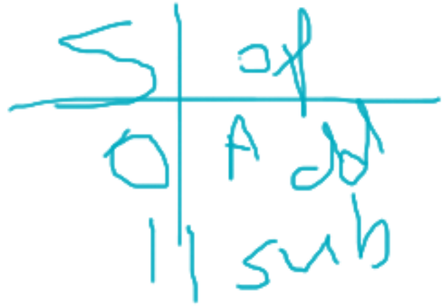
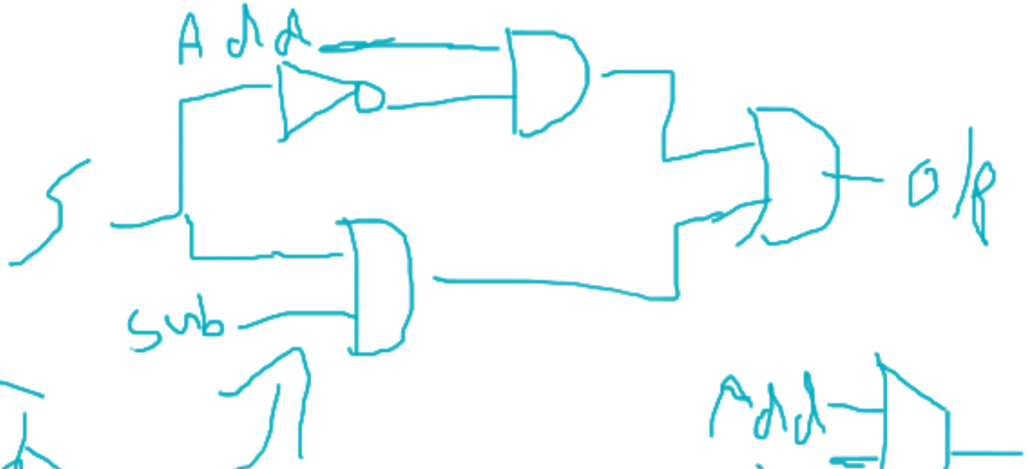
Mux





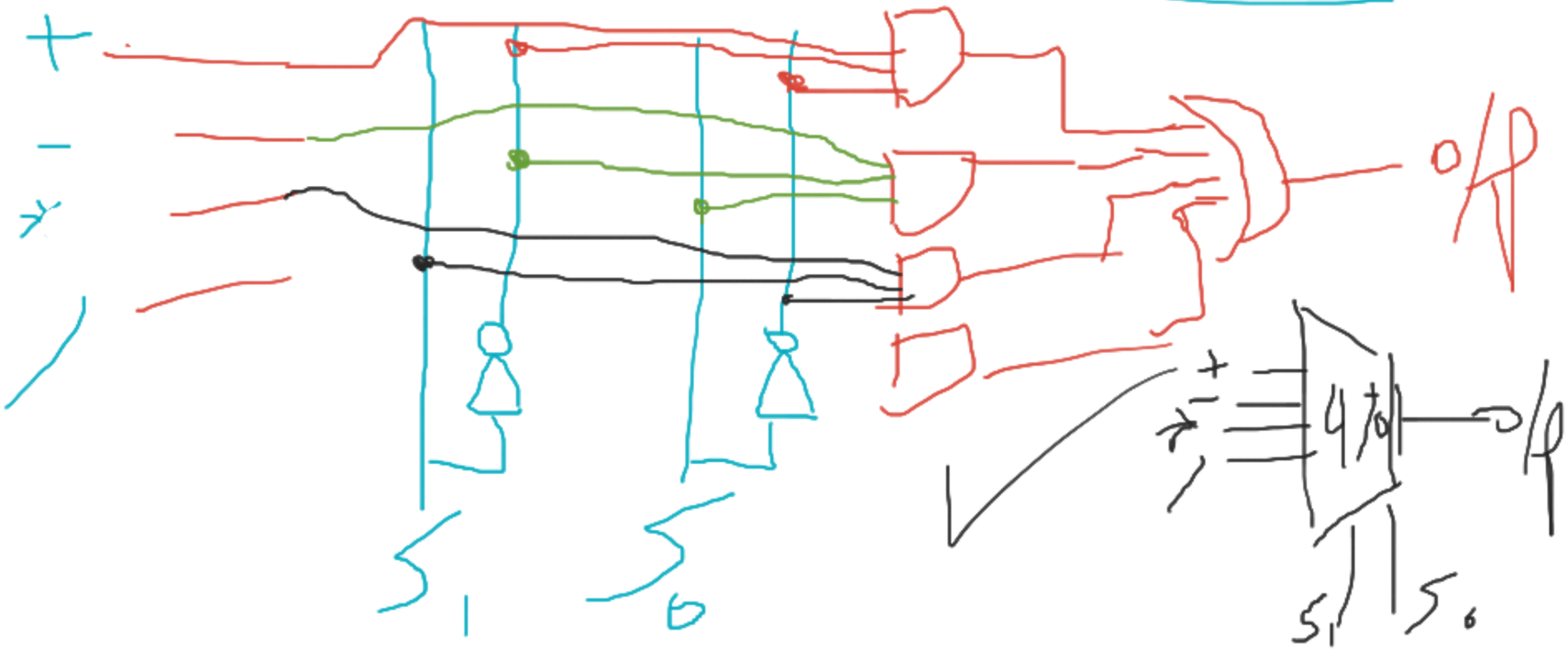
2 to 1

4 to 1

8 to 1

$\bar{S}.Add + S.Sub$

2⁶ → selectors



BCD

Table

Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
	1010	0101	0000	0001
Unused	1011	0110	0001	0010
bit	1100	0111	0010	0011
combi-	1101	1000	1101	1100
nations	1110	1001	1110	1101
	1111	1010	1111	1110

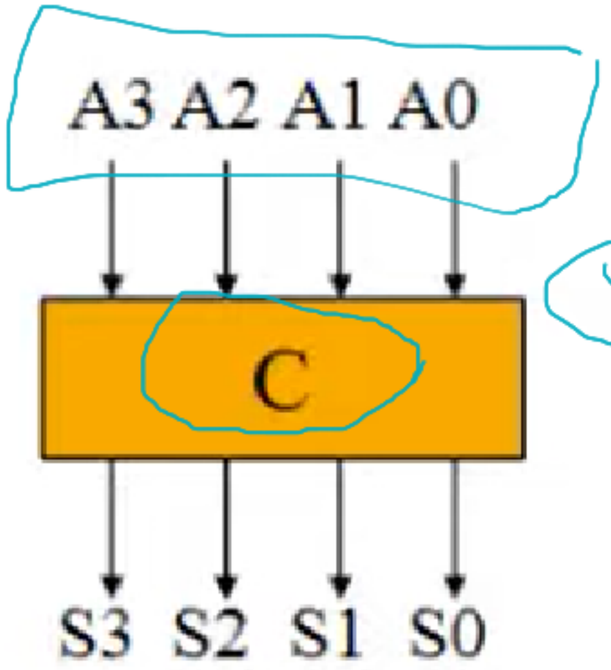
	Binary				Binary Coded Decimal (BCD)					
HEX	b3	b2	b1	b0	p4	p3	p2	p1	p0	BCD
0	0	0	0	0	0	0	0	0	0	00
1	0	0	0	1	0	0	0	0	1	01
2	0	0	1	0	0	0	0	1	0	02
3	0	0	1	1	0	0	0	1	1	03
4	0	1	0	0	0	0	1	0	0	04
5	0	1	0	1	0	0	1	0	1	05
6	0	1	1	0	0	0	1	1	0	06
7	0	1	1	1	0	0	1	1	1	07
8	1	0	0	0	0	1	0	0	0	08
9	1	0	0	1	0	1	0	0	1	09
A	1	0	1	0	1	0	0	0	0	10
B	1	0	1	1	1	0	0	0	1	11
C	1	1	0	0	1	0	0	1	0	12
D	1	1	0	1	1	0	0	1	1	13
E	1	1	1	0	1	0	1	0	0	14
F	1	1	1	1	1	0	1	0	1	15

Shift and Add-3 Algorithm

1. Shift the binary number left one bit.
2. If 8 shifts have taken place, the BCD number is in the *Hundreds*, *Tens*, and *Units* column.
3. If the binary value in any of the BCD columns is 5 or greater, add 3 to that value in that BCD column.
4. Go to 1.

unsigned

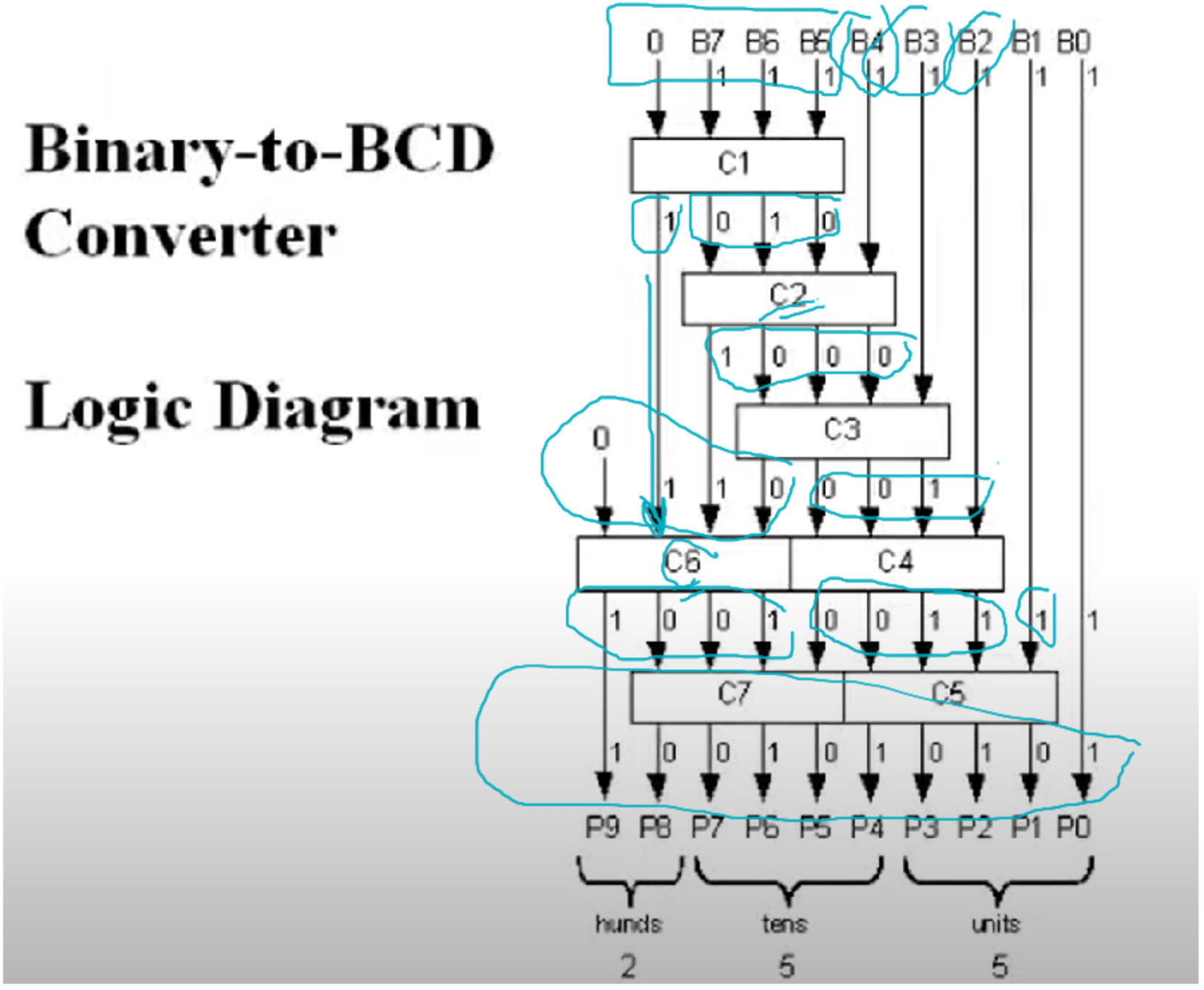
Operation	Hundreds	Tens	Units	Binary	
HEX				F	F
Start				1 1 1 1	1 1 1 1
Shift 1			1	1 1 1 1	1 1 1
Shift 2			1 1	1 1 1 1	1 1
Shift 3			1 1 1	1 1 1 1	1
Add 3		1	1 0 1 0	1 1 1 1	1
Shift 4		1	0 1 0 1	1 1 1 1	
Add 3		1	1 0 0 0	1 1 1 1	
Shift 5		1 1	0 0 0 1	1 1 1	
Shift 6		1 1 0	0 0 1 1	1 1	
Add 3		1 0 0 1	0 0 1 1	1 1	
Shift 7	1	0 0 1 0	0 1 1 1	1	
Add 3	1	0 0 1 0	1 0 1 0	1	
Shift 8	1 0	0 1 0 1	0 1 0 1		
BCD	2	5	5		

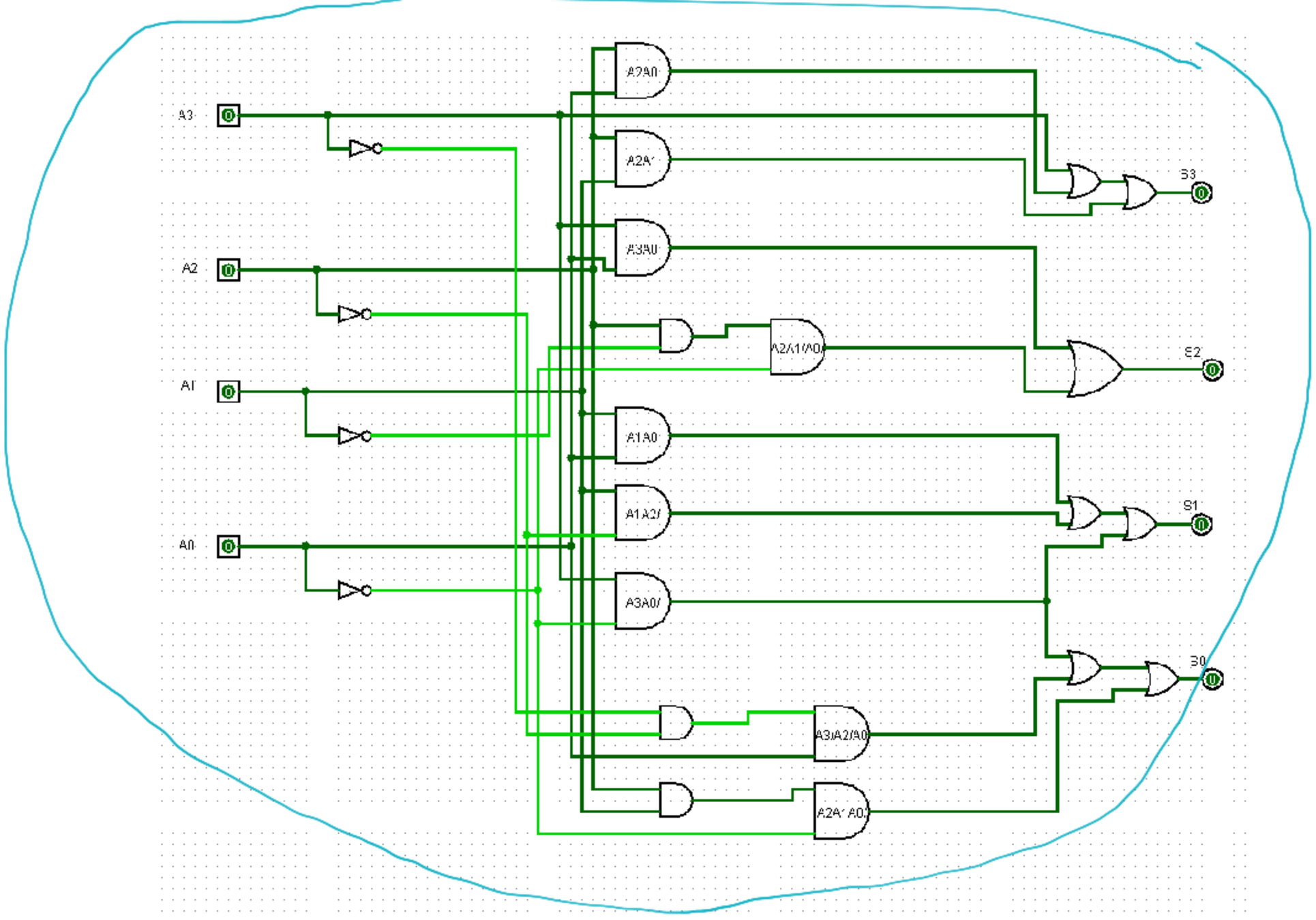


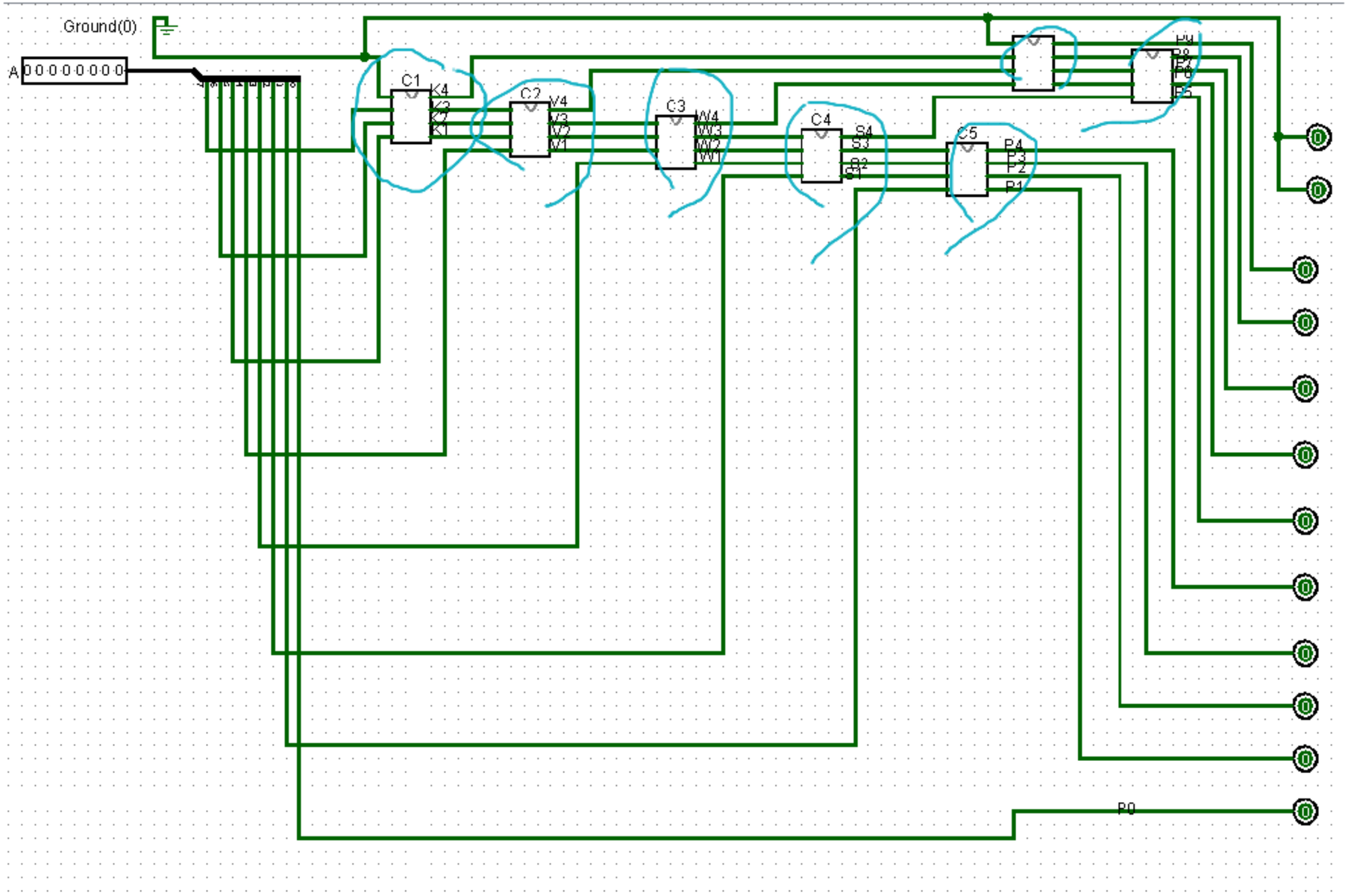
A3	A2	A1	A0	S3	S2	S1	S0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Binary-to-BCD Converter

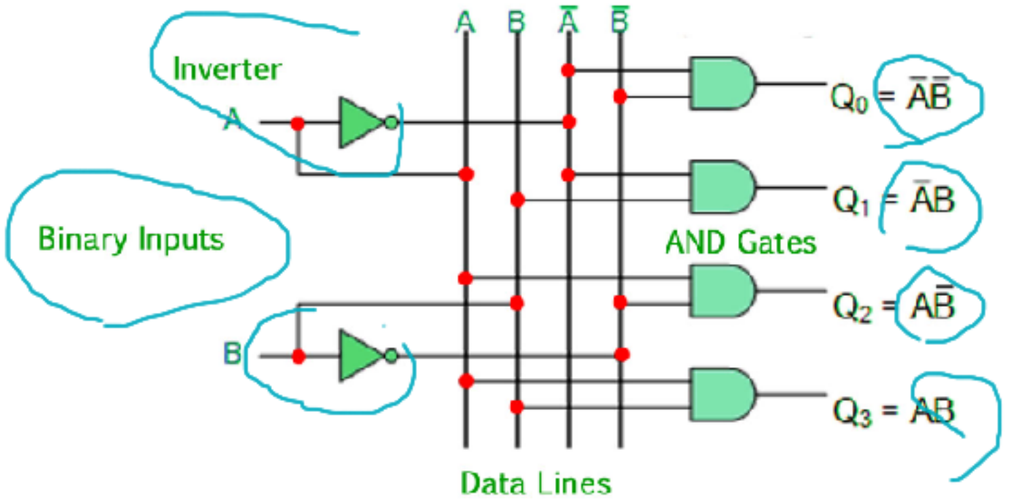
Logic Diagram




A3
A2
A1
A0
A2A0
A2A'
A3A0
A2A1A0
A1A0
A1A2'
A3A0'
A3A2A0
A2A' A0
S3
S2
S1
S0

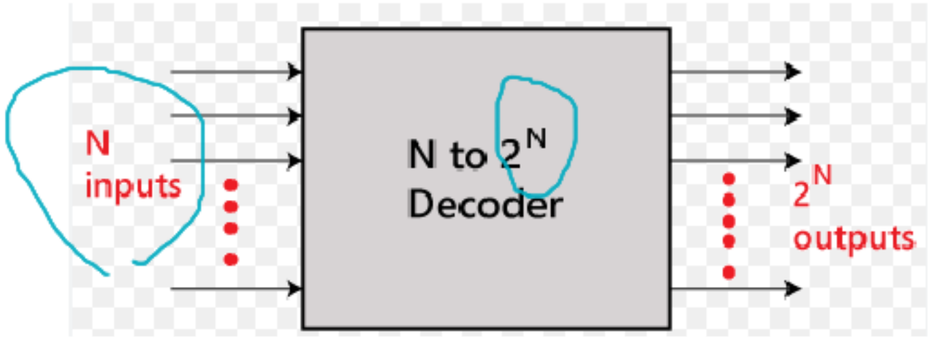

Ground(0)
A
C1
C2
C3
C4
C5
P0

Decoder



Decoded Output

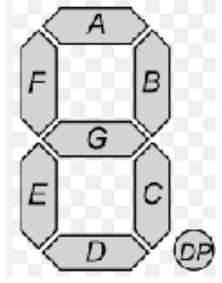
3 — 8



A	B	i/p
0	0	→ AB
0	1	→ ĀB
1	0	→ AB̄
1	1	→ AB

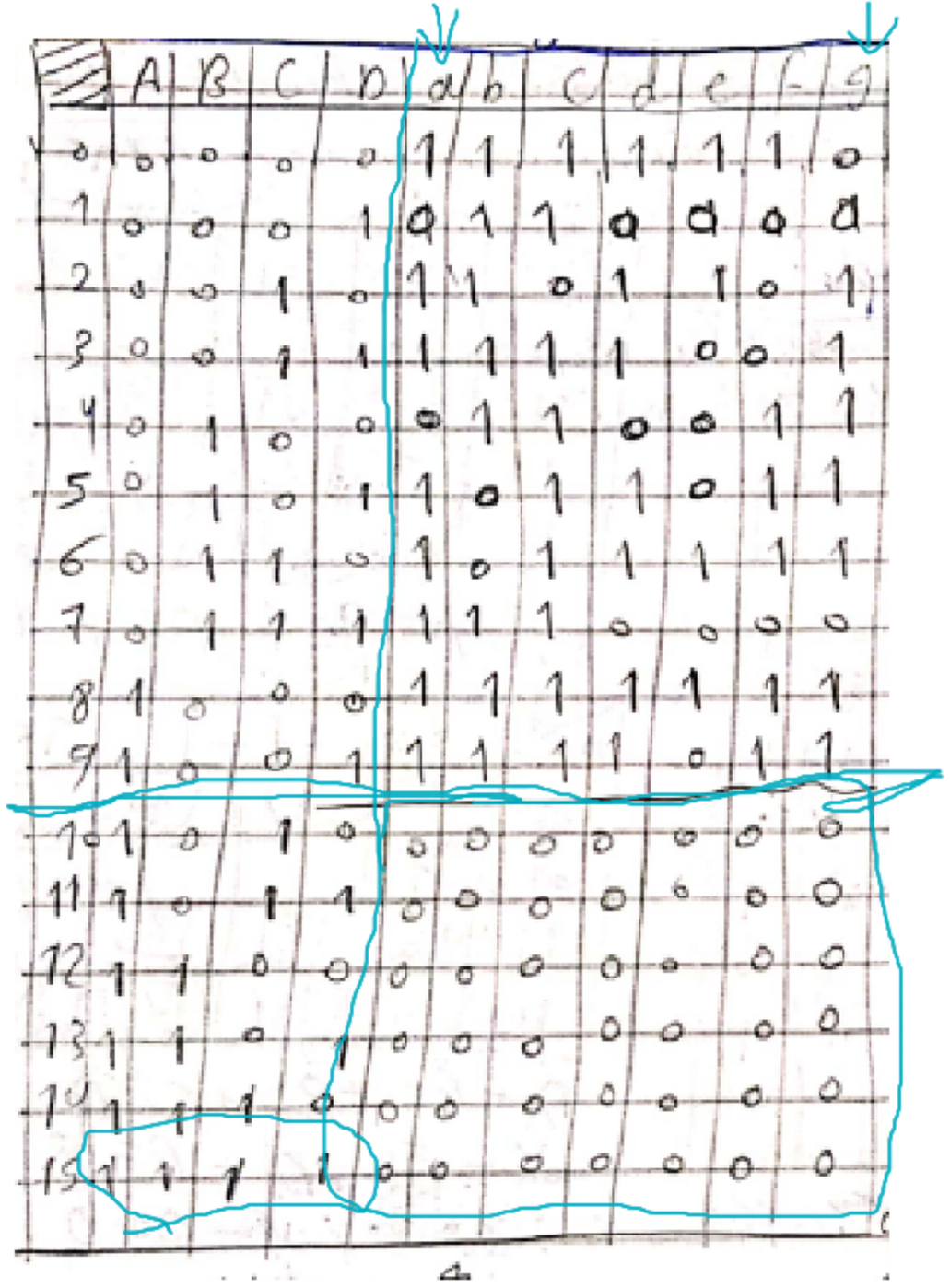
Search for Encoder

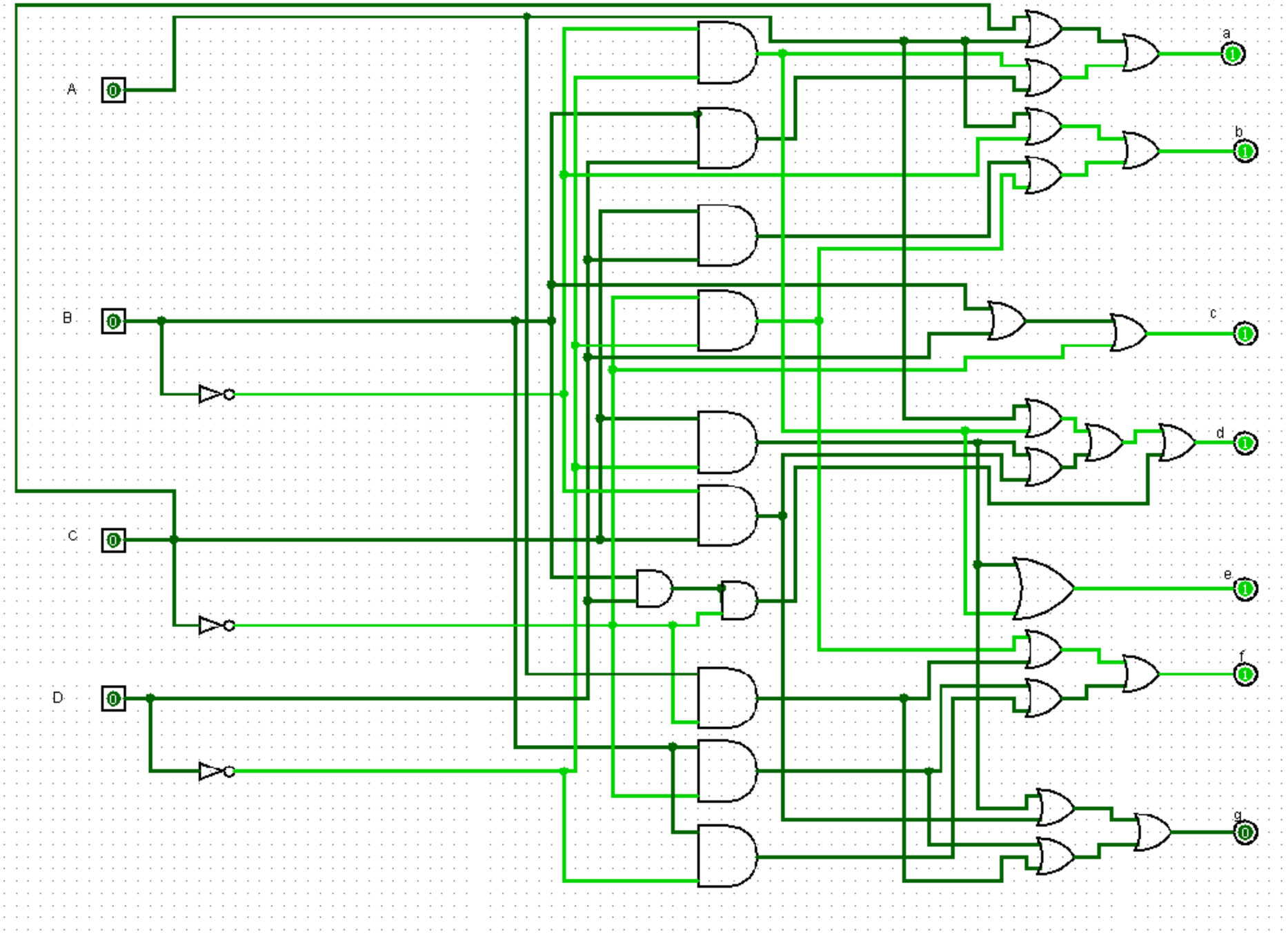
Decimal Digit	Input lines				Output lines							Display pattern
	A	B	C	D	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	0	1	0	1	1	0	1	1	0	1	2
3	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	1	0	1	1	1	1	1	6
7	0	1	1	1	1	1	1	0	0	0	0	7
8	1	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	1	1	1	1	0	1	1	9



$$a = \bar{A}C + \bar{A}B\bar{D} + \bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}$$

$$b = \bar{A}\bar{B} + \bar{A}\bar{C}\bar{D} + \bar{A}CD + A\bar{B}\bar{C}$$




A
B
C
D
a
b
c
d
e
f
g

Main Interface