9	Determine the equivalence partition and reduced table for the given state machine shown in Table 1.		L4	CO4	10 M

Table 1. State Machine

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
A	B	E	0	0
B	E	D	0	0
C	D	A	1	0
D	B	E	1	0
E	C	D	0	0

UNIT-V

10	a)	Explain propagation delay with the help of input–output waveform diagrams.	L4	CO4	5 M
	b)	Compare ECL, MOS and CMOS in terms of speed, power dissipation, and fan-out.	L2	CO4	5 M
		OR			
11	a)	Explain the working operation of Emitter Coupled Logic circuit.	L4	CO4	5 M
	b)	Discuss the operation of CMOS logic circuits and list any two advantages.	L2	CO4	5 M

PVP 23

Code: 23EE3502

III B.Tech - I Semester - Regular Examinations - NOVEMBER 2025

DIGITAL CIRCUITS
(ELECTRICAL & ELECTRONICS ENGINEERING)

Duration: 3 hours Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level CO – Course Outcome

PART – A

		BL	CO
1.a)	Show the use of don't care terms in K-map simplification.	L3	CO2
1.b)	Simplify F(A,B)=A'B+AB using Boolean theorems.	L3	CO1
1.c)	Distinguish between ROM and RAM.	L2	CO3
1.d)	Show two advantages of PLA over PAL.	L3	CO3
1.e)	State the function of a universal shift register.	L3	CO3
1.f)	Write the logic expression to convert the JK flip-flop to a T flip-flop.	L2	CO1
1.g)	Define state reduction.	L1	CO4
1.h)	Classify two methods of state assignment.	L2	CO4
1.i)	List any two features of TTL logic family.	L1	CO4
1.j)	Define power dissipation in digital ICs.	L1	CO1

PART – B

			BL	CO	Max. Marks
		UNIT-I			
2	a)	Simplify the Boolean function, (A,B,C,D) = Σ(0,1,2,5,8,9,10,14) using K-map and draw the logic circuit.	L3	CO2	5 M
	b)	Explain that the universal logic gates are commutative but not associative.	L2	CO1	5 M
		OR			
3	a)	Draw the logic circuits using AND, OR and NOT elements to represent the following expressions: i. $A\bar{B} + \bar{A}\,B$ ii. $A + B[C + D(B + \bar{C})]$	L2	CO1	5 M
	b)	Simplify the following expression using Quine McClausky method and verify using K-map. F(A,B,C,D) = Σ(0,1,2,3,4,6,8,10,12,14)	L3	CO2	5 M
		UNIT-II			
4	a)	Distinguish between a decoder and a demultiplexer.	L2	CO2	5 M
	b)	Demonstrate a 3 x 8 decoder using 2 x 4 decoders and explain its operation as a minterm generator.	L3	CO2	5 M
		OR			

5	a)	Apply the following Boolean function using 8:1 multiplexer. F(A,B,C) = Σm(1,3,5,6)	L3	CO2	5 M
	b)	Compare PROM, PAL, and PLA in terms of flexibility, hardware complexity, and applications.	L3	CO3	5 M
		UNIT-III			
6	a)	Illustrate the logic circuit of JK master slave flip flop and explain its working with the truth table.	L3	CO3	5 M
	b)	Compare ring counter and Johnson counter with examples.	L2	CO3	5 M
		OR			
7	a)	Demonstrate the truth table and state diagram of SR Flip Flop.	L2	CO3	5 M
	b)	Prepare a 3-bit asynchronous counter using JK flip-flops.	L3	CO3	5 M
		UNIT-IV			
8	a)	Construct a Moore machine for a sequence detector that detects "110".	L4	CO4	5 M
	b)	Analyze about sequential circuits, state table and state diagram.	L4	CO4	5 M
		OR			

E2

PVP23

III B.Tech I Semester Regular/ Supplementary Examinations
November 2025

Digital Circuits 23EE3502
(Electrical & Electronics Engineering)

Scheme

2	a)	K- map	2M
		Logic Circuit	3M
	b)	Explanation	5M
3	a)	(i) circuit	2M
		(ii) circuit	3M
	b)	Simplification	3M
		K-map	2M
4	a)	Differences	5M
	b)	Explanation	3M
		Diagram	2M
5	a)	Working	2M
		Diagram	3M
	b)	Comparison	5M
6	a)	Logic circuit	3M
		Truth table	2M
	b)	Comparison	5M
7	a)	Truth table	3M
		State Diagram	2M
	b)	Explanation	2M
		Diagram	3M
8	a)	Explanation	3M
		Diagram	2M
	b)	State Diagram	2M
		State Table	3M
9	a)	Explanation	3M
		State Diagram	3M
		State Reduction	4M
10	a)	Explanation	3M
		Waveforms	2M
	b)	Comparison	5M
11	a)	Explanation	3M
		Working	2M
	b)	Operation	4M
		Advantages	1M

Part A

1

a) Don't care terms (marked as 'X') in K-map simplification are used to form larger groups to achieve a more simplified Boolean expression. These terms represent input combinations for which the output is not specified, meaning they can be treated as a '1' or a '0' as needed. For Sum of Products (SOP), treat 'X' as '1' if it helps form a larger group; for Product of Sums (POS), treat 'X' as '0' if it helps reduce the expression.

b) F = (A+A') B = B

c) RAM is volatile memory for temporary data storage that is erased when power is off, while ROM is non-volatile memory used for permanent storage of system instructions that are retained when power is off.

d) Advantages of PLA: High Flexibility, Efficient Minterm Usage
Advantages of PAL: Higher Speed, Lower Cost and Simpler Implementation

e) A universal shift register's function is to perform multiple data operations, including left shifting, right shifting, and parallel loading. This versatility allows it to function as a flexible digital component for various tasks such as converting data between serial and parallel formats, acting as a data buffer, or performing arithmetic operations in digital systems.

f) J = T & K = T

g) The process of state reduction in sequential circuit design corresponds to the process of minimisation in combinational circuit design. State reduction can be done systematically with the aid of the state table and by using Caldwell's merging procedure which depends upon proving that two states are equivalent.

h) systematic, binary assignment, one-hot assignment, Gray code assignment, partition-based methods, algorithmic approaches. Any of these can be mentioned.

i) Technology, Speed, Propagation Delay, Fan-in, Fan-out, Power Dissipation, Power Consumption, Noise Margin, Supply Voltage. Mention any two.

j) This is the amount of power dissipated in an IC. It is determined by the current, ICC that it draws from the VCC supply, and is given by VCC x ICC. ICC is the average value of Icc(0) and ICC(1). This power is specified in mill watts.

2

a)

AB\CD	00	01	11	10
00	1	1		1
01		1		
11	1	1		1
10				1

Implicant Meaning Covers

B′D′ B=0, D=0 0, 2, 8, 10

B′C′ B=0, C=0 0, 1, 8, 9

A′C′D A=0, C=0, D=1 1, 5

ACD′ A=1, C=1, D=0 10, 14

F=B′D′+B′C′+A′C′D+ACD′

Logic Circuit Diagram (Description)
To draw the circuit:
Inputs
A, B, C, D
NOT Gates
Generate:
A′, B′, C′, D′

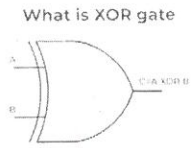
b)
Universal logic gates like NAND and NOR are commutative because the order of inputs does not change the output, and not associative because the grouping of inputs *does* change the output. For example, A NAND B=B NAND A, but (A NAND B) NAND C is not the same as A NAND (B NAND C).
Commutative property
What it means: The order of the inputs does not matter. The result is the same regardless of the input sequence.
How it applies to universal gates: For a two-input NAND gate, the output is the same whether the inputs are A NAND $B=(A\cdot B)'$ B NAND $A=(B\cdot A)'$
Since $A\cdot B$ is the same as $B\cdot A$, their complements are also the same: $(A\cdot B)'=(B\cdot A)'$
Non-associative property
What it means: The grouping of inputs changes the output. You cannot simply reorder the operations by grouping the inputs differently without changing the result.
How it applies to universal gates: For a three-input NAND gate, the output is not the same when the grouping is changed.

(A NAND B) NAND $C\neq A$ NAND (B NAND C)

This is because the intermediate output of the first operation is fed as an input to the next, and the order in which these calculations are performed affects the final result.

3
a)

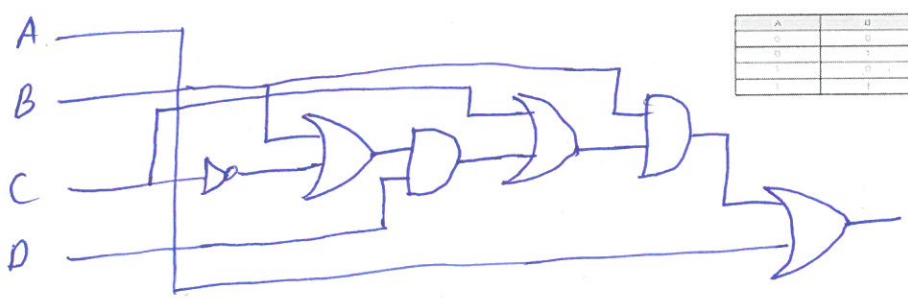
(i) AB'+A'B is XOR gate



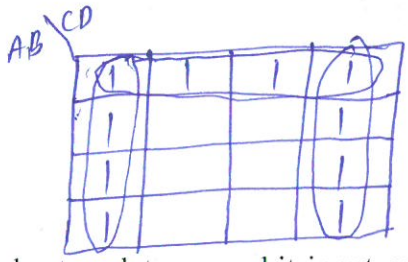
Truth Table of XOR gate

A	B	C=A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

(ii) A+B[C+D(B+C')]



b)



$F(A,B,C,D) = A'B' + C'D' + CD'$

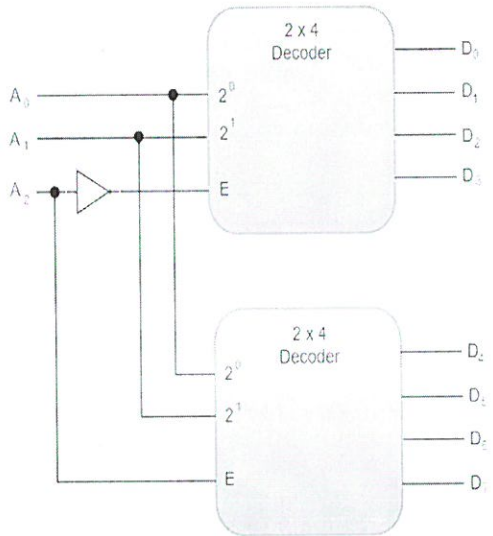
4

a)

A decoder translates an n-bit input code into a 2^n unique output, while a demultiplexer routes a single input signal to one of 2^n output lines based on n selection lines. The key difference is that a demultiplexer has a data input line and uses select lines to choose an output, whereas a decoder takes multiple control inputs that activate specific output lines without a separate data input.

Feature	Decoder	Demultiplexer
Primary Function	Translates an n-bit binary code into 2n2 to the n-th power $2n$ unique output lines.	Routes a single input signal to one of 2n2 to the n-th power $2n$ output lines.
Data Input	No separate data input; the control signals act as the input.	Has a single data input line.
Input Lines	Multiple n input lines.	One data input and n select/control lines.
Output Lines	2n2 to the n-th power $2n$ output lines.	2n2 to the n-th power $2n$ output lines.
Control/Selection	The n input lines are the control signals that activate the output.	n select lines determine which output receives the data input.
Analogy	Converts a code into a set of signals.	Distributes data to a selected output.

b)



A2	A1	A0	O/P
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

5

a)

Map Inputs to Multiplexer Selection Lines

The given Boolean function F(A,B,C)=Σ(1,3,5,6) has three variables: A, B, and C. These are connected to the 8:1 multiplexer's select lines $S2$, $S1$, and $S0$ respectively. The variable A corresponds to the most significant bit (MSB), and C corresponds to the least significant bit (LSB), determining which of the eight inputs (I_0 to I_7) is selected.

Determine Data Input Connections

The function is high (logic 1) for the minterms $m1$,3,$m5$, and $m6$. The multiplexer data inputs corresponding to these minterms are connected to logic HIGH (1), while all other inputs are connected to logic LOW (0).

Configure the Multiplexer

The connections for the 8:1 multiplexer are as follows:

- Connect the select lines $S2$, $S1$, $S0$ to A, B, .
- Connect the data inputs $I1$, $I3$, $I5$, and $I6$ to a logic HIGH signal (e.g., VCC).
- Connect the data inputs $I0$, $I2$, $I4$, and $I7$ to a logic LOW signal (e.g., GND).
- The multiplexer output Y will produce the function (A,,).

To apply the function F(A,B,C)=Σ(1,3,5,6) using an 8:1 multiplexer, connect the select lines $S2$, $S1$, and $S0$ to inputs A, B, and C respectively. The data inputs $I1$, $I3$,5, and $I6$ must be connected to logic HIGH (1), while $I0$, $I2$, $I4$, and $I7$ must be connected to logic LOW (0). The output of the multiplexer will then represent the desired function.

b)

PROM, PAL, and PLA are all programmable logic devices used to implement digital circuits, primarily differing in their internal architecture (fixed vs. programmable AND/OR arrays), which impacts their flexibility, hardware complexity, and applications.

Attribute	PROM (Programmable Read-Only Memory)	PAL (Programmable Array Logic)	PLA (Programmable Logic Array)
AND Array	Fixed	Programmable	Programmable
OR Array	Programmable	Fixed	Programmable
Flexibility	Limited (only some functions with all input variables)	Moderate (limited by fixed OR array)	High (both arrays programmable, can implement any SOP function efficiently)
Hardware Complexity	Simple (uses a basic decoder)	Less complex than PLA	High (due to two programmable arrays)
Cost	Low cost	Low cost (more available)	High cost (less available)
Speed	Slow (requires full decoding)	Faster than PLA	Slower than PAL
Applications	Storing permanent firmware, boot programs, and lookup tables in embedded systems and microcontrollers	Implementing simple to moderate combinational logic functions, such as decoders and state machines	Complex logic functions, control over datapath, and as a counter, decoder, or BUS interface in programmed I/O

6

a)

A JK master-slave flip-flop uses two JK flip-flops, the "master" and the "slave," to avoid the race-around condition. The master flip-flop responds to the inputs (J, K, and the clock) while the clock is high, and the slave flip-flop is inactive. When the clock goes low, the slave responds to the master's output, transferring the new state to the final output (Q).

Logic circuit

The circuit consists of two JK flip-flops connected in series.

- Master Flip-Flop: Receives the direct clock signal and the J and K inputs. Its output is connected to the inputs of the slave flip-flop.

- Slave Flip-Flop: Receives the inverted clock signal (via an inverter), so it is active when the clock is low. Its inputs are the outputs from the master flip-flop, and its outputs are the final Q and $\bar{Q}$ outputs.

Working principle

1. Clock is HIGH: The master flip-flop is enabled, and it captures the J and K inputs, changing its internal state accordingly. The slave flip-flop remains inactive because its clock input is low. The output Q does not change during this phase.
2. Clock is LOW: The master flip-flop becomes inactive. The slave flip-flop is now enabled and copies the state of the master flip-flop's output. This new state becomes the final output Q. This two-step process prevents the output from changing multiple times within a single clock cycle, which eliminates the race-around condition where the output would toggle continuously when J=K=1.

Truth table

The truth table for a JK master-slave flip-flop is identical to a standard JK flip-flop, as the master-slave configuration is a way to *implement* a JK flip-flop without race-around conditions. The output Qn+1represents the next state of the flip-flop based on the current inputs J, K, and the previous Qn.

J	K	Qn	Qn+1	**Operation**
0	0	0	0	No Change
0	0	1	1	No Change
0	1	0	0	Reset
0	1	1	0	Reset
1	0	0	1	Set
1	0	1	1	Set
1	1	0	1	Toggle
1	1	1	0	Toggle

b)

A ring counter circulates a single high/low state through the flip-flops, while a Johnson counter (twisted ring counter) feeds the inverted output of the last flip-flop back to the first. This key difference means a ring counter with 'n' flip-flops has 'n' states, while a Johnson counter has 2n states. Examples show a 4-bit ring counter has 4 states, whereas a 4-bit Johnson counter has 8 states, making it more efficient for generating a larger sequence with fewer components.

Feature	**Ring Counter**	**Johnson Counter**
Feedback Connection	Output of the last flip-flop is fed back to the input of the first flip-flop.	Inverted output of the last flip-flop is fed back to the input of the first.
Number of States (for 'n' flip-flops)	'n' states (equal to the number of flip-flops).	'2n' states (twice the number of flip-flops).

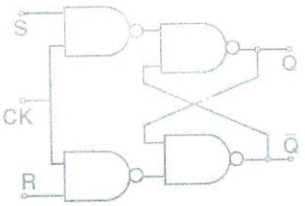
Efficiency	Less efficient; uses 'n' flip-flops for 'n' states.	More efficient; uses 'n' flip-flops for '2n' states.
Example: 4-bit	Has 4 states.	Has 8 states.
Applications	Simple sequence generation, frequency division (mod-n counters).	Digital clocks, frequency dividers, pattern generators, complex sequence generators.

Ring counter example (4-bit)

- **Setup:** A single '1' is loaded into the first flip-flop (e.g., 1000).
- **States:** The '1' shifts from one flip-flop to the next with each clock pulse, creating a sequence of states:
  - 1000
  - 0100
  - 0010
  - 0001

7

a)



Truth Table

S	R	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

State Diagram

The state diagram illustrates the transitions between the two possible states (0 and 1) based on the inputs S and R.

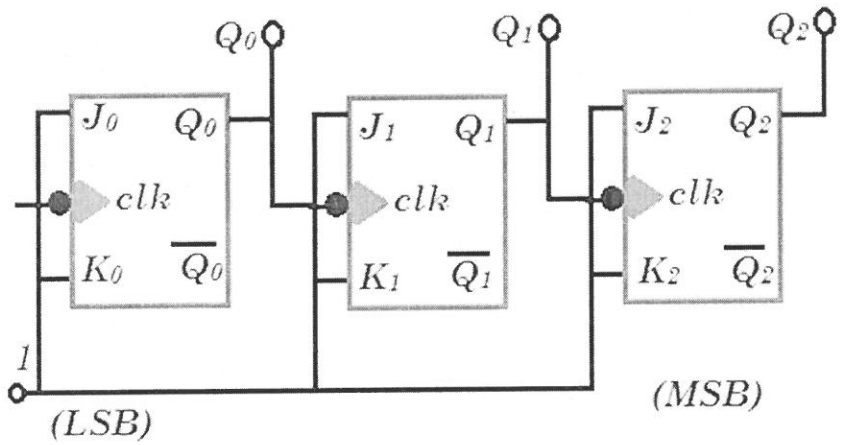
- States: The states are represented by circles, with the current output value indicated inside the circle.
- Transitions: The arcs between the states represent the transitions, and the input combination that causes the transition is labeled on the arc.

Example State Diagram:

- State 0 (Current Output Q=0)
  - If S=0,=0, the state remains 0 (Hold).
  - If S=1,=0, it transitions to State 1 (Set).
  - If S=0,=1, it remains in State 0 (Reset).
- State 1 (Current Output Q=1)
  - If S=0,=0, the state remains 1 (Hold).
  - If S=0,=1, it transitions to State 0 (Reset).
  - If S=1,=0, it remains in State 1 (Set).
- Invalid State (S=1,=1): This combination is not shown as a state transition because it leads to an indeterminate or invalid state and should be avoided.

b)

A 3-bit asynchronous counter is built with three JK flip-flops, where the clock input is applied to the first flip-flop (LSB), and the clock inputs of the subsequent flip-flops are connected to the output of the preceding flip-flop. For an up-counter, all J and K inputs are set to logic "1" to make each flip-flop function as a toggle flip-flop



Circuit diagram and connections

- Flip-Flop 1 (FF0 - LSB):
  - Clock Input: Connected to the external clock pulse.
  - J and K Inputs: Connected to logic "1" (or the positive voltage rail) to ensure it toggles on every clock pulse.
  - Output: Q0.
- Flip-Flop 2 (FF1):
  - Clock Input: Connected to the output Q0 of FF0.
  - J and K Inputs: Connected to logic "1".
  - Output: Q1.
- Flip-Flop 3 (FF2 - MSB):
  - Clock Input: Connected to the output Q1 of FF1.
  - J and K Inputs: Connected to logic "1".
  - Output: Q2.

How it works

- The counter is asynchronous because the clock for each flip-flop is not connected to a single, common clock signal.
- The first flip-flop (FF0) toggles its output on every clock pulse because its J and K inputs are held high.
- The second flip-flop (FF1) is triggered by the negative edge of the Q0 output of FF0. Since its J and K inputs are also high, it toggles whenever Q0 transitions from 1 to 0.
- Similarly, the third flip-flop (FF2) toggles its output on every negative edge of the Q1 output of FF1.
- This creates a ripple effect, where the state change "ripples" through the flip-flops, resulting in a binary count of 000, 001, 010, 011, 100, 101, 110, 111, and back to 000

8

a)

A Moore machine for a sequence detector that detects the binary sequence "110" can be designed with four states:

State	Output (Detect = 1)	Description
S0	0	Initial state / no relevant prefix detected yet
S1	0	"1" detected
S2	0	"11" detected
S3	1	"110" detected

State Transition Diagram

The transitions between states are determined by the current input bit:

- **From S0 (Start):**
  - If input is **0**, stay in S0.
  - If input is **1**, go to S1.
- **From S1 ("1" seen):**
  - If input is **0**, go back to S0 (sequence broken).
  - If input is **1**, go to S2 ("11" seen).
- **From S2 ("11" seen):**
  - If input is **0**, go to S3 ("110" seen).
  - If input is **1**, stay in S2 (still a "11" prefix for the *next* potential sequence).
- **From S3 ("110" seen):**
  - Regardless of input (0 or 1), return to S0 for the next clock cycle in a non-overlapping detector, or transition based on the new input for an overlapping detector (as is common for state machines):
  - If input is **0**, go to S0.
  - If input is **1**, go to S1.

State Transition Table

The behavior of the machine can be summarized in the following transition table:

Current State	Input 0	Input 1	Output
S0	S0	S1	0
S1	S0	S2	0
S2	S3	S2	0
S3	S0	S1	1

In this Moore machine, the output is solely dependent on the current state. When the machine enters state S3, the output becomes 1, indicating that the sequence "110" was detected on the immediately preceding input stream. The machine then transitions from S3 in the next clock cycle based on the new input.

b)

A state table and a state diagram are two methods for representing the behavior of sequential circuits, with the diagram providing a visual model and the table providing a detailed, structured one. A state diagram uses circles to represent states and arrows to show the transitions between them, with labels for the inputs and outputs that trigger the transitions.

State table

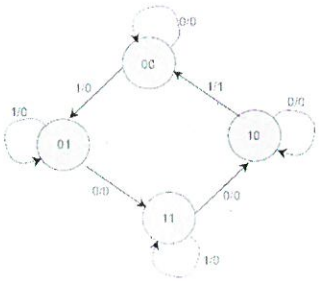
- **Purpose**: To provide a complete, tabular representation of all possible state transitions and outputs.
- **Structure**: A matrix where columns typically represent the current state and inputs, and rows represent the next state and outputs.
- **Components**:
  - **Present State**: The current state of the circuit.
  - **Inputs**: The inputs that affect the circuit's next state and outputs.
  - **Next State**: The state the circuit will transition to after the current clock cycle.
  - **Outputs**: The outputs produced by the circuit in the current state and for the current inputs.
- **Advantage**: Ensures that every possible transition is accounted for, which can be useful for detailed analysis and circuit design.

State diagram

- **Purpose**: To provide a graphical, visual representation of the system's behavior.
- **Structure**: A directed graph.
- **Components**:
  - **States**: Represented by circles or bubbles.
  - **Transitions**: Represented by directed arrows connecting the states.
  - **Input/Output**: Labeled on the arrows, indicating the input that causes the transition and the output produced.
- **Advantage**: Allows for a quick and intuitive understanding of how the system moves from one state to another, making it easier to visualize the overall logic.

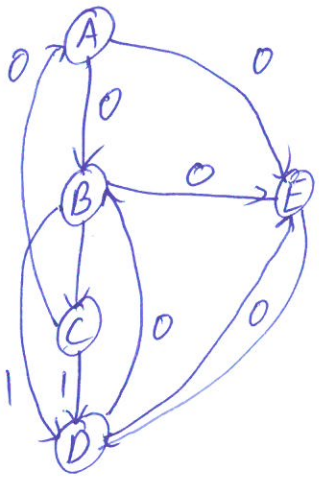
Relationship between the two

- A state diagram is typically derived from a state table, or vice versa.
- The information presented is the same in both, but the format is different. The table provides a detailed, explicit list, while the diagram provides a visual map of the flow.



9

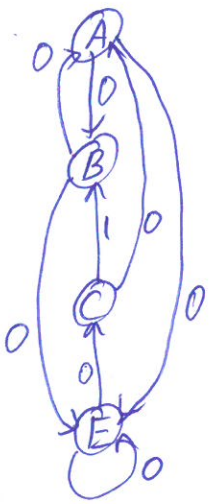
To find the equivalent partition and reduced table, you first group states based on identical outputs for each input, creating an initial partition. Then, you iteratively refine this partition by checking if states within the same group transition to states that are still in the same group in the previous partition. This process continues until no further changes can be made, at which point you have the final equivalence partition. Finally, you create the reduced table by representing each group as a single state and mapping transitions accordingly.



State reduction

A	B	E	0	0
B	E	D	0	0
C	D	A	1	0
D	B	E	1	0
E	C	D	0	0

Delete state D



A	B	E	0	0
B	E	A	0	0
C	B	A	1	0
E	C	~~D~~E	0	0

10

a)

Propagation delay is the time it takes for a signal to travel from its source to its destination. It can refer to the time it takes for a signal to travel through a physical medium, such as a cable, or the time it takes for an output of an electronic circuit to change after a change at its input. The delay is caused by the physical limitations of the transmission medium and is a critical factor in fields like computer networking and electronics.

In communication and electronics

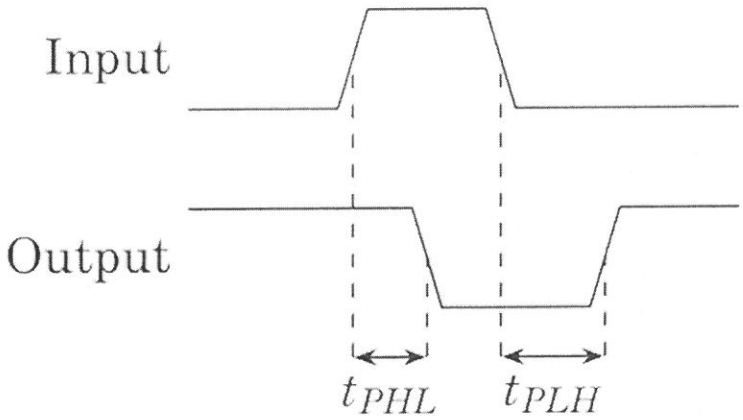
- Physical travel: In a network, it's the time from when a bit is sent until it is received.
- Logic gates: In a digital circuit, it's the time it takes for the output of a logic gate to change after its input has changed. This delay is due to the time required for the internal transistors to switch.
- Limit: This is why, for example, you see a lightning flash before you hear the thunder; the light travels faster than the sound, and both experience propagation delay, says Tech Target.

Factors that influence propagation delay

- Distance: The farther the signal must travel, the longer the delay.
- Medium: The material the signal travels through affects its speed. For example, signals travel at different speeds in copper cables, fiber optics, or air.
- Circuit complexity: In electronics, a signal passing through more components or more complex paths will have a higher overall propagation delay.

Calculation

- The fundamental limit is the speed of light in a vacuum. However, in a physical medium, the speed is less than c.
- The formula is: Propagation Delay (s) = Distance (m) / Propagation Speed (m/s).



b)

ECL is the fastest but consumes the most power, while CMOS is the slowest but has the lowest power dissipation. MOS is a category that includes CMOS and NMOS, but when compared directly, CMOS offers lower power and higher fan-out compared to ECL. ECL has high speed because its transistors don't operate in saturation, but this results in high power consumption. CMOS has low power dissipation due to its design, where there is no direct current path from power to ground in a stable state, and it has a high fan-out due to its high input impedance.

Feature	ECL	CMOS	MOS
Speed	Fastest	Moderate to high, but generally slower than ECL	Varies, but often slower than ECL. CMOS is a type of MOS.
Power Dissipation	Highest	Lowest	Varies; CMOS is the most power-efficient within the MOS family.
Fan-out	High, but less than CMOS	Highest	High

Detailed Comparison:

- Speed:
  - ECL: Fastest, with propagation delays around 0.75 to 2 ns. This is because transistors do not saturate, allowing for quick switching.
  - CMOS: Slower than ECL, but speeds have increased with advancements in transistor size and technology.
  - MOS: Speed varies depending on the specific implementation, but generally falls between ECL and CMOS or is similar to CMOS.
- Power Dissipation:
  - ECL: Highest power consumption due to continuous current flow through the transistors, even in static states. For example, a single ECL gate can consume about 60 mW, compared to 0.01 mW for a CMOS gate.
  - CMOS: Lowest power dissipation. When a CMOS gate is static (no switching), there is no direct current path from the power supply to ground, resulting in minimal power consumption.
  - MOS: Varies depending on the implementation. CMOS is the most power-efficient design within the MOS family, which also includes NMOS (which has higher power consumption).
- Fan-out:
  - ECL: High fan-out, but lower than CMOS.
  - CMOS: Highest fan-out due to its high input impedance, which draws very little current from the output of the preceding gate.
  - MOS: High fan-out is a key advantage, particularly for CMOS.

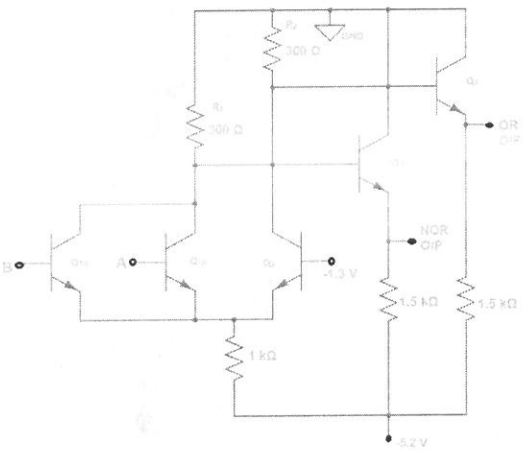
TABLE 47.5 *Comparison of Logic Families*

Parameter \ Logic Family	DTL	TTL	ECL	IIL	MOS	CMOS
Propagation Delay Time in ns	30	9	1	1	50	< 50
Power dissipation, per gate in mW	8-12	10	50	0.1	0.1	0.01
Noise margin in volt	0.3	0.4	0.25	0.35	1.5	5
Fan-in	8	8	5	5		10
Fan-out	5	10	10	8	10	50
Cost	Low	Low	High	Very low	Low	Low

11

a)

Emitter-Coupled Logic (ECL) operates by steering a constant current between two parallel bipolar junction transistors (BJTs) based on the input voltage, a principle known as current mode logic. By using transistors in a non-saturating mode, ECL achieves very high speed but consumes more power. When an input is a logic HIGH, current is directed through the corresponding transistor, turning the other one OFF; when all inputs are a logic LOW, current flows through the reference transistor, turning the input transistors OFF.

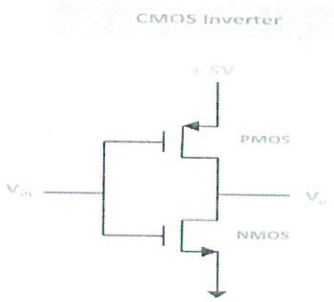


Working

- **Core components:** An ECL gate consists of a differential amplifier with two or more input transistors whose emitters are connected together, and a constant current source. A constant current is always forced to flow through the common emitter resistor (RE) and one of the parallel transistors.
- **Logic LOW input:** When a logic LOW is applied to an input, the base voltage is too low to forward-bias the corresponding transistor, keeping it OFF. The current from the constant current source is then steered through the reference transistor.
- **Logic HIGH input:** When a logic HIGH is applied, it raises the base voltage of the corresponding input transistor above the reference voltage (VR). This turns the input transistor ON, and because the total current is constant, the reference transistor is effectively turned OFF.
- **Non-saturating operation:** Unlike TTL logic, ECL transistors are never allowed to enter saturation. The constant current ensures that the transistors are always operating in their active region, which eliminates storage time delays and allows for very fast switching speeds.

- **Complementary outputs:** Each gate produces both an OR output and an inverted NOR output by tapping the collector voltages of the two transistors in the differential pair.
- **Output stage:** The collector voltages are fed into emitter-follower output transistors, which provide low output impedance and a stable logic level to drive subsequent gates.

b)



CMOS logic circuits operate by using two complementary sets of transistors: a pull-up network of p-type metal-oxide-semiconductor (PMOS) transistors and a pull-down network of n-type metal-oxide-semiconductor (NMOS) transistors. One network is always on while the other is off, preventing a direct path between power and ground and thus resulting in low power consumption. Two main advantages of CMOS circuits are their **low static power consumption** and **high noise immunity**.

Operation of CMOS logic circuits

- **Complementary Transistors:** A CMOS gate consists of a pull-up network of PMOS transistors and a pull-down network of NMOS transistors.
- **Pull-up Network:** The PMOS transistors are arranged so that they turn on when the input is low (logic 0), connecting the output to the positive power supply (VDD).
- **Pull-down Network:** The NMOS transistors are arranged so that they turn on when the input is high (logic 1), connecting the output to ground (VSS).
- **Complementary Action:** When the input is high, the PMOS transistors in the pull-up network are off, and the NMOS transistors in the pull-down network are on, pulling the output low. Conversely, when the input is low, the PMOS transistors are on, and the NMOS transistors are off, pulling the output high.
- **Low Power Consumption:** Because the pull-up and pull-down networks are complementary, there is no direct path from VDD to VSS in a steady state (static condition). Power is only dissipated during the brief moment when the circuit switches from one state to the other.

Two advantages of CMOS circuits

- **Low Static Power Consumption:** CMOS circuits consume very little power when idle because there is no through-current from the power supply to ground in a steady state. This makes them highly energy-efficient and ideal for battery-powered devices.
- **High Noise Immunity:** CMOS circuits are less susceptible to noise on the input signals compared to other logic families. The large voltage swing of the output (from VSS to VDD) provides a significant margin, making the logic states more distinct and the circuit more robust against interference.