Exercise 1

Cypher

Exercise 2.2.16

▶ Convert 252987_{10} to hexadecimal. Treat all numbers as unsigned. Use an accuracy of four fractional digits and don't round up.

$$\frac{252987}{16} = 158111 \quad R\,11 \quad B$$

$$\frac{158111}{16} = 988 \quad R\,3 \quad 3$$

$$\frac{988}{16} = 61 \quad R\,12 \quad C$$

$$\frac{61}{16} = 3 \quad R\,13 \quad D$$

$$\frac{3}{16} = 0 \quad R\,3 \quad 3$$

$= 3DC3B_{16}$

$\therefore 252987_{10} = 3DC3B_{16}$

Exercise 2.2.22

- Convert $AB.D_{16}$ to octal. Treat all numbers as unsigned.

Step 1 :

A	=	10
B	=	11
D	=	13

Step 2 :

		2^3	2^2	2^1	2^0
A	=	1	0	1	0
B	=	1	0	1	1
D	=	1	1	0	1

Step 3 :

0 1 0	1 0 | 1	0 1 1 |	1 1 0	1 0 0
2	4 + 1	2 + 1	4 + 2	4
2	5	3 .	6	4

Hence $AB.D_{16} = 253.64_8$

Exercise 2.4.11

▸ What is the decimal value of the 8-bit, two's complement code $0111\ 1110_2$?

Step 1:

$0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0_2$

$\downarrow \quad 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 = 126_{10}$

indicate (+ no)

Hence, $01111110_2 = 126_{10}$

Exercise 2.4.21

▸ Compute $-99_{10} + -11_{10}$ using 8-bit two's complement addition. You will need to first convert each decimal number into its 8-bit two's complement code and then perform binary addition (i.e., (9910)+(1110)). Provide the 8-bit result and indicate whether two's complement overflow occurred. Check your work by converting the 8-bit result back to decimal.

Step 1:

$$\begin{array}{lcccccccc}
 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
99_{10} = & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
\text{then inverting} & & & & & & & & \\
 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
\text{then} \quad + & & & & & & & & 1 \\
 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1
\end{array}$$

Hence, $-99_{10} = 10011101_2$

Step 2:

$$\begin{array}{lcccccccc}
11_{10} = & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
\text{then inverting} & & & & & & & & \\
 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
\text{then} \quad + & & & & & & & & 1 \\
 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1
\end{array}$$

Hence, $-11_{10} = 11110101_2$

Step 3:

$$\begin{array}{ccccccccc}
 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
+ & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{array}$$

Then invert

$$\begin{array}{ccccccccc}
 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
+ & & & & & & & & 1 \\
 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0
\end{array}$$

$$\begin{array}{lcccccccc}
 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
\text{Step 4:} & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0
\end{array}$$

$= 110_{10}$

Hence

$-99_{10} + -11_{10} = 110_{10}$

Additional Exercises

- Compute the following numbers using 8-bit two's complement addition and give the result as a decimal number:

a) ▸ 1001 0001 + 0011 0010
b) ▸ 0101 1101 + 0111 0010
c) ▸ 1010 1001 + 1001 1110
d) ▸ 1010 1010 + 0010 1111

a)
  1 0 0 1 0 0 0 1
+ 0 0 1 1 0 0 1 0
-ve value ← 1 1 0 0 0 0 1 1
− 1
  1 1 0 0 0 0 1 0

Then, invert
  0 0 1 1 1 1 0 1

$2^7 \; 2^6 \; 2^5 \; 2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0$

$= 2^5 + 2^4 + 2^3 + 2^2 + 2^0$

$= -61$

b)
  0 1 0 1 1 1 0 1
+ 0 1 1 1 0 0 1 0
-ve value ← 1 1 0 0 1 1 1 1
− 1
  1 1 0 0 1 1 1 0

Invert,
  0 0 1 1 0 0 0 1

$2^7 \; 2^6 \; 2^5 \; 2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0$

$= 2^5 + 2^4 + 2^0$

$= -49$

c)
  1 0 1 0 1 0 0 1   −87
+ 1 0 0 1 1 1 1 0   −98
1 0 1 0 0 0 1 1 1   185

= Overflow, invalid answer.

d)
  1 0 1 0 1 0 1 0
+ 0 0 1 0 1 1 1 1
-ve value ← 1 1 0 1 1 0 0 1
− 1
  1 1 0 1 1 0 0 0

  0 0 1 0 0 1 1 1

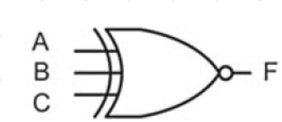
$2^7 \; 2^6 \; 2^5 \; 2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0$

$= 2^5 + 2^2 + 2^1 + 2^0$

$= -39$

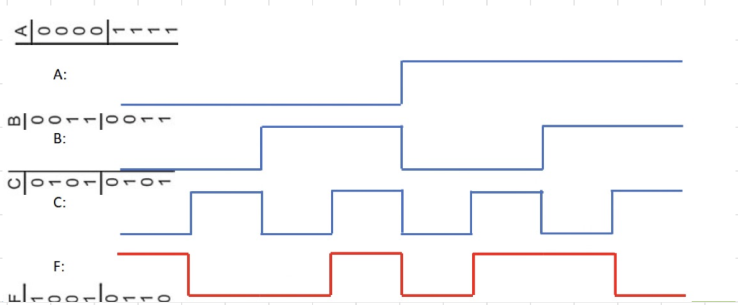
Exercise 3.1.9

▸ Give the logic waveform for a 3-input XNOR gate with the input variables A, B, and C and output F.



A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

C \ AB	00	01	11	10
0	1	0	1	0
1	0	1	0	1



Exercise 3.2.16

- Using the data sheet excerpt from Fig. 3.20, give the maximum fall time (tf) for the 74HC04 inverter when powered with VCC = 2 V.

Switching Characteristics

over operating free-air temperature range, C_L = 50 pF (unless otherwise noted) (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V_{CC}	T_A = 25°C			SN54HC04		SN74HC04		UNIT
				MIN	TYP	MAX	MIN	MAX	MIN	MAX	
t_{pd}	A	Y	2 V		45	95		125		120	ns
			4.5 V		9	19		29		24	
			6 V		8	16		25		20	
t_f		Y	2 V		38	75		110		95	ns
			4.5 V		8	15		22		19	
			6 V		6	13		19		16	

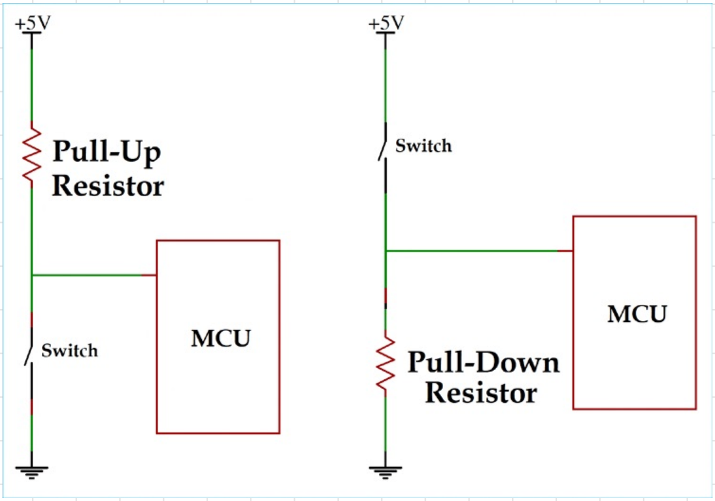
- Fall time (tf) : time for High → Low

  Max tf = 95 ns

Additional Exercises

- Why are pull-down and pull-up resistors used in circuits? Explain your answer using the example of connecting a button to the input of a microcontroller.

pull-down resistor — to avoid noise at input terminal
pull-up resistor — to make input at terminal always HIGH and 0 if there is input on button.



Exercise 3.4.2

▸ For the pull-down driver configuration shown in Fig. 3.39, calculate the value of the pull-down resistor (R) in order to ensure that the output current does not exceed 20 mA.

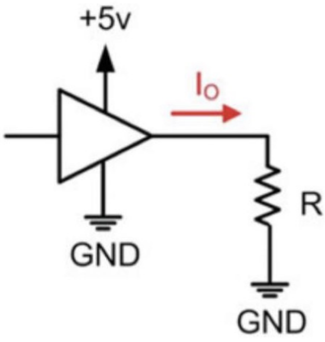


Fig. 3.39

$$V = IR$$

$$+5V = (20\,mA)\,R$$

$$R > 250\,\Omega$$

Exercise 3.4-5

▸ For the LED driver configuration shown in Fig. 3.42 where an output of LOW on the driver will turn on the LED, calculate the value of the resistor (R) in order to set the LED forward current to 5 mA. The LED has a forward voltage of 1.9 V.

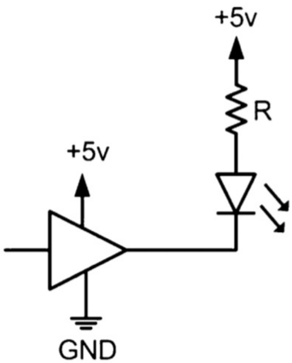


Fig. 3.42

$$V = IR$$

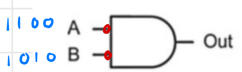
$$5V - 1.9V = (5mA)\,R$$

$$R = 620\ \Omega$$

Exercise 4.1.20

- Use proof by exhaustion to prove that an AND gate with its inputs inverted is equivalent to an OR gate with its outputs inverted.

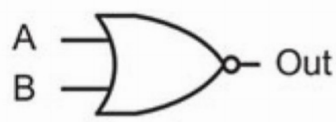
Symbol



Truth table

A	B	A`	B`	Out
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Symbol



Truth Table

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Therefore, it is proven.

Exercise 4.2.8

▸ For the logic diagram given in Fig. 4.29, give the truth table for the output F.

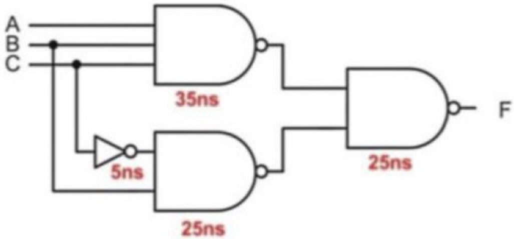


Fig. 4.29
Combinational Logic Analysis 3

F = ((A·B·C)'·(B·C')')'

F = ((A·B·C)'·(B·C')')'

A	B	C	C'	B·C'	(B·C')'	(A·B·C)'	(A·B·C)'·(B·C')'	F
0	0	0	1	0	1	1	1	0
0	0	1	0	0	1	1	1	0
0	1	0	1	1	0	1	0	1
0	1	1	0	0	1	1	1	0
1	0	0	1	0	1	1	1	0
1	0	1	0	0	1	1	1	0
1	1	0	1	1	0	1	0	1
1	1	1	0	0	1	0	0	1

Exercise 4.3.26

▶ For the 3-input minterm list in Fig. 4.34, give the canonical sum of products (SOP) logic diagram.

$$F = \Sigma_{A,B,C}(2,4,6)$$

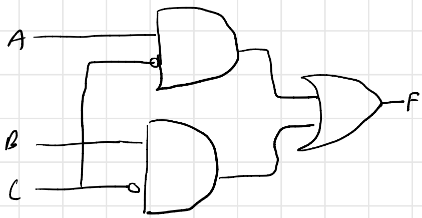
Fig. 4.34
Combinational Logic Synthesis 5

$$F = \Sigma_{A,B,C}(2,4,6)$$

L	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	**0**	**1**	**0**	1
3	0	1	1	0
4	**1**	**0**	**0**	1
5	1	0	1	0
6	**1**	**1**	**0**	1
7	1	1	1	0

C \ AB	00	01	11	10
0	0	1	1	1
1	0	0	0	0

$F = B.C' + AC'$



Exercise 4.3.43

▶ For the 4-input minterm list in Fig. 4.37, give the canonical sum of products (SOP) logic expression.

$$F = \Sigma_{A,B,C,D}(4,5,7,12,13,15)$$

Fig. 4.37
Combinational Logic Synthesis 8

$$F = \Sigma_{A,B,C,D}(4,5,7,12,13,15)$$

A	B	C	D	F(ABCD)	
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	1	A' · B · C' · D' = 4
0	1	0	1	1	A' · B · C' · D = 5
0	1	1	0	0	
0	1	1	1	1	A' · B · C · D = 7
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	1	A · B · C' · D' = 12
1	1	0	1	1	A · B · C' · D = 13
1	1	1	0	0	
1	1	1	1	1	A · B · C · D = 15

Optional

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

K-map

$F = (B \cdot C') + (BD)$

Exercise 4-4-33

- For the 4-input truth table and K-map in Fig. 4.54, give the minimal product of sums (POS) logic expression by exploiting "don't cares."

A	B	C	D	F
0	0	0	0	1
0	0	0	1	X
0	0	1	0	X
0	0	1	1	X
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	X
1	0	0	1	X
1	0	1	0	0
1	0	1	1	X
1	1	0	0	0
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

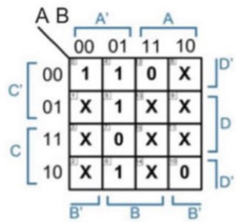


Fig. 4.54
Logic Minimization 16

- K-map

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

A

$C+D$

$F = (A') \cdot (C' + D')$

Exercise 4-5-8

▸ For the 4-input truth table and K-map in Fig. 4.52, give the sum term that helps eliminate static-0 timing hazards in this circuit.

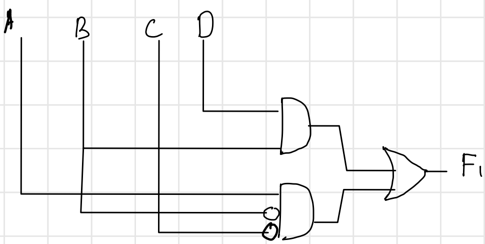
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

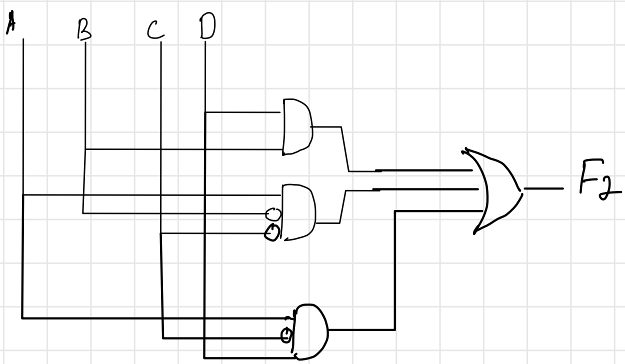
CD \ AB	00	01	11	10
00	0	0	0	1
01	0	1	1	1
11	0	1	1	0
10	0	0	0	0

Fig. 4.52
Logic Minimization 14

$F_1 = (B \cdot D) + (A \cdot B' + C')$

$F_2 = (B \cdot D) + (A . B' + C') + (A \cdot C' + D)$





Additional Exercises

- Simplify the following functions using Boolean algebra axioms und theorems:

a) - $F = A' \cdot C + B' \cdot C + D' \cdot B \cdot C + A \cdot D \cdot C$

b) - $Z = A' \cdot B \cdot C + B \cdot C + A \cdot B \cdot C + A \cdot B' \cdot C$

a)
$$F = A' \cdot C + B' \cdot C + D' \cdot B \cdot C + A \cdot D \cdot C$$
$$F = C \cdot (A' + A \cdot D) + C \cdot (B' + B \cdot D')$$
$$F = C \cdot (A' + A \cdot D) + C \cdot (B' + B \cdot D')$$
$$F = C \cdot (A' + D) + C \cdot (B' + D')$$
$$F = C \cdot A' + C \cdot D + C \cdot B' + C \cdot D'$$
$$F = C \cdot A' + C \cdot B' + C \cdot (D + D')$$
$$F = C \cdot A' + C \cdot B' + C \cdot (D + D')$$
$$F = C \cdot A' + C \cdot B' + C$$
$$F = C \cdot (A' + B' + 1)$$
$$F = C$$

b)
$$Z = A' \cdot B \cdot C + B \cdot C + A \cdot B \cdot C + A \cdot B' \cdot C$$
$$Z = B \cdot C \cdot (A' + 1) + A \cdot C \cdot (B + B')$$
$$Z = B \cdot C \cdot (1) + A \cdot C \cdot (1)$$
$$Z = B \cdot C + A \cdot C$$
$$Z = C \cdot (B + A)$$