

Digital Electronics (4321102) - Summer 2024 Solution

Milav Dabgar

June 20, 2024

Question 1(a) [3 marks]

Convert: $(110101)_2 = (\quad)_{10} = (\quad)_8 = (\quad)_{16}$

Solution

Step-by-step conversion of $(110101)_2$:

Table 1. Binary Conversion Table

Binary $(110101)_2$	Decimal	Octal	Hexadecimal
$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$	$32+16+0+4+0+1 = 53$	$6 \times 8^1 + 5 \times 8^0 = 48 + 5 = 53$	$3 \times 16^1 + 5 \times 16^0 = 48 + 5 = 53$
$(110101)_2$	$(53)_{10}$	$(65)_8$	$(35)_{16}$

Mnemonic

"Binary Digits Out Here" (BDOH) for Binary→Decimal→Octal→Hexadecimal conversion.

Question 1(b) [4 marks]

Perform: (i) $(11101101)_2 + (10101000)_2$ (ii) $(11011)_2 * (1010)_2$

Solution

Table for binary addition and multiplication:

Table 2. Binary Arithmetic

(i) Binary Addition	(ii) Binary Multiplication
<div>1 11101101 2 + 10101000 3 ----- 4 110010101</div>	<div>1 11011 2 × 1010 3 ----- 4 00000 5 11011 6 00000 7 11011 8 ----- 9 11101110</div>

Decimal verification:

- (i) $(11101101)_2 = 237$, $(10101000)_2 = 168$, Sum = $405 = (110010101)_2$
- (ii) $(11011)_2 = 27$, $(1010)_2 = 10$, Product = $270 = (11101110)_2$

Mnemonic

"Carry Up Makes Sum" for addition and "Shift Left Add Product" for multiplication.

Question 1(c) [7 marks]

- (i) Convert: $(48)_{10} = (\quad)_2 = (\quad)_8 = (\quad)_{16}$
 (ii) Subtract using 2's Complement method: $(1110)_2 - (1000)_2$
 (iii) Divide $(1111101)_2$ with $(101)_2$

Solution**(i) Conversion Table:****Table 3.** Decimal (48) Conversion

Decimal $(48)_{10}$	Binary	Octal	Hexadecimal
$48 \div 2 = 24$ rem 0	110000	60	30
$24 \div 2 = 12$ rem 0			
$12 \div 2 = 6$ rem 0			
$6 \div 2 = 3$ rem 0			
$3 \div 2 = 1$ rem 1			
$1 \div 2 = 0$ rem 1			
$(48)_{10}$	$(110000)_2$	$(60)_8$	$(30)_{16}$

(ii) Subtraction Table:**Table 4.** 2's Complement Subtraction

2's Complement Method	Steps
$(1110)_2 - (1000)_2$	1. Find 2's complement of $(1000)_2$
1's complement of $(1000)_2$	$(0111)_2$
2's complement	$(0111)_2 + 1 = (1000)_2$
$(1110)_2 + (1000)_2$	$(10110)_2$
Discard carry	$(0110)_2$
Result	$(0110)_2 = 6_{10}$

(iii) Division:

```

1      11001
2      -----
3 101)1111101
4      101
5      ---
6      0101
7      101
8      ---
9      0000
10     000

```

```

11  ----
12      001
13      000
14      ---
15      1

```

Quotient = $(11001)_2$, Remainder = $(1)_2$

Mnemonic

"Division Drops Down Remainders" for long division process.

Question 1(c OR) [7 marks]

Explain Codes: ASCII, BCD, Gray

Solution

Table of Common Digital Codes:

Table 5. Digital Codes

Code	Description	Example
ASCII (American Standard Code for Information Interchange)	7-bit code representing 128 characters including alphabets, numbers, and special symbols	A = 65 $(1000001)_2$
BCD (Binary Coded Decimal)	Represents each decimal digit (0-9) using 4 bits	42 = 0100 0010
Gray Code	Binary code where adjacent numbers differ by only one bit	$(0,1,3,2) = (00,01,11,10)$

Diagram: Gray Code Generation:



Binary: 0011 $\xrightarrow{\text{XOR}}$ Gray: 0010

Figure 1. Gray Code Concept

Question 2(a) [3 marks]

Simplify using Boolean Algebra: $Y = A B + A' B + A' B' + A B'$

Solution

Step-by-step simplification:

Table 6. Boolean Simplification

Step	Expression	Boolean Law
$Y = AB + A'B + A'B' + AB'$	Initial expression	-
$Y = A(B + B') + A'(B + B')$	Factoring	Distributive law
$Y = A(1) + A'(1)$	Complement law	$B + B' = 1$
$Y = A + A'$	Simplification	-
$Y = 1$	Complement law	$A + A' = 1$

Mnemonic

"Factor, Simplify, Finish" for Boolean simplification steps.

Question 2(b) [4 marks]

Simplify the following Boolean function using K-map: $f(A,B,C,D) = \Sigma m(0,3,4,6,8,11,12)$

Solution

K-map Solution:

		AB				
		00	01	11	10	
CD	00	00	1	0	0	1
	01	01	0	0	0	1
	11	11	0	1	0	0
	10	10	0	0	1	0

Grouping:

- Group 1: $m(0,8) = A'C'D'$
- Group 2: $m(4,12) = BD'$
- Group 3: $m(3,11) = CD$
- Group 4: $m(6) = A'B'CD'$

Simplified expression: $f(A, B, C, D) = A'C'D' + BD' + CD + A'B'CD'$

Mnemonic

"Group Powers Of Two" for K-map grouping strategy.

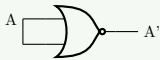
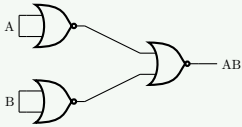

Question 2(c) [7 marks]

Explain NOR gate as a universal gate with neat diagrams.

Solution

NOR as Universal Gate:

Table 7. NOR Implementations

Function	Implementation using NOR	Truth Table		
NOT Gate		A	A'	
		0	1	
		1	0	
AND Gate		A	B	AB
		0	0	0
		0	1	0
		1	0	0
		1	1	1
OR Gate		A	B	A+B
		0	0	0
		0	1	1
		1	0	1
		1	1	1

Mnemonic

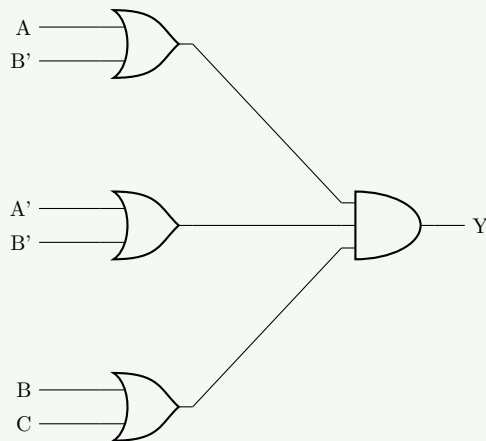
"NOT AND OR, NOR does more" for remembering NOR gate implementations.

Question 2(a OR) [3 marks]

Draw logic circuit for Boolean expression: $Y = (A + B') \cdot (A' + B') \cdot (B + C)$

Solution

Logic Circuit Implementation:



Truth Table Verification:

- Term 1: $(A + B')$
- Term 2: $(A' + B')$
- Term 3: $(B + C)$
- Output: $Y = \text{Term1} \cdot \text{Term2} \cdot \text{Term3}$

Mnemonic

"Each Term Separately" for breaking complex expressions.

Question 2(b OR) [4 marks]

State De-Morgan's theorems and prove it.

Solution

De-Morgan's Theorems and Proof:

Table 8. De-Morgan's Theorems

Theorem	Statement	Proof by Truth Table						
Theorem 1	$(A \cdot B)' = A' + B'$	A	B	AB	$(AB)'$	A'	B'	$A' + B'$
		0	0	0	1	1	1	1
		0	1	0	1	1	0	1
		1	0	0	1	0	1	1
		1	1	1	0	0	0	0
Theorem 2	$(A + B)' = A' \cdot B'$	A	B	A+B	$(A+B)'$	A'	B'	$A' \cdot B'$
		0	0	0	1	1	1	1
		0	1	1	0	1	0	0
		1	0	1	0	0	1	0
		1	1	1	0	0	0	0

Mnemonic

"Break BAR, Change Operation, Invert Inputs" for applying De-Morgan's law.

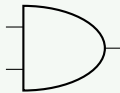
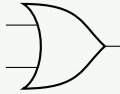
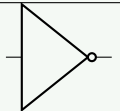
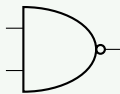
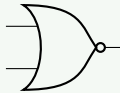
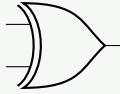
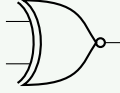
Question 2(c OR) [7 marks]

Explain all the Logic Gates with the help of Symbol, Truth table and equation.

Solution

Logic Gates Summary:

Table 9. Logic Gates Overview

Gate	Symbol	Truth Table		Equation	Description	
AND		0	0	0	$Y = A \cdot B$	Output 1 only when all inputs are 1
		0	1	0		
		1	0	0		
		1	1	1		
OR		0	0	0	$Y = A + B$	Output 1 when any input is 1
		0	1	1		
		1	0	1		
		1	1	1		
NOT		0	1	$Y = A'$	Inverts the input	
1	0					
NAND		0	0	1	$Y = (A \cdot B)'$	AND followed by NOT
		0	1	1		
		1	0	1		
		1	1	0		
NOR		0	0	1	$Y = (A + B)'$	OR followed by NOT
		0	1	0		
		1	0	0		
		1	1	0		
XOR		0	0	0	$Y = A \oplus B$	Output 1 when inputs are different
		0	1	1		
		1	0	1		
		1	1	0		
XNOR		0	0	1	$Y = (A \oplus B)'$	Output 1 when inputs are same
		0	1	0		
		1	0	0		
		1	1	1		

Mnemonic

"All Operations Need Necessary eXecution" (first letter of each gate - AND, OR, NOT, NAND, NOR, XOR).

Question 3(a) [3 marks]

Briefly explain 4:2 Encoder.

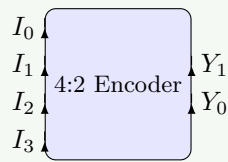
Solution

4-to-2 Encoder Overview:

Truth Table:

I_0	I_1	I_2	I_3	Y_1	Y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Diagram:

**Mnemonic**

"Input Position Creates Output" for encoder function.

Question 3(b) [4 marks]

Explain 4-bit Parallel adder using full adder blocks.

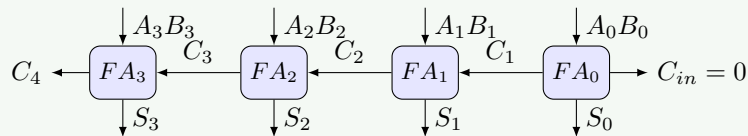
Solution

4-bit Parallel Adder:

Table 10. Component Function

Component	Function
Full Adder	Adds 3 bits (A, B, Carry-in) producing Sum and Carry-out
Parallel Adder	Connects 4 full adders with carry propagation

Diagram: 4-bit Parallel Adder:

**Mnemonic**

"Carry Always Passes Right" for the carry propagation in parallel adder.

Question 3(c) [7 marks]

Describe 8:1 Multiplexer with truth table, equation and circuit diagram.

Solution

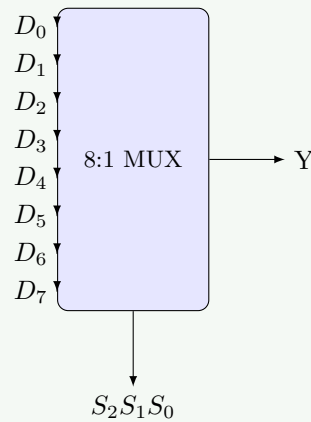
8:1 Multiplexer:

Truth Table:

S_2	S_1	S_0	Output Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Boolean Equation:

$$Y = S'_2S'_1S'_0D_0 + S'_2S'_1S_0D_1 + S'_2S_1S'_0D_2 + S'_2S_1S_0D_3 + S_2S'_1S'_0D_4 + S_2S'_1S_0D_5 + S_2S_1S'_0D_6 + S_2S_1S_0D_7$$

Diagram:**Mnemonic**

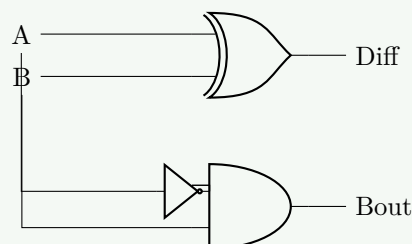
"Select Decides Data Output" for multiplexer operation.

Question 3(a OR) [3 marks]

Draw the logic circuit of half Subtractor and explain its working.

Solution**Half Subtractor:****Truth Table:**

A	B	Diff	Bout
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Logic Circuit:**Equations:**

- Difference (D) = $A \oplus B$
- Borrow out (Bout) = $A' \cdot B$

Mnemonic

"Different Bits Borrow" for half subtractor operation.

Question 3(b OR) [4 marks]

Explain 3:8 Decoder with truth table and circuit diagram.

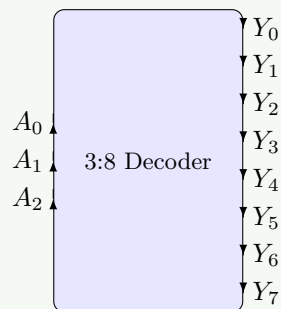
Solution

3:8 Decoder:

Truth Table (Partial):

A_2	A_1	A_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
...
1	1	1	0	0	0	0	0	0	0	0

Circuit Diagram:



Equations:

- $Y_0 = A'_2 \cdot A'_1 \cdot A'_0$
- $Y_1 = A'_2 \cdot A'_1 \cdot A_0$
- ...
- $Y_7 = A_2 \cdot A_1 \cdot A_0$

Mnemonic

"Binary Input Activates Output" for decoder operation.

Question 3(c OR) [7 marks]

Explain Gray to Binary code converter with truth table, equation and circuit diagram.

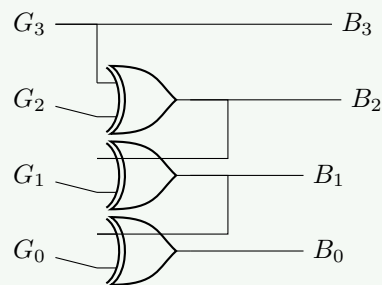
Solution

Gray to Binary Converter:

Table: Gray to Binary:

Gray	Binary
0000	0000
0001	0001
0011	0010
0010	0011
0110	0100
...	...

Circuit Diagram:



Equations:

- $B_3 = G_3$
- $B_2 = B_3 \oplus G_2$
- $B_1 = B_2 \oplus G_1$
- $B_0 = B_1 \oplus G_0$

Mnemonic

"MSB Stays, Rest XOR" for Gray to Binary conversion.

Question 4(a) [3 marks]

Explain D flip flop with truth table and circuit diagram.

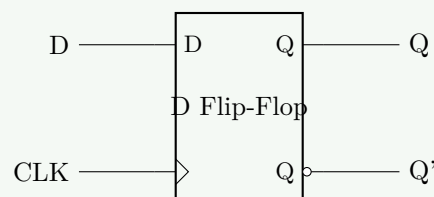
Solution

D Flip-Flop:

Truth Table:

CLK	D	Q	Q'
↑	0	0	1
↑	1	1	0

Circuit Diagram:



Characteristic Equation: $Q_{next} = D$

Mnemonic

"Data Delays one clock" for D flip-flop operation.

Question 4(b) [4 marks]

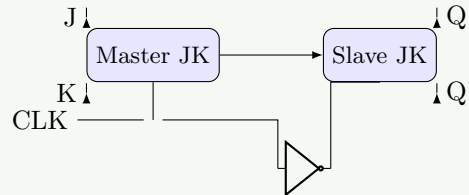
Explain working of Master Slave JK flip flop.

Solution**Master-Slave JK Flip-Flop:****Table 11.** Operation Principle

Component	Operation
Master	Samples inputs when CLK = 1
Slave	Transfers master output when CLK = 0

Truth Table:

J	K	Q(next)
0	0	No change
0	1	0
1	0	1
1	1	Toggle

Diagram:**Mnemonic**

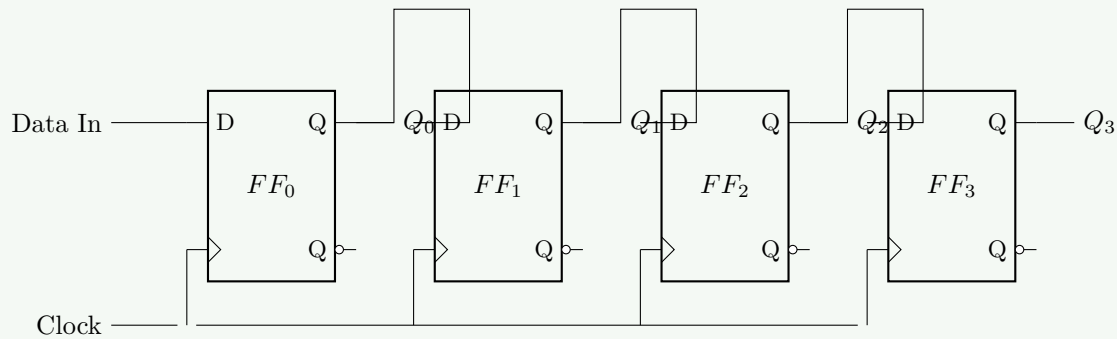
"Master Samples, Slave Transfers" for master-slave operation.

Question 4(c) [7 marks]

Classify Shift Registers with the help of Block diagram and Explain any one of them in detail.

Solution**Shift Register Classification:****Table 12.** Shift Register Types

Type	Description	Function
SISO	Serial In Serial Out	Data enters/exits serially
SIPO	Serial In Parallel Out	Data enters serially, exits parallel
PISO	Parallel In Serial Out	Data enters parallel, exits serially
PIPO	Parallel In Parallel Out	Data enters/exits parallel

SIPO Shift Register in Detail:**Timing Diagram:**

	Clk 1	Clk 2	Clk 3	Clk 4
Din:	1	0	0	0
Q0:	0			
Q1:		0		
Q2:			0	
Q3:				0

Mnemonic

"Serial Inputs Parallel Outputs" for SIPO operation.

Question 4(a OR) [3 marks]

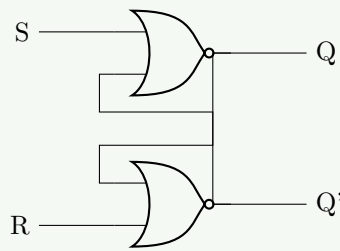
Explain SR flip flop with truth table and circuit diagram.

Solution

SR Flip-Flop:
Truth Table:

S	R	Q	Q'
0	0	No Change	No Change
0	1	0	1
1	0	1	0
1	1	Invalid	Invalid

Circuit Diagram:

**Mnemonic**

"Set to 1, Reset to 0" for SR flip-flop operation.

Question 4(b OR) [4 marks]

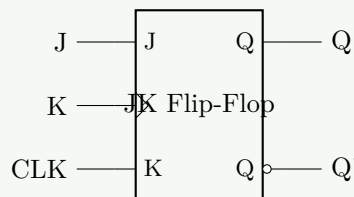
Describe JK flip flop with truth table and circuit diagram.

Solution

JK Flip-Flop:
Truth Table:

J	K	Q(next)
0	0	No Change
0	1	0
1	0	1
1	1	Toggle

Circuit Diagram:



Equation: $Q_{next} = JQ' + K'Q$

Mnemonic

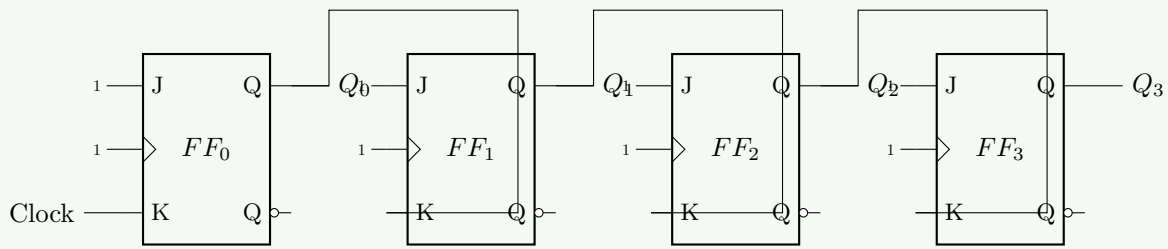
"Jump-Keep-Toggle" for JK flip-flop states.

Question 4(c OR) [7 marks]

Describe 4-bit Asynchronous UP Counter with truth table and circuit diagram.

Solution

4-bit Asynchronous UP Counter:
Count Sequence: 0000 → 1111
Circuit Diagram:

**Working:**

- Clock drives only first FF (FF_0).
- Output of each FF drives clock of next FF.
- FFs toggle on negative edge.

Mnemonic

"Ripple Carries Propagation Delay" for asynchronous counter operation.

Question 5(a) [3 marks]

Compare following logic families: TTL, CMOS, ECL

Solution**Logic Families Comparison:**

Table 13. Comparison of Logic Families

Parameter	TTL	CMOS	ECL
Technology	Bipolar	MOSFETs	Bipolar
Power	Medium	Very low	High
Speed	Medium	Low-Medium	Very high
Noise Immunity	Medium	High	Low
Fan-out	10	50+	25
Voltage	5V	3-15V	-5.2V

Mnemonic

"Technology Controls Many Electrical Characteristics" for comparing logic families.

Question 5(b) [4 marks]

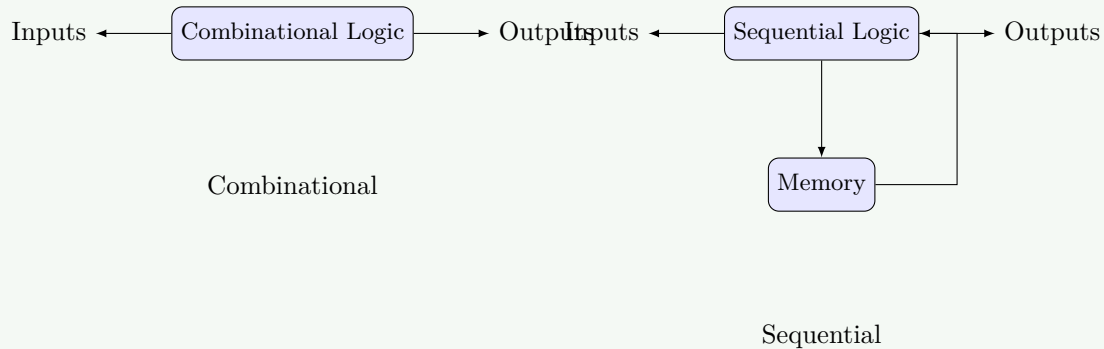
Compare Combinational and Sequential Logic Circuits.

Solution**Combinational vs Sequential Circuits:**

Table 14. Combinational vs Sequential

Parameter	Combinational	Sequential
Output depends on	Current inputs only	Inputs and previous state
Memory	No memory	Has memory elements
Feedback	No feedback paths	Contains feedback
Examples	Adders, MUX	Flip-flops, Counters
Clock	Not required	Often required

Diagram:



Mnemonic

"Current Only vs Memory States" for differentiating combinational and sequential circuits.

Question 5(c) [7 marks]

Define: Fan in, Fan out, Noise margin, Propagation delay, Power dissipation, Figure of merit, RAM

Solution

Digital Electronics Key Definitions:

Table 15. Definitions

Term	Definition	Typical Values
Fan-in	Max inputs a gate can handle	TTL: 2-8
Fan-out	Max gate inputs driven by one output	TTL: 10
Noise margin	Max noise voltage without error	TTL: 0.4V
Prop. delay	Time for input change to cause output change	TTL: 10ns
Power dissipation	Power consumed during operation	TTL: 10mW
Figure of merit	Speed \times Power (lower is better)	TTL: 100pJ
RAM	Random Access Memory (Volatile)	SRAM, DRAM

Mnemonic

"Fast Power Needs Proper Figure Ratings" for remembering the parameter terms.

Question 5(a OR) [3 marks]

Describe steps and the need of E-waste management of Digital ICs.

Solution**E-waste Management:**

1. **Collection:** Separate collection prevents improper disposal.
2. **Segregation:** Separating ICs from other parts.
3. **Dismantling:** Removing hazardous parts.
4. **Recovery:** Extracting gold/silicon.
5. **Safe disposal:** Managing non-recyclables.

Need:

- Hazardous Materials (Lead, Mercury).
- Resource Conservation.
- Environmental Protection.

Mnemonic

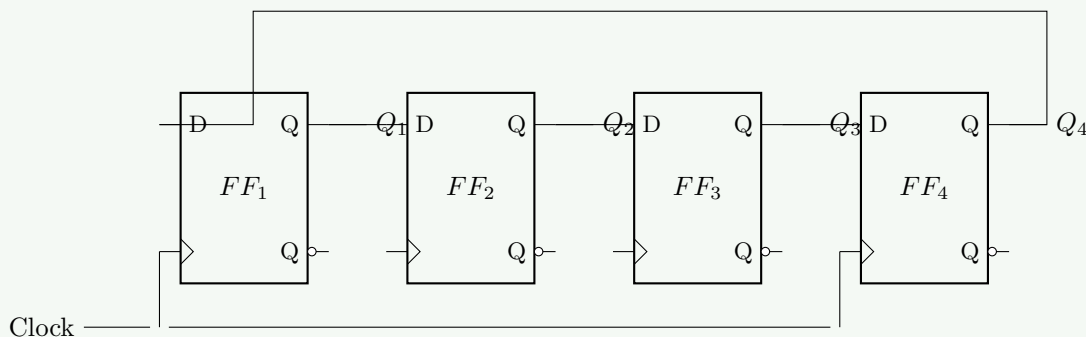
"Collection Starts Dismantling Recovery Safely" for e-waste management steps.

Question 5(b OR) [4 marks]

Explain working of Ring Counter with circuit diagram.

Solution**Ring Counter:**

Sequence: 1000 → 0100 → 0010 → 0001 → 1000

Circuit Diagram:**Mnemonic**

"One Bit Rotates Only" for ring counter operation.

Question 5(c OR) [7 marks]

Classify: (i) Memories (ii) Different Logic Families

Solution**(i) Memory Classification:**

Table 16. Memory Types

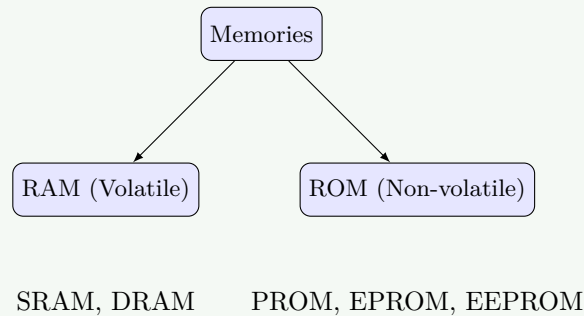
Type	Subtypes	Characteristics
RAM	SRAM	Static, Fast, Flip-flop based
	DRAM	Dynamic, Slower, Capacitor based
ROM	PROM	One-time programmable
	EPROM	Erasable (UV), Reprogrammable
	EEPROM	Electrically Erasable, Byte-level
	Flash	Block-level erasure, Non-volatile

(ii) Logic Families:

Table 17. Logic Families

Technology	Family (Characteristics)
Bipolar	TTL (Medium speed), ECL (High speed), I ² L (High density)
MOS	NMOS, PMOS, CMOS (Low power, High noise immunity)
Hybrid	BiCMOS (High speed + Low power)

Diagram:



Mnemonic

"Remember Simple Division: Programmable Erasable Electrical" for memory types.