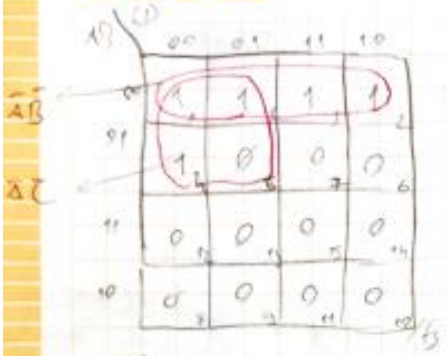
fonksiyonu gerçekleştirmek için diğer soruda olduğu gibi

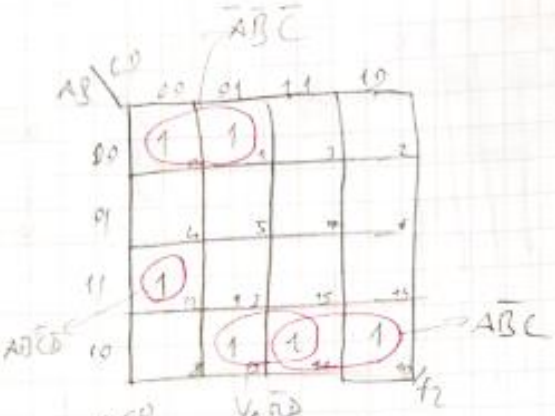
a) $f = x - 5$ fonksiyonunu yalnızca NAND kapıları kullanarak gerçekleştiriniz. x burada 4 bitlik bir değişken

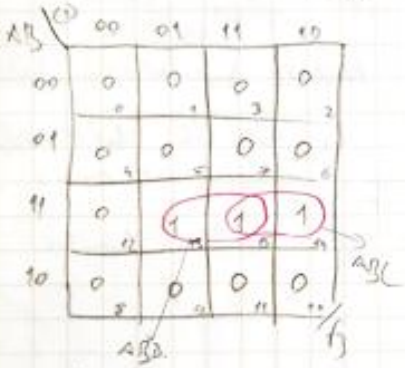
negatif değerler için bir lamba, fonksiyonun mutlak değerleri yeşil lamba ile temsil edilecek.

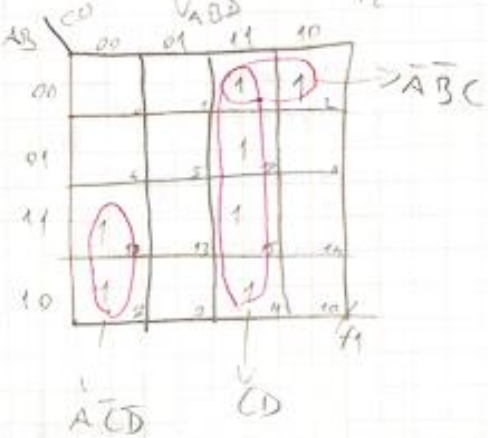
Çözüm:

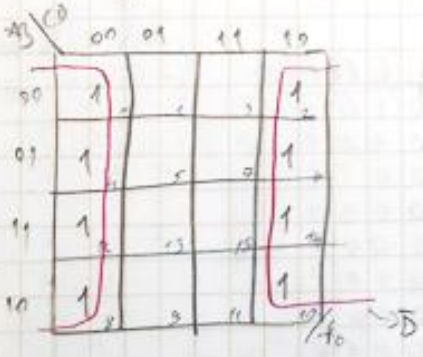
	A	B	C	D	f=x-5	f_3	f_2	f_1	f_0
0	0	0	0	0	-5	1	0	1	1
1	0	0	0	1	-4	1	0	0	0
2	0	0	1	0	-3	1	0	0	1
3	0	0	1	1	-2	1	0	1	0
4	0	1	0	0	-1	1	0	0	1
5	0	1	0	1	0	0	0	0	0
6	0	1	1	0	+1	0	0	0	1
7	0	1	1	1	+2	0	0	1	0
8	1	0	0	0	+3	0	0	1	1
9	1	0	0	1	+4	0	0	1	0
10	1	0	1	0	+5	0	0	1	1
11	1	0	1	1	+6	0	0	1	0
12	1	1	0	0	+7	0	0	1	1
13	1	1	0	1	+8	0	1	0	0
14	1	1	1	0	+9	0	1	0	1
15	1	1	1	1	+10	0	1	0	0











$\bar{A}\bar{C}\bar{D}$

$f_3 = \overline{\bar{A}\bar{B} + \bar{A}\bar{C}} = \overline{\bar{A}\bar{B}} \cdot \overline{\bar{A}\bar{C}}$

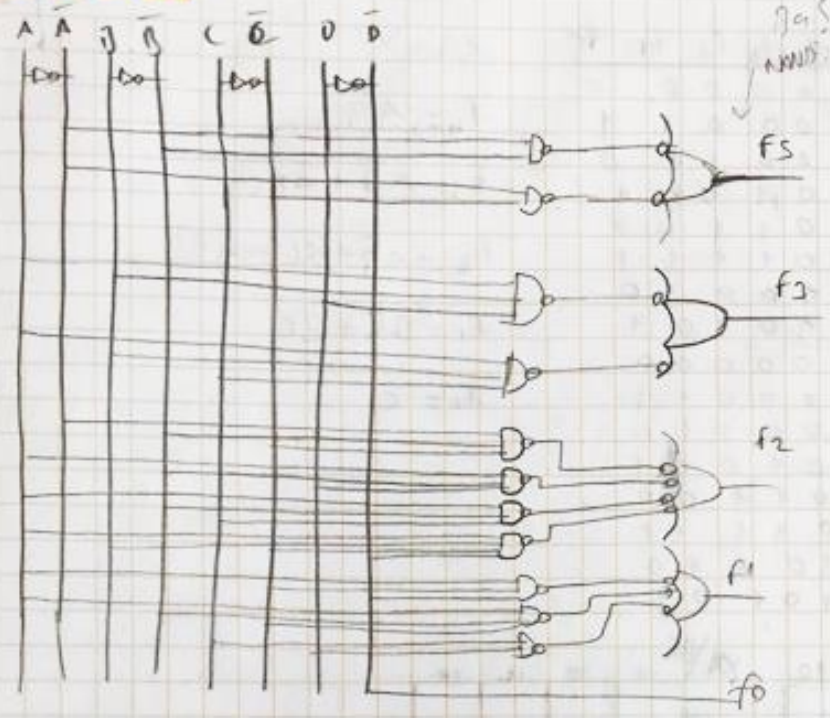
$f_3 = \overline{ABD + ABC} =$

$f_2 = \bar{A}\bar{B}C + A\bar{B}D + A\bar{B}C + AB\bar{C}\bar{D}$

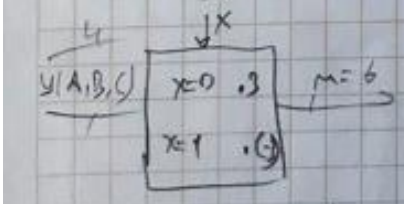
$f_1 = \bar{A}\bar{C}D + \bar{A}\bar{B}C + CD$

$f_0 = \overline{\bar{D}} = \bar{D}$

OSOW



P/ Bir lojik devrede y(A,B,C) 3 bitlik binary veri girişi x ise kontrol girişidir. x, 0 olduğu zaman y, +3 ile x, 1 olduğu zaman y, -3 ile çarpılmaktadır. Böyle bir devreyi yalnızca nand kapıları kullanarak gerçekleştiriniz. (Fonksiyonun işareti için; negatif değerler için lojik 1 pozitif değerler için lojik 0 kullanılmıştır.)



7 → 3 +21

-3 -21

OSOU

Dec Y	X	A	B	C	f	f_5	f_4	f_3	f_2	f_1	f_0
0	0	0	0	0	00	Ø	0	0	0	0	0
1	0	0	0	1	3	0	0	0	0	1	1
2	0	0	1	0	6	0	0	0	1	1	0
3	0	0	1	1	9	0	0	1	0	0	1
4	0	1	0	0	12	0	0	1	1	0	0
5	0	1	0	1	15	0	0	1	1	1	1
6	0	1	1	0	18	0	1	0	0	1	0
7	0	1	1	1	21	0	1	0	1	0	1
0	1	0	0	0	00	Ø	0	0	0	0	0
1	1	0	0	1	3	1	0	0	0	1	1
2	1	0	1	0	6	1	0	0	1	1	0
3	1	0	1	1	9	1	0	1	0	0	1
4	1	1	0	0	12	1	0	1	1	0	0
5	1	1	0	1	15	1	0	1	1	1	1
6	1	1	1	0	18	1	1	0	0	1	0
7	1	1	1	1	21	1	1	0	1	0	1

$f_5 = X$

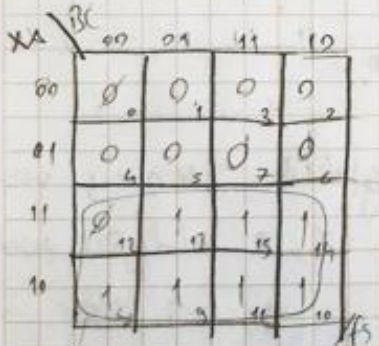
$f_4 = AB$

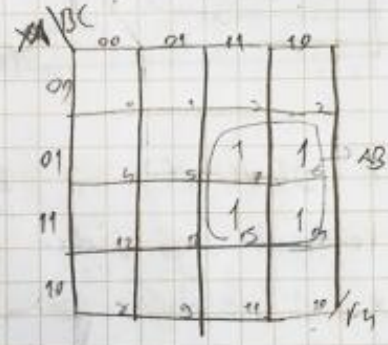
$f_3 = A\bar{B} + \bar{A}BC$

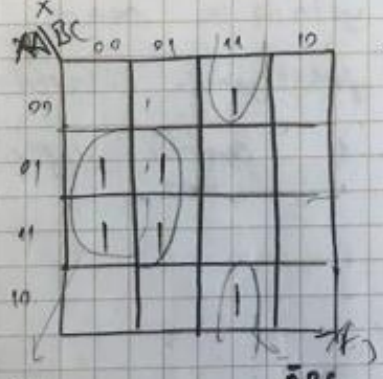
$f_2 = A\bar{B} + AC + \bar{A}B\bar{C}$

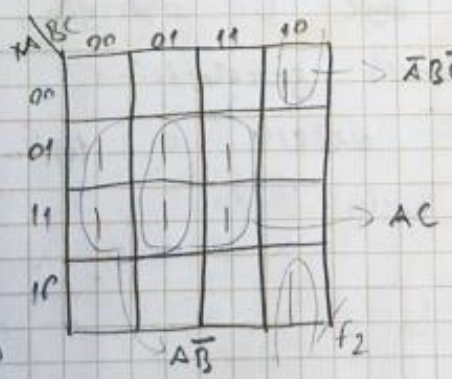
$f_1 = B\bar{C} + \bar{B}C$

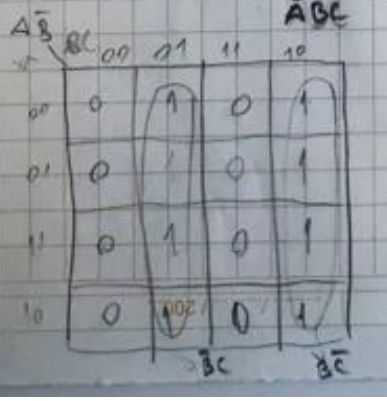
$f_0 = C$

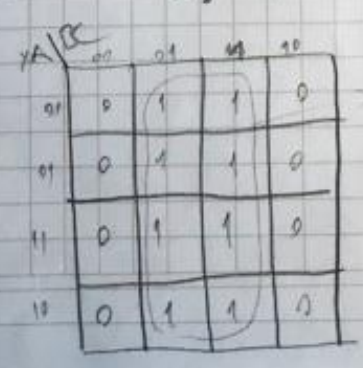












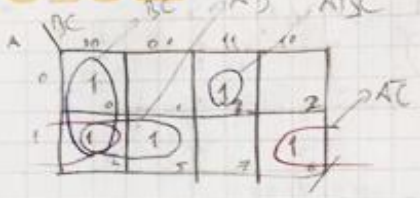
Multiple Output Minimization

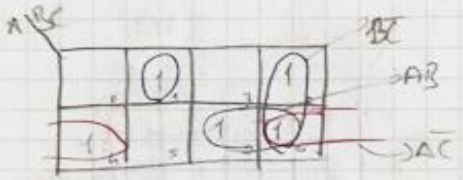
Ör na/ $f_1 = \sum m(0,3,4,5,6)$

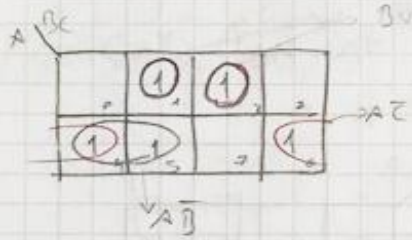
$f_2 = \sum m(1,2,4,6,7)$

$f_3 = \sum m(1,3,4,5,6)$

OSOW



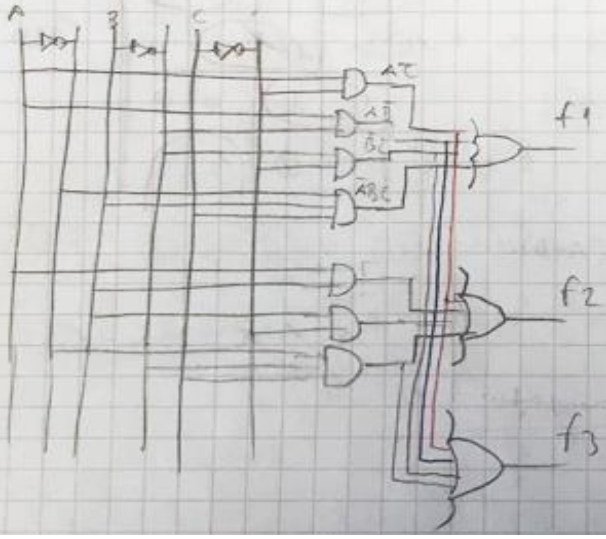




$f_1 = A\bar{C} + A\bar{B} + B\bar{C} + \bar{A}BC$

$f_2 = A\bar{C} + AB + B\bar{C} + \bar{A}\bar{B}C$

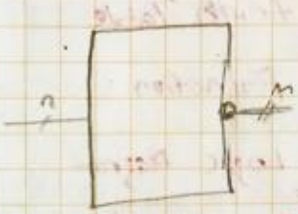
$f_3 = A\bar{C} + A\bar{B} + \bar{A}BC + \bar{A}\bar{B}C$



COMBINATIONAL LOGIC DESIGN

Gate Perspective (Approach)

Modular Perspective (Approach)



The Building Blocks

1) Data Manipulation

2) - Code conversion

3) - Data selection from various sources

4) - Data Routing from source to various Destinations

5) Data Error Detection

6) - Data busing from one part of the digital system to another

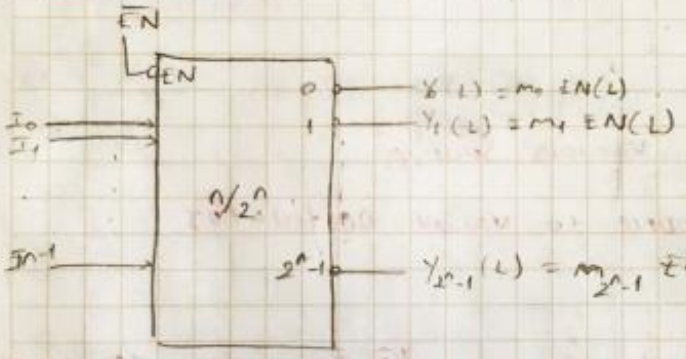
Arithmetic Circuit Type — Multipurpose Circuit

ADDER	DECODER - ENCODER
SUBTRACTOR	CODE CONVERTER
MULTIPLIER	MULTIPLEXER - DEMULTIPLEXER
DIVIDER	SHIFTER
COMPARATOR	ALU (Arithmetic and Logic unit)
PARITY GENERATOR	ROM (Read only memory)
ERROR DETECTOR	PLA (Programmable Logic Arrays)
	Array Logic

DESIGN PROCEDURE

1- Understand the Device
2- State the Algorithm
3- Construct a truth Table
4- Obtain output Function
5- Construct the Logic Diagram
6- Check (Test) the Results

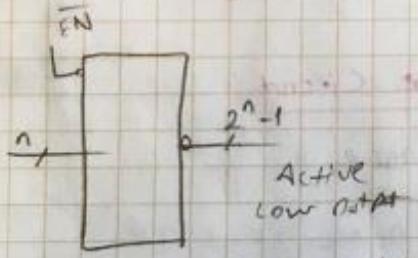
DECODER AND ENCODER

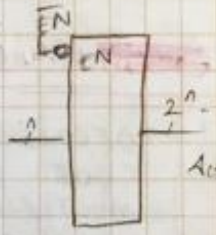


$m_i(L) = M_i(H)$

$M_i(L) = m_i(H)$

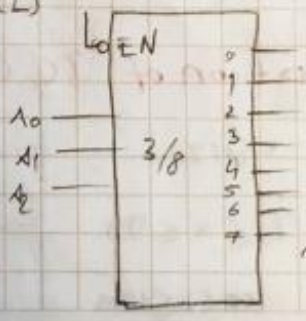
$y_i = m_i \overline{EN}(L)$





$\overline{EN}$	A_2	A_1	A_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

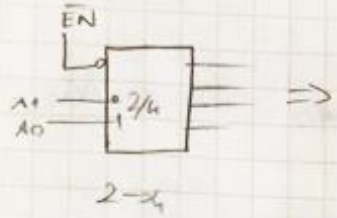
$y_i = m_i \overline{EN}(L)$



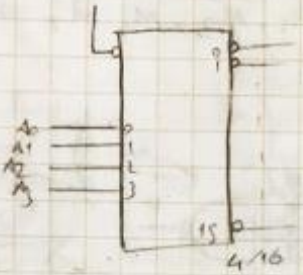
$\overline{A}_2 \overline{A}_1 \overline{A}_0 \overline{EN}(L)$

$\overline{A}_2 A_1 A_0 \overline{EN}(L)$

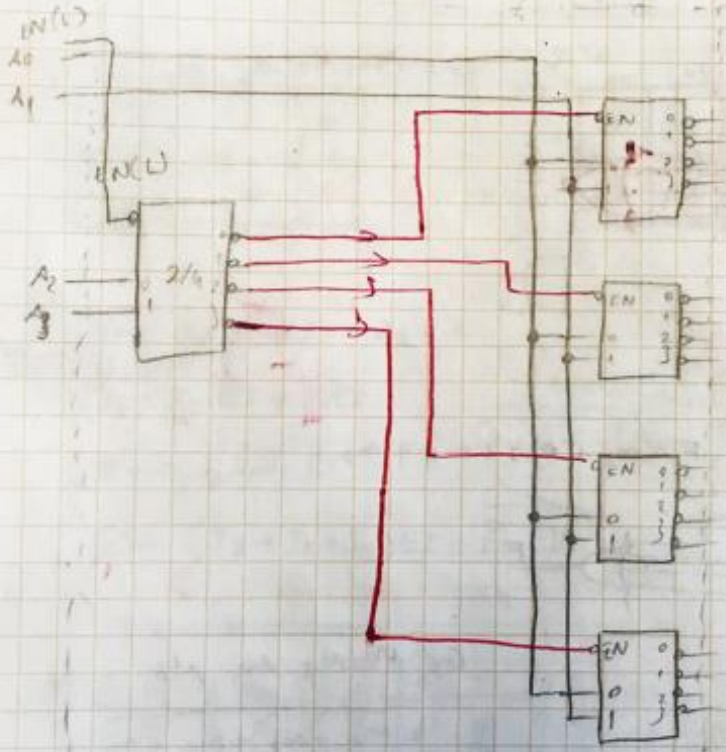
EXPANSION FORMAT



4→16 Decoder

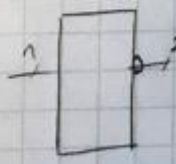


2/4 den 4/16 farke
elde etmek için



Combinational Logic Design with Decoder

Decoder



Logic Function

Canonical SOP form

Canonical POS form