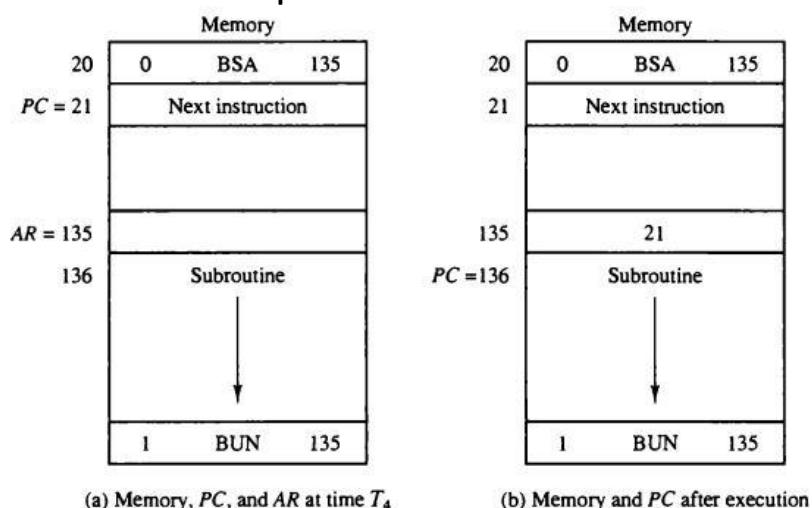
When executed, the BSA instruction stores the address of the next instruction in sequence (which is available in PC) into a memory location specified by the effective address.

The effective address plus one is then transferred to PC to serve as the address of the first instruction in the subroutine.

This operation was specified with the following register transfer: A numerical example that demonstrates how this instruction is used with a subroutine

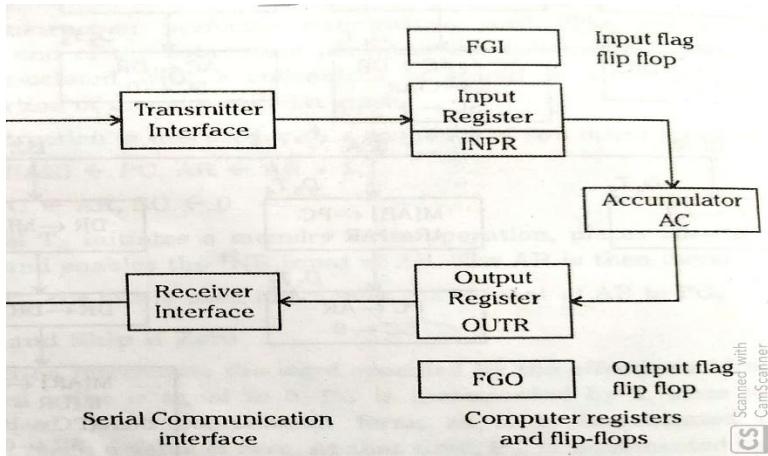
The BSA instruction is assumed to be in memory at address 20.

- The I bit is 0 and the address part of the instruction has the binary equivalent of 135.
- After the fetch and decode phases, PC contains 21, which is the address of the next instruction in the program (referred to as the return address). AR holds the effective address 135.
- This is shown in part (a) of the figure.
- The BSA instruction performs the following numerical operation:
- The result of this operation is shown in part (b) of the figure.
 - The return address 21 is stored in memory location 135 and control continues with the subroutine program starting from address 136.
 - The return to the original program (at address 21) is accomplished by means of an indirect BUN instruction placed at the end of the subroutine.



24. Explain with a neat block diagram the input-output configuration ?

Each quantity of information consists of 8 bits of an alphanumeric code. The serial information from the Keyboard is shifted into the input register INPR through the transmitter interface from the input register, the information is shifted in parallel into the Accumulator.



- The terminal sends and receives serial information.
 - Each quantity of information has eight bits of an alphanumeric code.
 - The serial information from the keyboard is shifted into the input register **INPR**.
 - The serial information for the printer is stored in the output register **OUTR**.
 - These two registers communicate with a communication interface serially and with the AC in parallel.
 - The input—output configuration is shown in Fig.

 - The input register **INPR** consists of eight bits and holds alphanumeric input information.
 - The 1-bit input flag **FGI** is a control flip-flop.
 - The flag bit is set to 1 when new information is available in the input device and is cleared to 0 when the information is accepted by the computer.
 - The output register **OUTR** works similarly but the direction of information flow is reversed.
 - Initially, the output flag **FGO** is set to 1.
 - The computer checks the flag bit; if it is 1, the information from **AC** is transferred in parallel to **OUTR** and **FGO** is cleared to 0.
 - The output device accepts the coded information, prints the corresponding character, and when the operation is completed, it sets **FGO** to 1.
- Input-Output Instructions:**
- Input and output instructions are needed for transferring information to and from **AC** register, for checking the flag bits, and for controlling the interrupt facility.
 - Input-output instructions have an operation code 1111 and are recognized by the control when **D7 = 1** and **I = 1**.
 - The remaining bits of the instruction specify the particular operation.

25. Explain the types of program interrupts.

Program interrupts: refers to the transfer of programs control from a currently running program to another service program as a result of an external or internal generated request.

Types of Interrupts:

There are three types of interrupts. They are:

- (1) External interrupts
- (2) Internal interrupts

(3) Software interrupts

- (1) External interrupts are initiated from
- ✓ Input-output devices(I/O devices)
 - ✓ Timing devices
 - ✓ Circuit monitoring power supply
 - ✓ Any external source

External interrupts are caused by

- ✓ I/O device requesting transfer of data
- ✓ I/O device finishing transfer of data
- ✓ Elapsed time of an event (timeout-program in endless loop) Power failure

- (2) Internals interrupts are also called Traps. They arise from illegal or erroneous use of an instruction or date.

Interrupts caused by internal error conditions are

- ✓ Register overflow
- ✓ Attempt to divide by zero
- ✓ Invalid operation code
- ✓ Stack overflow
- ✓ Protection violation

- (3) Software interrupt is initiated by executing an instruction . It is a special call instruction that behaves like an interrupt. It can be used to initiate an interrupt procedure at any desired point in the program.

26. What are the major characteristics of RISC architecture?

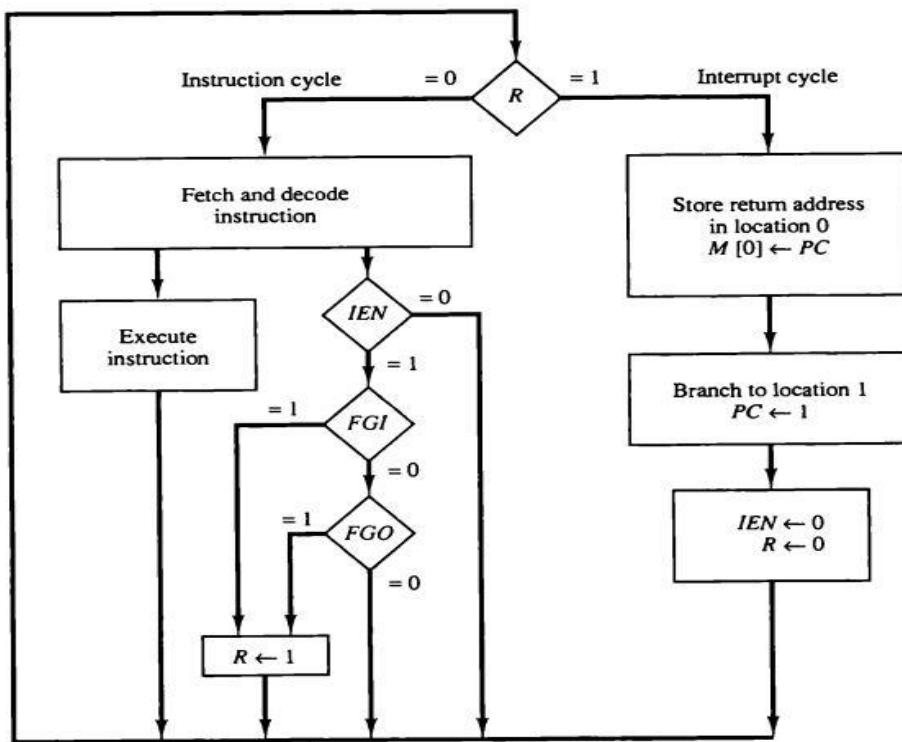
Characteristics of RISC :

1. It has relatively fewer instructions.
2. There are relatively fewer and simple addressing modes.
3. Memory access is limited only to load and store instructions.
4. There are a large number of registers in the CPU.
5. All operations are carried out using the registers of the CPU.
6. Instructions format is of fixed length.
7. Instruction is easily decodable.
8. Instruction execution takes place in a single clock cycle.
9. Control is hardwired and not microprogrammed.
10. There are few data types in hardware

27. Explain the operation of interrupt cycle with a flow chart.

The interrupt cycle is a hardware implementation of a branch and save return address operation.

1. The return address may be stored in a processor register, a memory stack or a specific memory location. Here, the memory location at address 0 is chosen to store the return address.
2. Then address 1 is inserted into Program Counter(PC), to branch to location 1.
3. The IEN and R are cleared so that no other interruptions can occur until the interrupt request from the flag has been serviced.



Flowchart for Interrupt Cycle

There is an interrupt flip flop R .

- When R is 0, the computer goes through an instruction cycle.
- When R is 1, the computer goes through an interrupt cycle.

28. Write a note on RISC and CISC.

Complex Instruction Set Computer(CISC):

A computer with a large number of instructions is called Complex Instruction Set Computer.

Features:

- The instructions provide direct manipulation of operands residing in memory.
- A large number of instruction about 100 to 250 are provided.
- A large variety of addressing modes about 5 to 20 are provided.

Examples of CISC architecture are Digital Equipment Corporation VAX computer and the IBM 370 computer.

Reduced Instruction Set Computer(RISC):

Computer that uses fewer instructions with simple constructs so that they can be executed much faster within the CPU without having to use memory often is called Reduced Instruction Set Computer.

Features:

- It has relatively fewer instructions.
- Memory access is limited only to load and store instructions.

- There are a large number of registers in the CPU

RISC families include Alpha, ARC, ARM, AVR, MIPS, SPARC, PIC, SuperH.

29. Explain input-output instructions.

Input/Output – These instructions are for communication between computer and outside environment. The IR(14 – 12) is 111 (differentiates it from memory reference) and IR(15) is 1 (differentiates it from register reference instructions). The rest 12 bits specify I/O operation.



Example –

IR register contains = 1111100000000000, i.e. INP after fetch and decode cycle we find out that it is an input/output instruction for inputting character. Hence, INPUT character from peripheral device.

The set of instructions incorporated in 16 bit IR register are:

Symbol	Hexadecimal Code	Description
INP	F800	Input character to AC
OUT	F400	Output character from AC
SKI	F200	Skip on input flag
SKO	F100	Skip on output flag
IEN	F080	Interrupt On

30. Explain input and output interface unit with neat diagram?

Input Output Interface provides a method for transferring information between internal storage and external I/O devices. Peripherals connected to a computer need special communication links for interfacing them with the central processing unit. The purpose of communication link is to resolve the differences that exist between the central computer and each peripheral.

The Major Differences are:-

- Peripherals are electro technical and electromagnetic devices and CPU and memory are electronic devices. Therefore, a conversion of signal values may be needed.
- The data transfer rate of peripherals is usually slower than the transfer rate of CPU and consequently, a synchronization mechanism may be needed.
- Data codes and formats in the peripherals differ from the word format in the CPU and memory.
- The operating modes of peripherals are different from each other and must be controlled so as not to disturb the operation of other peripherals connected to the CPU

- To Resolve these differences, computer systems include special hardware components between the CPU and Peripherals to supervises and synchronizes all input and out transfers

These components are called Interface Units because they interface between the processor bus and the peripheral devices.

I/O BUS and Interface Module: It defines the typical link between the processor and several peripherals. The I/O Bus consists of data lines, address lines and control lines. The I/O bus from the processor is attached to all peripherals interface. To communicate with a particular device, the processor places a device address on address lines. Each Interface decodes the address and control received from the I/O bus, interprets them for peripherals and provides signals for the peripheral controller. It is also synchronizing the data flow and supervises the transfer between peripheral and processor. Each peripheral has its own controller.

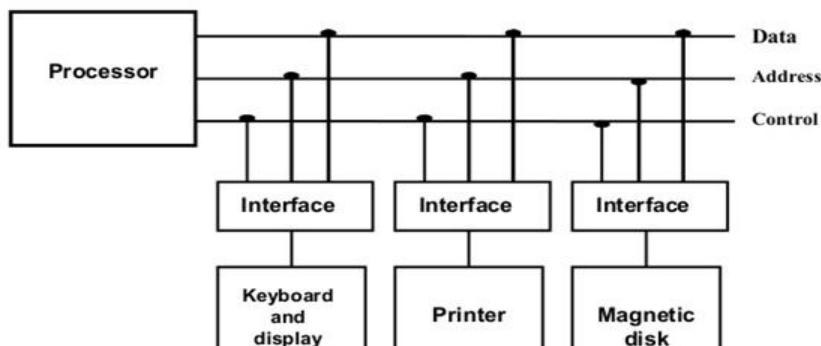
For example, the printer controller controls the paper motion, the print timing. The control lines are referred as I/O command. The commands are as following:
Control command- A control command is issued to activate the peripheral and to inform it what to do.

Status command- A status command is used to test various status conditions in the interface and the peripheral.

Data Output command- A data output command causes the interface to respond by transferring data from the bus into one of its registers.

Data Input command- The data input command is the opposite of the data output. In this case the interface receives on item of data from the peripheral and places it in its buffer register. I/O Versus Memory Bus

I/O BUS and Interface Module



Connection of I/O bus to input-output devices

31. Write a note on isolated vs memory mapped i/o?

Memory mapped I/O and Isolated I/O

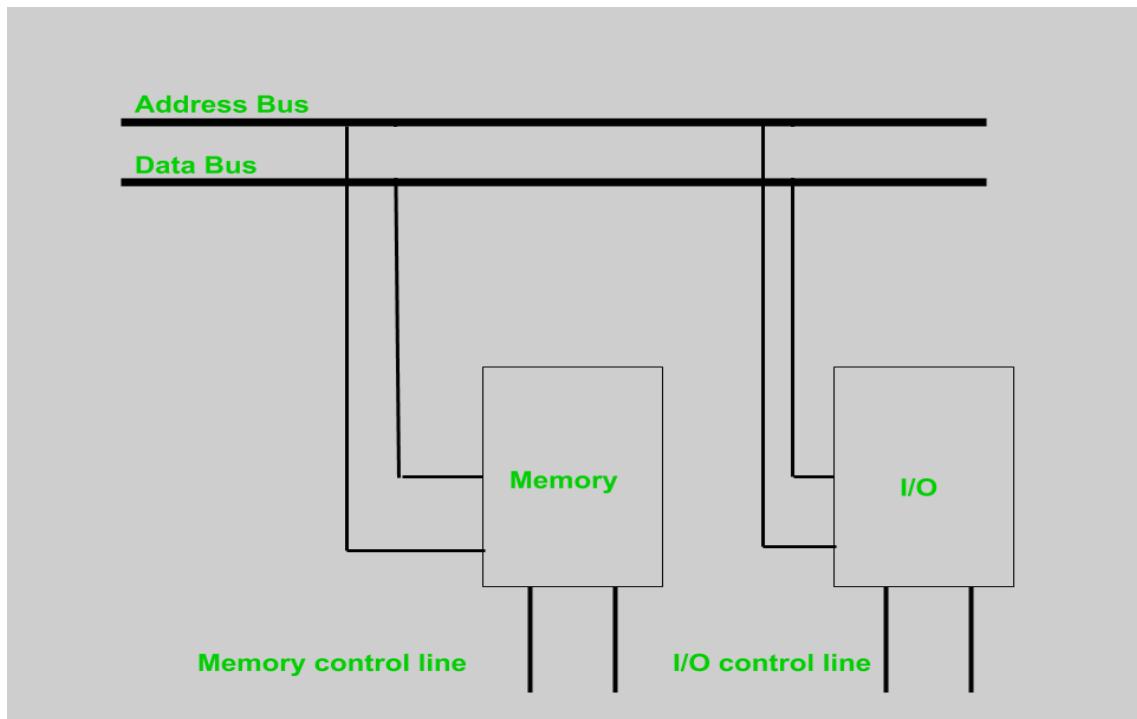
As a CPU needs to communicate with the various memory and input-output devices (I/O) as we know data between the processor and these devices flow with the help of the system bus. There are three ways in which system bus can be allotted to them :

1. Separate set of address, control and data bus to I/O and memory.
2. Have common bus (data and address) for I/O and memory but separate control lines.
3. Have common bus (data, address, and control) for I/O and memory.

In first case it is simple because both have different set of address space and instruction but require more buses.

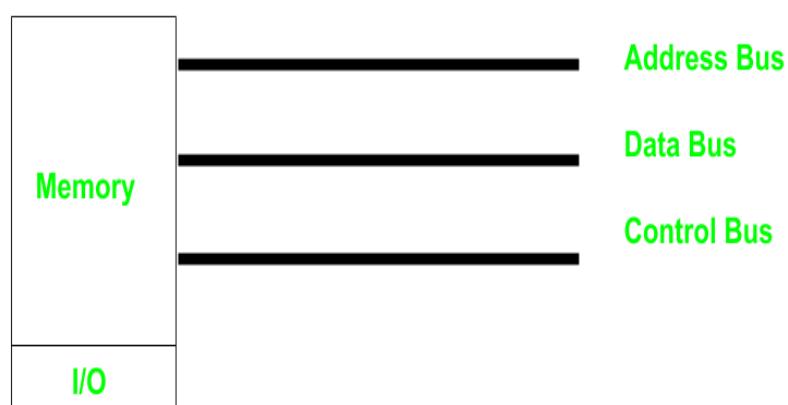
Isolated I/O

Then we have Isolated I/O in which we have common bus(data and address) for I/O and memory but separate read and write control lines for I/O. So when CPU decode instruction then if data is for I/O then it places the address on the address line and set I/O read or write control line on due to which data transfer occurs between CPU and I/O. As the address space of memory and I/O is isolated and the name is so. The address for I/O here is called ports. Here we have different read-write instruction for both I/O and memory.



Memory Mapped I/O

In this case every bus is common due to which the same set of instructions work for memory and I/O. Hence we manipulate I/O same as memory and both have same address space, due to which addressing capability of memory become less because some part is occupied by the I/O.



32. Differences between memory mapped I/O and isolated I/O –

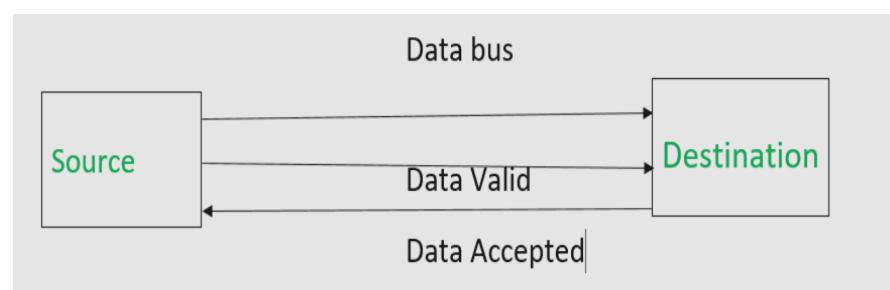
Isolated I/O	Memory Mapped I/O
Memory and I/O have separate address space	Both have same address space
All address can be used by the memory	Due to addition of I/O addressable memory become less for memory
Separate instruction control read and write operation in I/O and Memory	Same instructions can control both I/O and Memory
In this I/O address are called ports.	Normal memory address are for both
More efficient due to separate buses	Lesser efficient
Larger in size due to more buses	Smaller in size
It is complex due to separate logic is used to control both.	Simpler logic is used as I/O is also treated as memory only.

33. Explain source initiated data transfer using handshaking

Source initiated Handshaking – When source initiates the data transfer process. It consists of signals:

DATA VALID: if ON tells data on the data bus is valid otherwise invalid.

DATA ACCEPTED: if ON tells data is accepted otherwise not accepted.



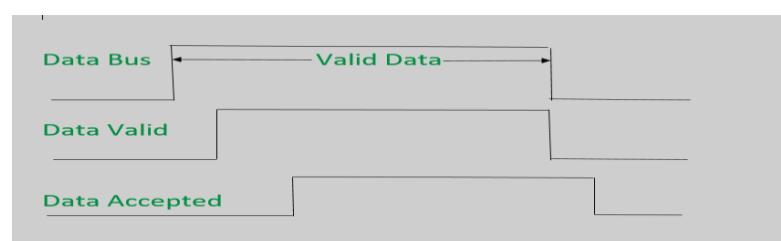
(i) Source places data on the data bus and enable Data valid signal.

(ii) Destination accepts data from the data bus and enable Data accepted signal.

(iii) After this, disable Data valid signal means data on data bus is invalid now.

(iv) Disable Data accepted signal and the process ends.

Now there is surely that destination has read the data from the data bus through data accepted signal. Signals can be seen as:



34. What is polling? Explain.

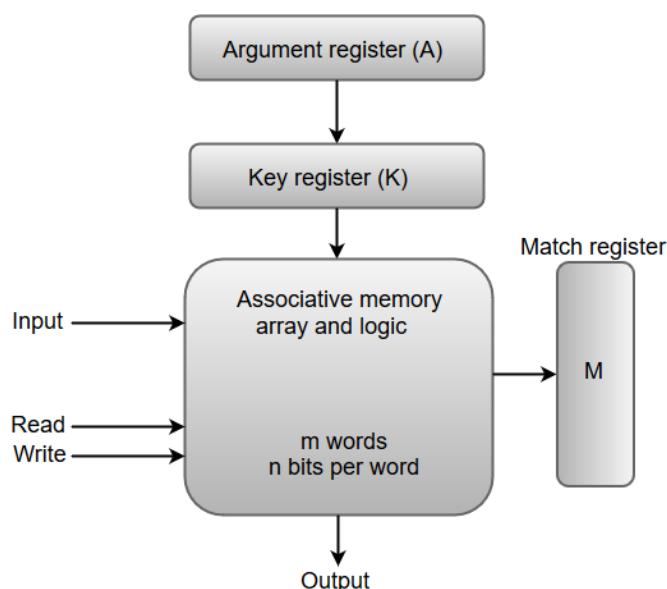
The software method used to identify the highest priority source is called Polling. There is one common branch address for all interrupts.

1. The program begins at the branch address and polls the interrupt sources in sequence.
2. The order in which they are tested decides the priority of the interrupts.
3. First the highest priority source is tested. If its interrupt signal is on, control branches to a service routine for this source. Otherwise, the next lower priority source is tested and so on.
4. The initial service routine for all interrupts consists of a program that tests the interrupt sources in sequence and branches to one of many possible service routines.
5. The particular service routine reached belongs to the highest priority device among all devices that interrupted the computer.

35. Explain Associative memory with a neat block diagram.

The associative memory has cells that have storage capability as well as logic circuits for matching its content with an external argument. These memories are used in applications where the search time is very critical and must be very short. A memory unit accessed by content is called an associative memory or content addressable memory(CAM)

Block diagram of associative memory



- Each word in memory is compared in parallel with the content of the argument register.
- The words that match the bits of the argument register set a corresponding bit in the match register.
- Those bits in the match register that have been set indicate that their corresponding words have been matched.
- A sequential access to memory for those words, whose corresponding bits in the match register have been set, causes reading of words in memory.

- The key register provides a mask for choosing a particular field or key in the argument word.
- Those bits in the argument that have 1's in their corresponding position of the key register are compared.

36. What are the important characteristics of memory?

Characteristics of memory:

1.Location	2.Capacity	3.Unit of transfer	4.Access Method	5.Performance
6.Physical type	7.Physical characteristics	8.Organization		

1. Location:

It deals with location of the memory device in the computer system. There are three possible locations:

- CPU : This is often in the form of CPU registers and small amount of cache.
- Internal or main: This is the main memory like RAM or ROM. The CPU can directly access the main memory
- External or secondary: It comprises of secondary storage devices like hard disks, magnetic tapes.

2. Capacity:

The capacity of any memory device is expressed in terms of: 1) Word size
2) Number of words

- Word size: words are expressed in bytes(8 bits). A word can however mean my number of bytes.
- Number of words: This specifies the number of words available in the particular memory device.

3. Unit of transfer:

It is the maximum number of bits that can be read or written into the memory.

4. Access Methods:

It is fundamental characteristics of memory devices. It is the sequence or order in which memory can be accessed.

5. Performance

The performance of system is determined using three parameters:

- **Access Time :** In random access memories, it is the time taken by memory to complete the read/write operation from the instant that an address is sent to the memory.
- **Memory cycle time:** It is defined only for random access memories and is the sum of the access time and the additional time required before the second access can commence.
- **Transfer rate:** It is defined as the rate at which data can be transferred into or out of a memory unit.

6. Physical rate:

Memory devices can be either semiconductor memory(like RAM) or magnetic surface memory(Like hard disks).

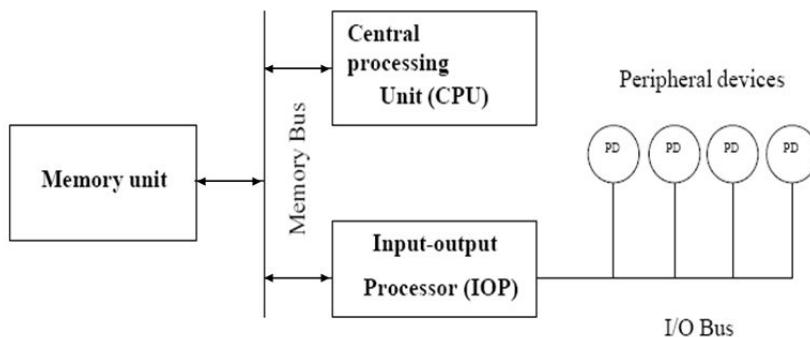
7. Physical Characteristics:

- ✓ **Volatile/Non Volatile:** If a memory devices continues hold data even if power is turned off. The memory device is non-volatile else it is volatile.

8. Organizations:

- ✓ **Erasable/Non-erasable:** The memories in which data once programmed cannot be erased are called Non-erasable memories. Memory device in which data in the memory can be erased is called erasable memory.

37. Explain the block diagram of a computer with I/O processors.



CPU is the master while the IOP is a slave processor. The CPU performs the task of initiating all operations.

The operations include

- ✓ Starting an I/O transfer
- ✓ Testing I/O status conditions needed for making decisions on various I/O activities.

I/O instructions are executed in the IOP. The IOP asks for the attention of the CPU by means of an interrupt. It also responds to CPU requests by placing a status word in a prescribed location in memory to be examined by CPU program.

For an I/O operation execution, the CPU informs the IOP where to find the I/O program and then leaves the transfer details to the IOP.

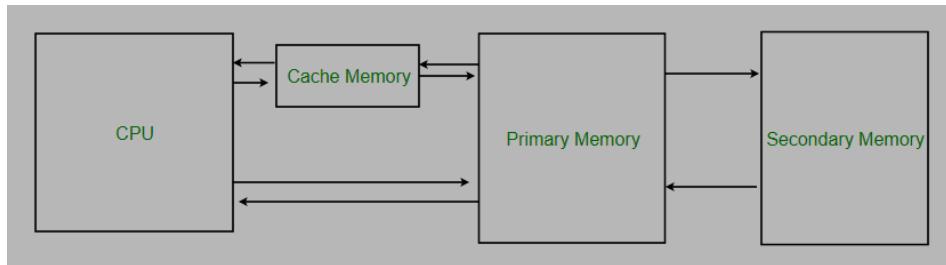
The instructions that are read from memory by an IOP are sometimes called Commands, to distinguish them from instructions that are read by the CPU.

38. Write a note on cache memory.

The active portions of the program and data are placed in a fast small memory. This reduces the average memory access time and hence the total execution time of the program. This memory is called Cache memory. It is placed between the CPU and the main memory. Performance of cache memory is frequently measured in terms of a quantity called Hit ratio. Loops and subroutines tend to localize the references to memory for fetching instruction. Reference to memory at any given interval of time tend to be confined within a few localized areas in memory. This phenomenon is known as the property of locality of reference.

The basic operations of cache can be as follows:

1. When CPU wants to access memory, it first examines the cache.
 - a. If the word is found in the cache, it is read from this memory.
 - b. If the word is not found in the cache, the main memory is accessed.
2. A block of words containing the most recent accessed one is then transferred from main memory to cache memory so that future references can find the required words in cache.



39. Write a note on modes of data transfer

Modes of I/O Data Transfer

Data transfer between the central unit and I/O devices can be handled in generally three types of modes which are given below:

1. Programmed I/O
2. Interrupt Initiated I/O
3. Direct Memory Access

Programmed I/O :

- Programmed I/O instructions are the result of I/O instructions written in computer program. Each data item transfer is initiated by the instruction in the program. Usually the program controls data transfer to and from CPU and peripheral. Transferring data under programmed I/O requires constant monitoring of the peripherals by the CPU.

Interrupt Initiated I/O

- In the programmed I/O method the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer. This is time consuming process because it keeps the processor busy needlessly.

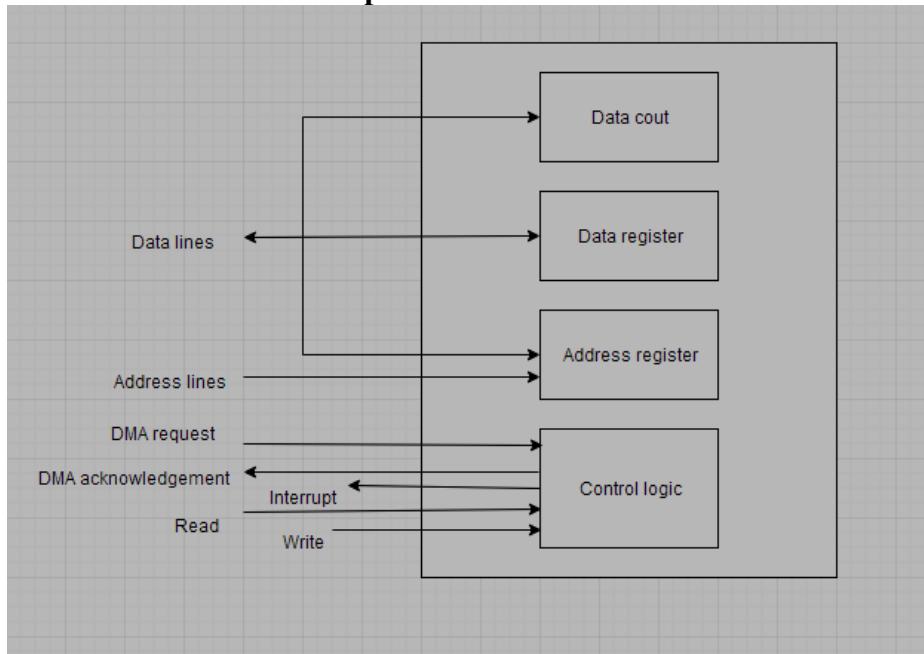
This problem can be overcome by using interrupt initiated I/O. In this when the interface determines that the peripheral is ready for data transfer, it generates an interrupt. After receiving the interrupt signal, the CPU stops the task which it is processing and service the I/O transfer and then returns back to its previous processing task.

Direct Memory Access

- Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This technique is known as DMA. In this, the interface transfer data to and from the memory through memory bus. A DMA controller manages to transfer data between peripherals and memory unit.

Many hardware systems use DMA such as disk drive controllers, graphic cards, network cards and sound cards etc. It is also used for intra chip data transfer in multicore processors. In DMA, CPU would initiate the transfer, do other operations

while the transfer is in progress and receive an interrupt from the DMA controller when the transfer has been completed.



Above figure shows block diagram of DMA

40. Write a note on DMA

Direct Memory Access (DMA):

In the Direct Memory Access (DMA) the interface transfer the data into and out of the memory unit through the memory bus. The transfer of data between a fast storage device such as magnetic disk and memory is often limited by the speed of the CPU. Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This transfer technique is called Direct Memory Access(DMA).

During the DMA transfer, the CPU is idle and has no control of the memory buses. A DMA Controller takes over the buses to manage the transfer directly between the I/O device and memory.

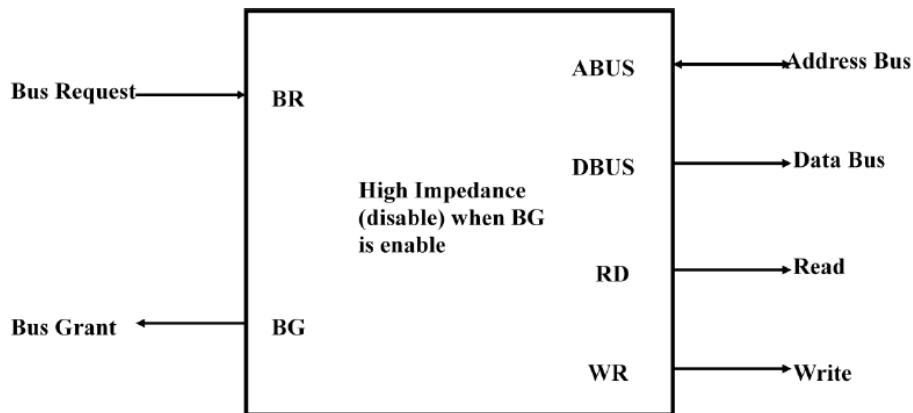
The CPU may be placed in an idle state in a variety of ways. One common method extensively used in microprocessor is to disable the buses through special control signals

such as:

- Bus Request (BR)
- Bus Grant (BG)

These two control signals in the CPU that facilitates the DMA transfer. The *Bus Request*

(BR) input is used by the *DMA controller* to request the CPU. When this input is active, the CPU terminates the execution of the current instruction and places the address bus, data bus and read write lines into a *high Impedance state*. High Impedance state means that the output is disconnected



CPU bus Signals for DMA Transfer

The CPU activates the **Bus Grant (BG)** output to inform the external DMA that the Bus Request (BR) can now take control of the buses to conduct memory transfer without processor. When the DMA terminates the transfer, it disables the **Bus Request (BR)** line. The CPU disables the **Bus Grant (BG)**, takes control of the buses and return to its normal operation.

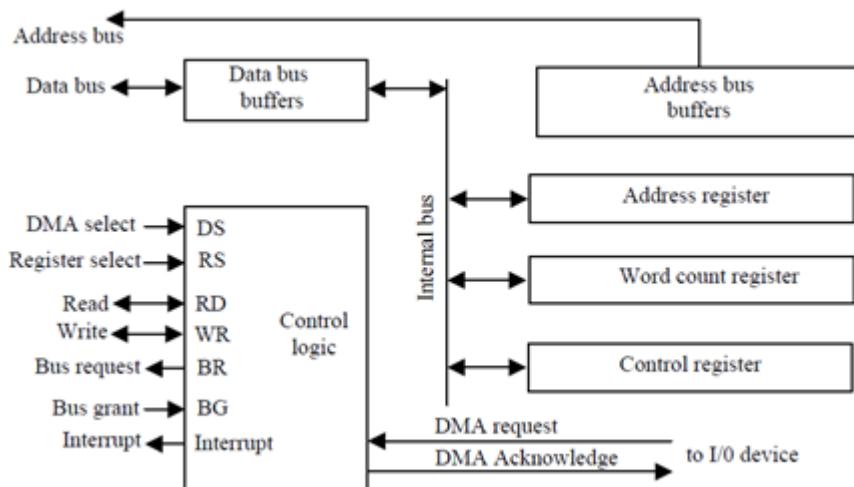
The transfer can be made in several ways that are:

- DMA Burst
- Cycle Stealing

- DMA Burst :- In DMA Burst transfer, a block sequence consisting of a number of memory words is transferred in continuous burst while the DMA controller is master of the memory buses.
- Cycle Stealing :- Cycle stealing allows the DMA controller to transfer one data word at a time, after which it must return control of the buses to the CPU.

41. Explain DMA controller with a block diagram.

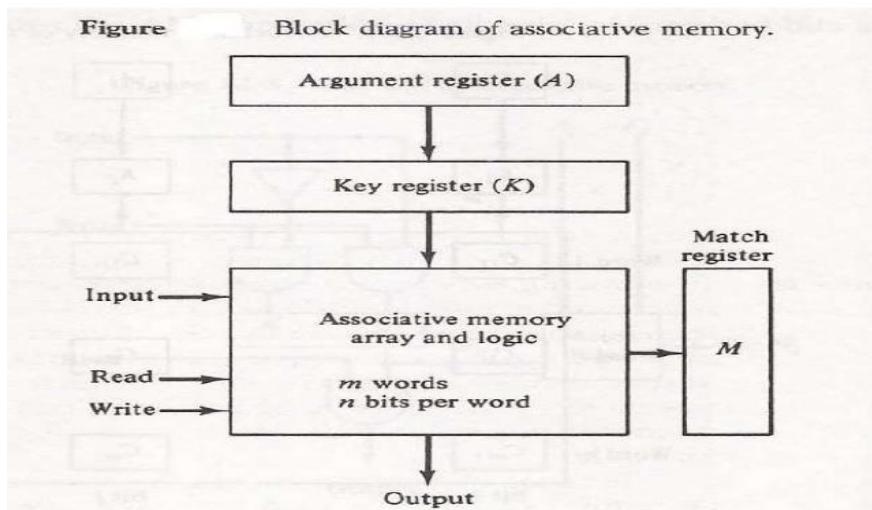
- DMA controller provides an interface between the bus and the input-output devices. Although it transfers data without intervention of processor, it is controlled by the processor. The processor initiates the DMA controller by sending the starting address, Number of words in the data block and direction of transfer of data i.e. from I/O devices to the memory or from main memory to I/O devices. More than one external device can be connected to the DMA controller.
- DMA controller contains an address unit, for generating addresses and selecting I/O device for transfer. It also contains the control unit and data count for keeping counts of the number of blocks transferred and indicating the direction of transfer of data. When the transfer is completed, DMA informs the processor by raising an interrupt. The typical block diagram of the DMA controller is shown in the figure below.



- DMA controller has to share the bus with the processor to make the data transfer. The device that holds the bus at a given time is called bus master. When a transfer from I/O device to the memory or vice versa has to be made, the processor stops the execution of the current program, increments the program counter, moves data over stack then sends a DMA select signal to DMA controller over the address bus. If the DMA controller is free, it requests the control of bus from the processor by raising the bus request signal. Processor grants the bus to the controller by raising the bus grant signal, now DMA controller is the bus master. The processor initiates the DMA controller by sending the memory addresses, number of blocks of data to be transferred and direction of data transfer. After assigning the data transfer task to the DMA controller, instead of waiting ideally till completion of data transfer, the processor resumes the execution of the program after retrieving instructions from the stack.

42. Explain the working of Associative memory,

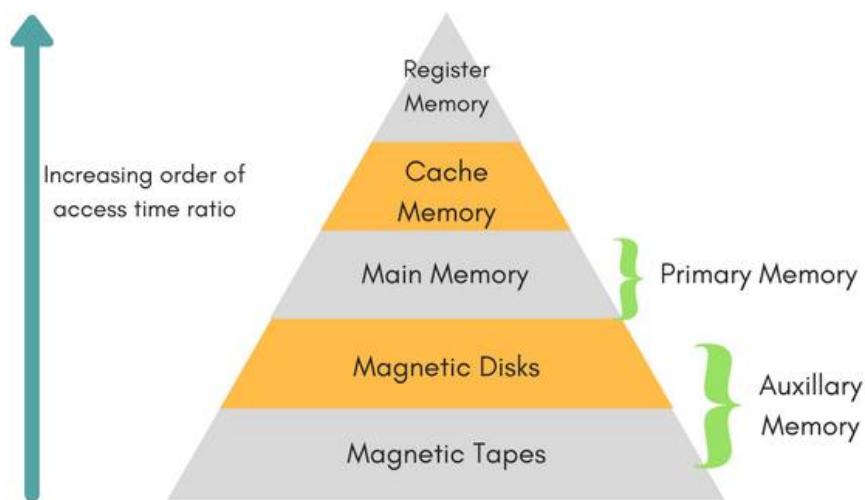
The associative memory has cells that have storage capability as well as logic circuits for matching its content with an external argument. These memories are used in applications where the search time is very critical and must be very short. A memory unit accessed by content is called an associative memory or content addressable memory(CAM)



- Each word in memory is compared in parallel with the content of the argument register.
- The words that match the bits of the argument register set a corresponding bit in the match register.
- Those bits in the match register that have been set indicate that their corresponding words have been matched.
- A sequential access to memory for those words, whose corresponding bits in the match register have been set, causes reading of words in memory.
- The key register provides a mask for choosing a particular field or key in the argument word.
- Those bits in the argument that have 1's in their corresponding position of the key register are compared.
-

43. Explain memory hierarchy.

Memory Hierarchy is the combination of storage components that make up the overall physical memory system of a computer. The various components are typically arranged from fastest to the slowest in terms of their access time. So the term 'hierarchy' is used. The goal of using hierarchy is to obtain the highest possible access speed while minimizing total cost of entire memory system. The memory hierarchy system consists of all storage devices used in a computer system from smaller and faster cache memory to the slower high capacity auxiliary memory.



Memory Hierarchy in a computer

The total memory capacity of a computer can be visualized by hierarchy of components. The memory hierarchy system consists of all storage devices contained in a computer system from the slow Auxillary Memory to fast Main Memory and to smaller Cache memory.

Auxillary memory access time is generally 1000 times that of the main memory, hence it is at the bottom of the hierarchy.

The main memory occupies the central position because it is equipped to communicate directly with the CPU and with auxiliary memory devices through Input/output processor (I/O).

When the program not residing in main memory is needed by the CPU, they are brought in from auxiliary memory. Programs not currently needed in main memory are transferred into auxiliary memory to provide space in main memory for other programs that are currently in use.

The cache memory is used to store program data which is currently being executed in the CPU. Approximate access time ratio between cache memory and main memory is about 1 to 7~10.

44. Write a note on virtual memory?

virtual memory is a memory management capability of an operating system (OS) that uses hardware and software to allow a computer to compensate for physical memory shortages by temporarily transferring data from random access memory (RAM) to disk storage. Virtual address space is increased using active memory in RAM and inactive memory in hard disk drives (HDDs) to form contiguous addresses that hold both the application and its data.

Virtual memory was developed at a time when physical memory -- the installed RAM -- was expensive. Computers have a finite amount of RAM, so memory can run out, especially when multiple programs run at the same time. A system using virtual memory uses a section of the hard drive to emulate RAM. With virtual memory, a system can load larger programs or multiple programs running at the same time, allowing each one to operate as if it has infinite memory and without having to purchase more RAM.

While copying virtual memory into physical memory, the OS divides memory into pagefiles or swap files with a fixed number of addresses. Each page is stored on a disk and when the page is needed, the OS copies it from the disk to main memory and translates the virtual addresses into real addresses.
