Design and Analysis of Universality of NAND and NOR Gate, Synchronous Random Counters,4-Bit Logic Units and Universal Logic Gates.

Presented by Tashnin Khan Tanisha (23201112), Nowrin Akhter Riya (23201117), Fahmida Tabassum (23201132), Fariya Rahman (23201136)

Universality of NAND and NOR Gate

This experiment highlights how NAND and NOR gate perform all basic logic operations, establishing their universality in digital logic circuit design .

Did you know?
Every digital device we use from calculator to computers! Is built using universal gates like NAND and NOR!

All logic operations-AND,OR,NOT can be built using just NAND or NOR gates. Universality at it's finest!

TRUTH TABLE

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

AND

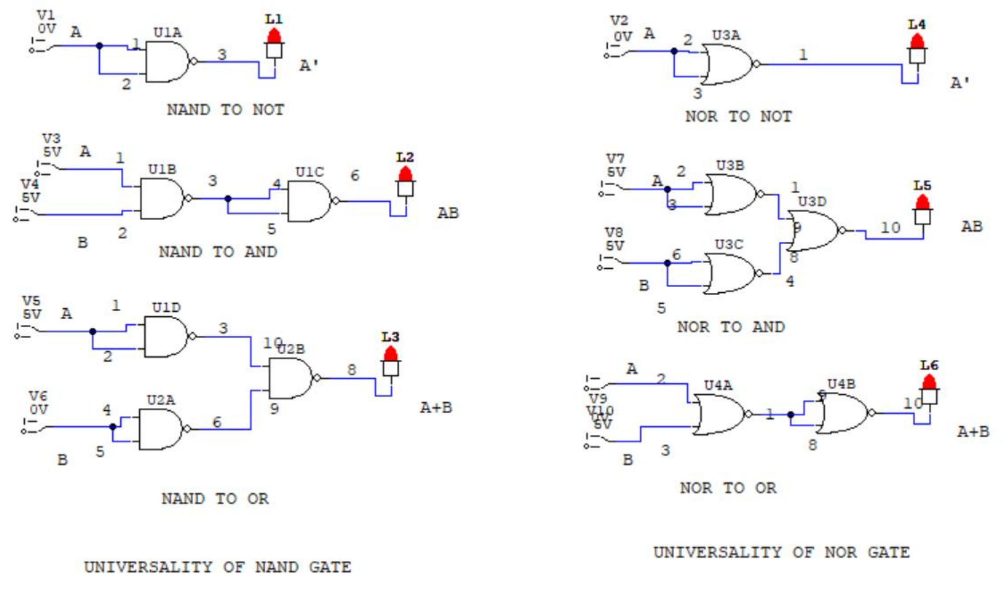
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

OR

A	Y
0	1
1	0

NOT

CIRCUIT DIAGRAM



- ✓ AND waits for both 1s
- ✓ OR needs just one
- ✓ NOT loves to flip the game

From truth tables to glowing results the power of universal gates!

Synchronous Random Counter

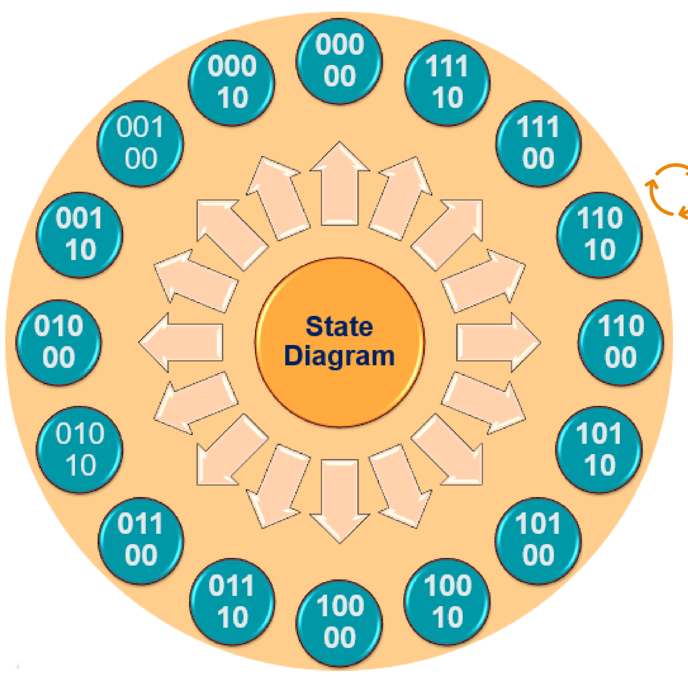
A 5-bit synchronous counter that counts even numbers **-30, 28, 26, 24, 22, 20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 0.**, step by step, in reverse!

Each clock pulse brings a perfectly timed descent in the digital sequence. Precision, control, and rhythm — that's synchronous logic in action!

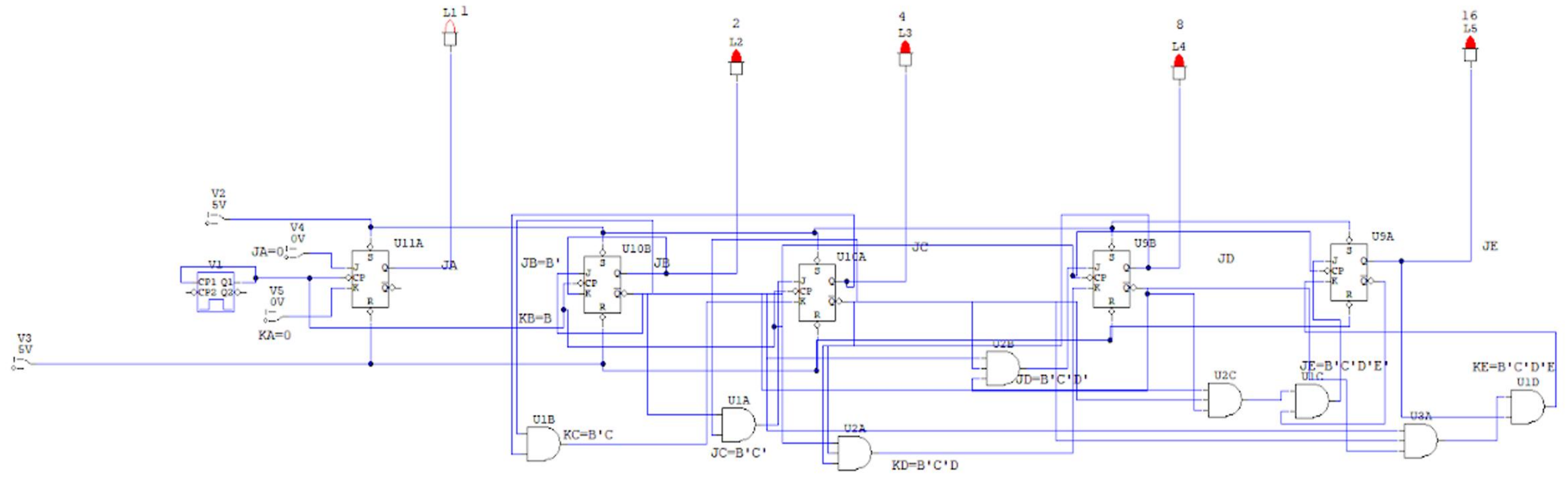
Configure the flip-flops — JA=0, KA=0; JB=B', KB=B; JC=B'C', KC=B'C; JD=B'C'D', KD=B'C'D; JE=B'C'D'E', KE=B'C'D'E., And watch it do wonders!

TRUTH TABLE

Q1	Qn +1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



CIRCUIT DIAGRAM



4-Bit Logic Units and Universal Logic Gates

This experiment demonstrates a 4-bit logic unit where AND,OR,NOT and XOR operations are implemented using multiplexers (MUX),showing how MUX can be used to realize multiple logic functions efficiently.

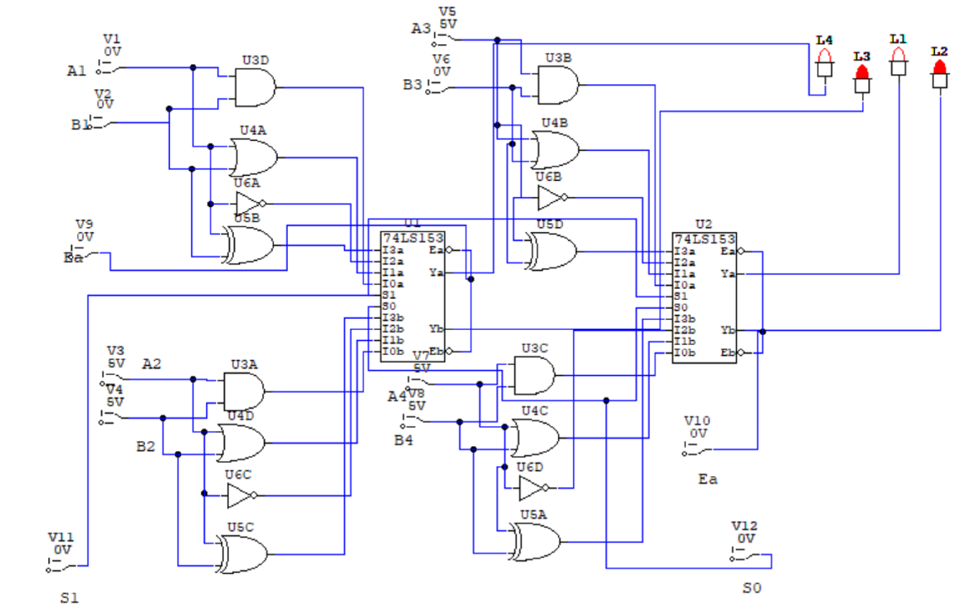
This unit emphasizes the role of MUX is simplifying and optimizing digital circuits. It highlights modularity, versatility, and compactness in digital circuit design.

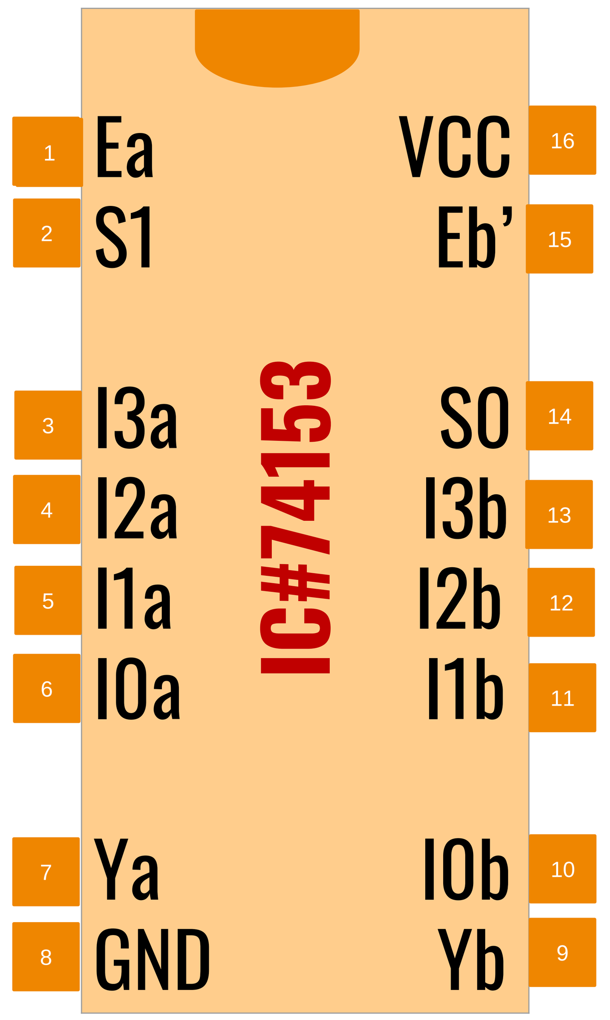
TRUTH TABLE

S1	S2	Q
0	0	I0
0	1	I1
1	0	I2
1	1	I3

- ✓ S1 = 0, S2 = 0 ⟶ **AND:** Boom! It's AND!
- ✓ S1 = 0, S2 = 1 ⟶ **OR:** Flip the switch, it's OR!
- ✓ S1 = 1, S2 = 0 ⟶ **NOT:** Invert and there you go, NOT!
- ✓ S1 = 1, S2 = 1 ⟶ **XOR:** Two inputs differ? That's XOR!

CIRCUIT DIAGRAM





PIN CONFIGURATION