DIGITAL LOGIC & DESIGN

ALYSIA NORALES | CAMERON TILLETT

PROBLEM STATEMENT

YOU HAVE BEEN ENLISTED TO DESIGN A SODA MACHINE DISPENSER FOR THE IT DEPARTMENT STUDENTS' LOUNGE. SODAS ARE PARTIALLY SUBSIDIZED BY THE STUDENT CHAPTER OF THE ACM, SO THEY COST ONLY 25 CENTS EACH. THE MACHINE ACCEPTS 5 CENTS, 10 CENTS, AND SHILLINGS.

WHEN ENOUGH COINS HAVE BEEN INSERTED, IT DISPENSES THE SODA AND RETURNS ANY NECESSARY CHANGE. DESIGN AN FSM CONTROLLER FOR THE SODA MACHINE. THE FSM INPUTS ARE 5 CENT, 10 CENT, AND SHILLING, INDICATING WHICH COIN WAS INSERTED. ASSUME THAT EXACTLY ONE COIN IS INSERTED ON EACH CYCLE. THE OUTPUTS ARE DISPENSE, RETURN5CENTS, RETURN10CENTS, AND RETURNTWO10CENTS. WHEN THE FSM REACHES 25 CENTS, IT ASSERTS DISPENSE AND THE NECESSARY RETURN OUTPUTS REQUIRED TO DELIVER THE APPROPRIATE CHANGE.

THEN IT SHOULD BE READY TO START ACCEPTING COINS FOR ANOTHER SODA.

WE ARE TO PROVIDE

THE MODIFIED FSM (SHOWING 5 CENTS, 10 CENTS AND SHILLINGS INSTEAD OF NICKELS, DIMES, AND QUARTERS)

THE STATE ENCODING TABLE
[HINT: USE 10 BITS TO REPRESENT EACH OF THE 10 STATES (S0 - S45).
USE A ONE-HOT ENCODING SCHEME WITH S0 = 0000000001 AND S45 = 1000000000

THE STATE TRANSITION TABLE AND THE RESULTING SIMPLIFIED EQUATIONS

THE OUTPUT TABLE AND THE RESULTING SIMPLIFIED EQUATIONS

THE CIRCUIT

TABLE OF CONTENTS

FINITE STATE MACHINE

FSMS: STAND FOR FINITE STATE MACHINES

- IT IS AN ABSTRACT MODEL OF COMPUTATION THAT IS USED TO MODEL LOGIC.

A LANGUAGE IS CONSIDERED REGULAR IF — AND ONLY IF — IT CAN BE RECOGNISED BY A FSM.

- FSMS CONSIST OF A FINITE NUMBER OF STATES THAT ARE REPRESENTED BY CIRCLES

- FSMS PROVIDE A STRAIGHT-FORWARD IMPLEMENTATION PLAN

FINITE STATE MACHINE

CHARACTERISTICS

- SET OF STATES
- A SET OF DIRECTED EDGES
- INITIAL/SINGLE STATE
- SET OF FINAL STATES

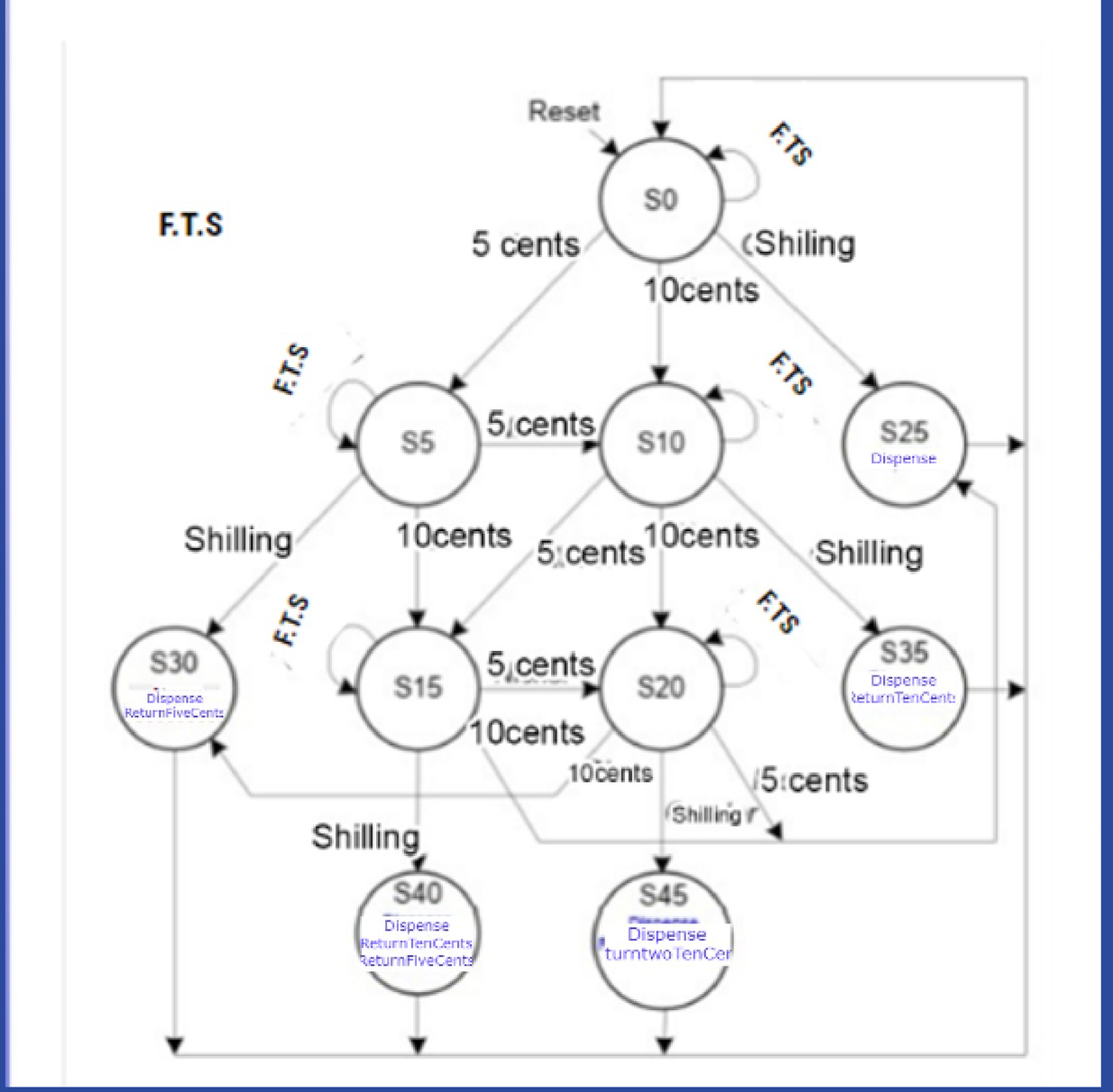


DIAGRAM OF FINITE STATE MACHINE

ONE-HOT ENCODING

ONE-HOT ENCODING IS AN ALTERNATIVE STATE ASSIGNMENT METHOD THAT ATTEMPTS TO MINIMIZE THE COMBINATIONAL LOGIC BY INCREASING THE NUMBER OF FLIP-FLOPS

THE GOAL OF THE METHOD IS TO TRY TO REDUCE THE NUMBER OF CONNECTIONS BETWEEN THE LOGIC GATES IN THE COMBINATIONAL CIRCUIT OF THE FSM.

ONE HOT ENCODING HAS BEEN USED IN THE PROCESS OF CATEGORIZING DATA VARIABLES SO THEY CAN BE USED IN MACHINE LEARNING ALGORITHMS TO MAKE BETTER PREDICTIONS.

STATE ENCODING

STATE ENCODING ASSIGNS A UNIQUE PATTERN OF ONES AND ZEROS TO EACH DEFINED STATE OF A FINITE-STATE MACHINE (FSM).

WITH BINARY ENCODING, EACH STATE IS REPRESENTED AS A BINARY NUMBER. BECAUSE K BINARY NUMBERS CAN BE REPRESENTED BY LOG2K BITS, A SYSTEM WITH K STATES NEEDS ONLY LOG2K BITS OF STATE. IN ONE-HOT ENCODING, A SEPARATE BIT OF STATE IS USED FOR EACH STATE.

DIAGRAM OF STATE ENCODING TABLE

STATE	ENCODING S9:0
S0	0000000001
S5	0000000010
S10	0000000100
S25	0000001000
S30	0000010000
S15	0000100000
S20	0001000000
S35	0010000000
S40	0100000000
S45	1000000000

DIAGRAM OF NUMBERED STATE ENCODING TABLE

Number	STATE	ENCODING S9:0
S0	S0	0000000001
S1	S5	0000000010
S2	S10	0000000100
S3	S25	0000001000
S4	S30	0000010000
S5	S15	0000100000
S6	S20	0001000000
S7	S35	0010000000
S8	S40	0100000000
S9	S45	1000000000

STATE TRANSITION

A STATE TRANSITION TABLE REPRESENTS A FINITE STATE MACHINE IN A TABLE FORMAT. INSTEAD OF DRAWING STATES AND TRANSITIONS IN A STATEFLOW CHART, WE USE A STATE TRANSITION TABLE TO MODEL A STATE MACHINE IN A CONCISE, COMPACT FORMAT THAT REQUIRES MINIMAL MAINTENANCE OF GRAPHICAL OBJECTS.

STATE-TRANSITION DIAGRAMS DESCRIBE ALL OF THE STATES THAT AN OBJECT CAN HAVE, THE EVENTS UNDER WHICH AN OBJECT CHANGES STATE (TRANSITIONS), THE CONDITIONS THAT MUST BE FULFILLED BEFORE THE TRANSITION WILL OCCUR, AND THE ACTIVITIES UNDERTAKEN DURING THE LIFE OF AN OBJECT.

DIAGRAM OF STATE TRANSITION TABLE

CURRENT STATE s	Input			NEXT STATE s'
	F	T	S	
S0	0	0	0	S0
S0	0	0	1	S25
S0	0	1	0	S10
S0	1	0	0	S5
S5	0	0	0	S5
S5	0	0	1	S30
S5	0	1	0	S15
S5	1	0	0	S10
S10	0	0	0	S10

CURRENT STATE s	Input			NEXT STATE s'
	F	T	S	
S10	0	0	1	S35
S10	0	1	0	S20
S10	1	0	0	S15
S25	X	X	X	S0
S30	X	X	X	S0
S15	0	0	0	S15
S15	0	0	1	S40
S15	0	1	0	S25
S15	1	0	0	S20
S20	0	0	0	S20
S20	0	0	1	S45
S20	0	1	0	S30
S20	1	0	0	S25
S35	X	X	X	S0
S40	X	X	X	S0
S45	X	X	X	S0

DIAGRAM OF STATE TRANSITION TABLE

DIAGRAM OF STATE TRANSITION TABLE

CURRENT STATE s	Input			NEXT STATE s'
	F	T	S	
0000000001	0	0	0	0000000001
0000000001	0	0	1	0000001000
0000000001	0	1	0	0000000100
0000000001	1	0	0	0000000010
0000000010	0	0	0	0000000010
0000000010	0	0	1	0000010000
0000000010	0	1	0	0000100000
0000000010	1	0	0	0000000100
0000000100	0	0	0	0000000100
0000000100	0	0	1	0010000000
0000000100	0	1	0	0000000100
0000000100	1	0	0	0010000000
0000000100	X	X	X	0000000001
0000010000	X	X	X	0000000001

0000100000	0	0	0	0000100000
0000100000	0	0	1	0100000000
0000100000	0	1	0	0000001000
0000100000	1	0	0	0001000000
0001000000	0	0	0	0001000000
0001000000	0	0	1	1000000000
0001000000	0	1	0	0000010000
0001000000	1	0	0	0000001000
0010000000	X	X	X	0000000001
0100000000	X	X	X	0000000001
1000000000	X	X	X	0000000001

OUTPUT TABLES

THE TRUTH TABLE OF A LOGIC SYSTEM (E.G. DIGITAL ELECTRONIC CIRCUIT) DESCRIBES THE OUTPUT(S) OF THE SYSTEM FOR GIVEN INPUT(S). THE INPUT(S) AND OUTPUT(S) ARE USED TO LABEL THE COLUMNS OF A TRUTH TABLE, WITH THE ROWS REPRESENTING ALL POSSIBLE INPUTS TO THE CIRCUIT AND THE CORRESPONDING OUTPUTS.

WHAT IS OUTPUT IN LOGIC GATES?
LOGIC GATES ARE THE FUNDAMENTAL BUILDING BLOCKS OF DIGITAL ELECTRONIC CIRCUITS.

OUTPUT TABLES

THE OUTPUT IS "TRUE" IF EITHER OR BOTH OF THE INPUTS ARE "TRUE." IF BOTH INPUTS ARE "FALSE," THEN THE OUTPUT IS "FALSE." IN OTHER WORDS, FOR THE OUTPUT TO BE 1, AT LEAST INPUT ONE OR TWO MUST BE 1. THIS WOULD CREATE AN OR GATE.

DIAGRAM OF OUTPUT TABLE

STATE	DISPENSE	RETURNFIVECENTS	RETURNTENCENTS	RETURNTWOTENCENTS
S5	X	X	X	X
S10	X	X	X	X
S15	X	X	X	X
S20	X	X	X	X
S25	1	X	X	X
S30	1	1	X	X
S25	1	X	1	X
S40	1	1	1	X
S45	1	X	X	1

STATE ENCODING	DISPENSE	RETURNFIVECENTS	RETURNTENCENTS	RETURNTWOTENCENTS
0000001000	1	X	X	X
0000010000	1	1	X	X
0010000000	1	X	1	X
0100000000	1	1	1	X
1000000000	1	X	X	1

OUTPUT TABLE EQUATION

$$S_9 = S_6 S$$
$$S_8 = S_5 S$$
$$S_7 = S_2 S$$
$$S_6 = S_2 T + S_5 + S_6 \overline{F}\,\overline{T}\,\overline{S}$$
$$S_5 = S_1 T + S_2 + S_5 F T S$$
$$S_4 = S_1 S + S_6 T$$
$$S_3 = S_0 S + S_5 T + S_6 F$$
$$S_2 = S_0 T + S_1 F + S_2 \overline{F}\,\overline{T}\,\overline{S}$$
$$S_1 = S_0 F + S_1 F T S$$
$$S_0 = S_0 \overline{F}\,\overline{T}\,\overline{S} + S_3 + S_4 + S_7 + S_8 + S_9$$

DISPENSE $S_3 + S_4 + S_7 + S_8 + S_9$

RETURNFIVECENTS $S_4 + S_8$

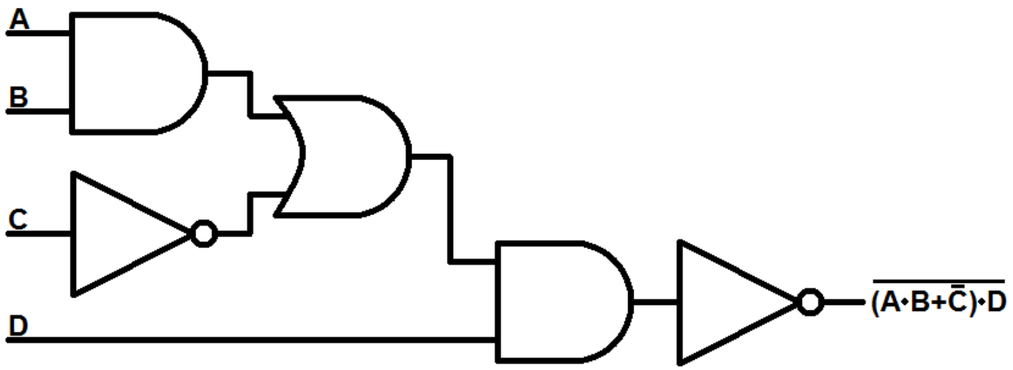
RETURNTENCENTS $S_7 + S_8$

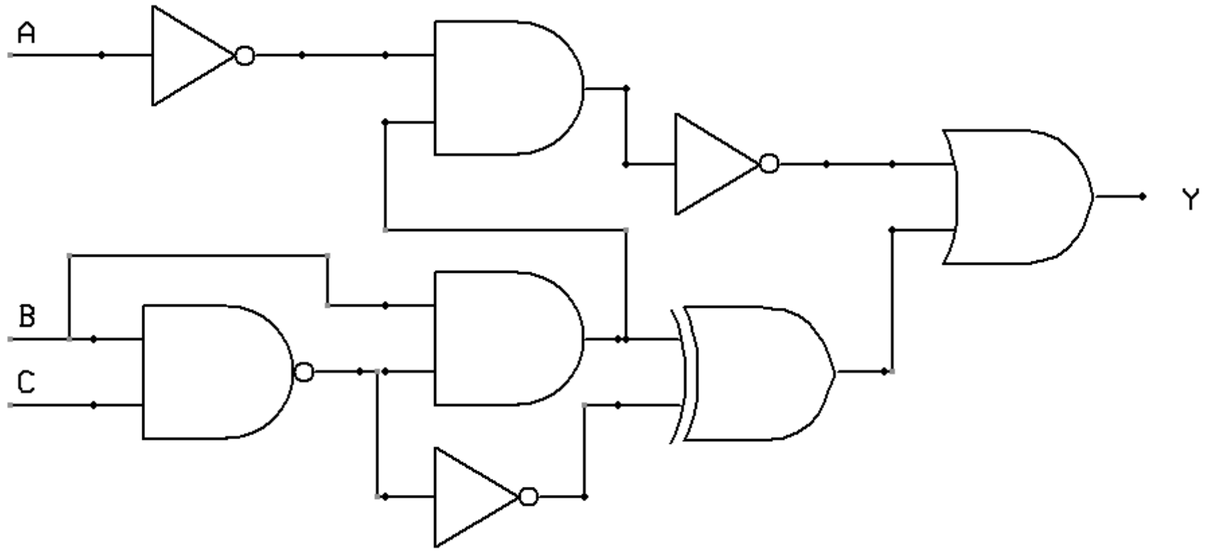
RETURNTWOTENCENTS S_9

CIRCUIT DIAGRAMS

A DIGITAL CIRCUIT IS A COLLECTION OF INTERCONNECTED DIGITAL COMPONENTS CALLED GATES.

THESE CIRCUITS ARE GRAPHICAL REPRESENTATION OF A PROGRAMS LOGIC.



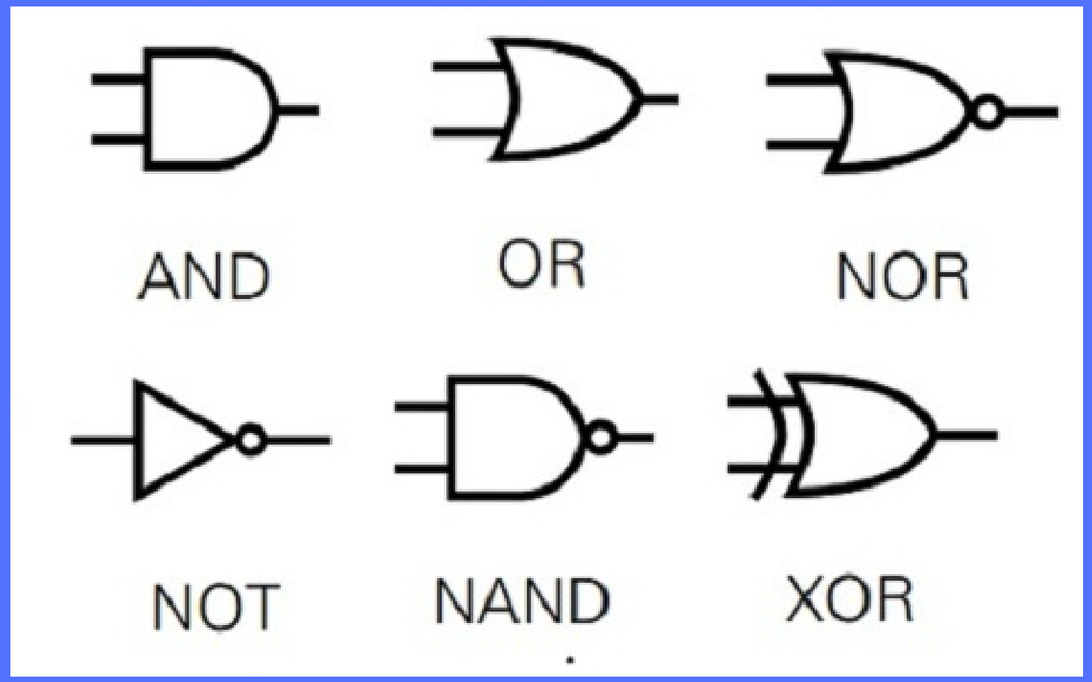


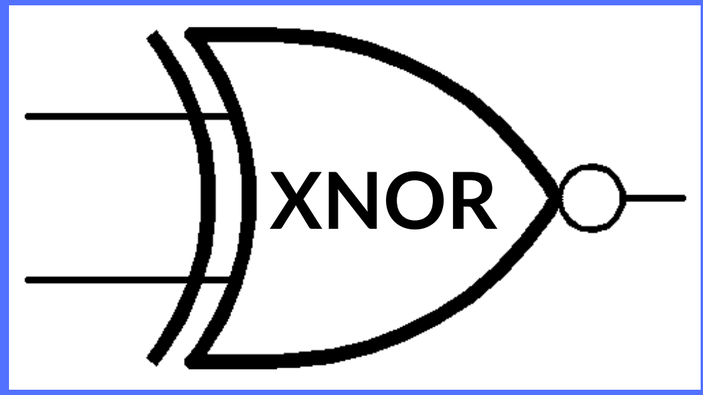
LOGIC GATES

IA LOGIC GATE IS A TOOL THAT DEVELOPERS USE TO SHOW LOGIC.

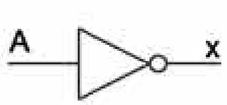
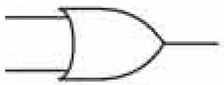
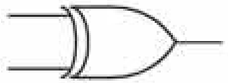
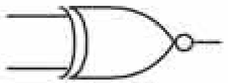
THESE GATES ACT AS THE FOUNDATION FOR DIGITAL CIRCUITS. THEY PERFORM BASIC LOGICAL FUNCTIONS THAT ARE FUNDAMENTAL TO DIGITAL CIRCUITS.

THERE ARE SEVEN BASIC LOGIC GATES: AND, OR, XOR, NOT, NAND, NOR, AND XNOR.





LOGIC GATES

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$
Symbol							
Truth Table	A | X 0 | **1** 1 | **0**	B A | X 0 0 | **0** 0 1 | **0** 1 0 | **0** 1 1 | **1**	B A | X 0 0 | **1** 0 1 | **1** 1 0 | **1** 1 1 | **0**	B A | X 0 0 | **0** 0 1 | **1** 1 0 | **1** 1 1 | **1**	B A | X 0 0 | **1** 0 1 | **0** 1 0 | **0** 1 1 | **0**	B A | X 0 0 | **0** 0 1 | **1** 1 0 | **1** 1 1 | **0**	B A | X 0 0 | **1** 0 1 | **0** 1 0 | **0** 1 1 | **1**

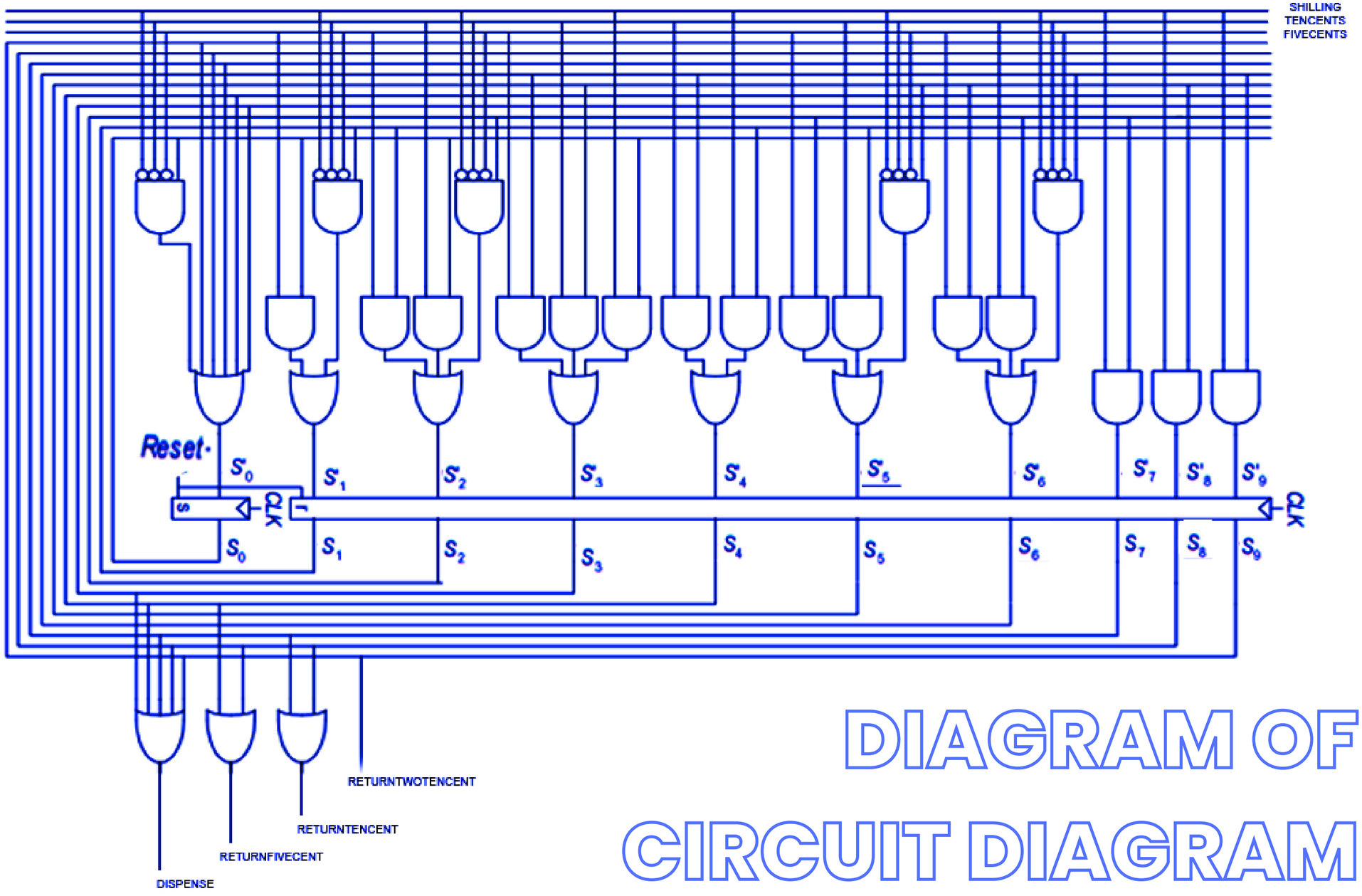


DIAGRAM OF CIRCUIT DIAGRAM

THANK YOU FOR EVERYTHING!