Chapter 4

Karnaugh Map

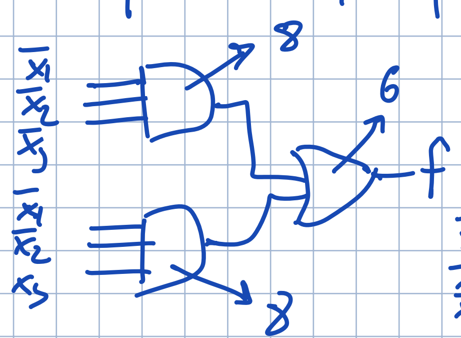
Task: Combinational circuit minimization

$$f(x_1,x_2,x_3) = \sum m(0,1) = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2 x_3 = \bar{x}_1\bar{x}_2(\bar{x}_3 + x_3) = \bar{x}_1 \cdot \bar{x}_2 = \overline{x_1 + x_2}$$

combining theorem

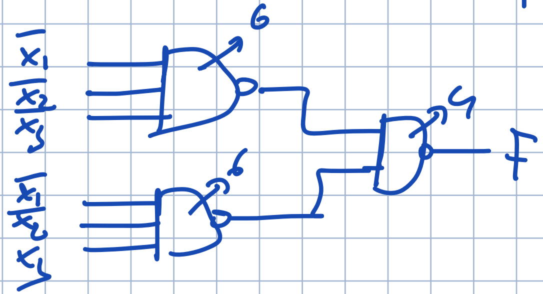
Assume that complements of inputs are available

Circuit 1:



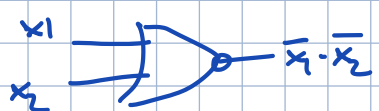
22 transistors in CMOS

3 gates, 8 gate inputs



16 transistors = 2x # gate inputs.

Circuit 2:



4 transistors, 2 gate inputs

3 variable K-MAP

x_1	x_2	x_3	
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7

$x_3 \backslash x_1x_2$	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

neighbors

f:

$x_3 \backslash x_1x_2$	00	01	11	10
0	1	0	0	0
1	1	0	0	0

x_1 (columns 11, 10); x_3 (row 1); $\bar{x}_1\bar{x}_2$ (column 00); x_2 (columns 01, 11)

$f = X_1 \cdot X_2$

- Each neighbor differs only in value of a single variable (Gray encoded)
- Neighbors combine
- Zero values = blank squares

f:

$x_3 \backslash x_1x_2$	00	01	11	10
0	1			
1	1			

	00	01	11	10
0	1	1		
1	1	1		

$\overline{x}_1$

$\cdot$

	00	01	11	10
0	1			1
1	1			1

$\overline{x}_2$

$=$

	00	01	11	10
0	1			
1	1			

$\overline{x}_1 \cdot \overline{x}_2$

Systematic Method for Simplification

literal: a variable or its complement

ex: $\overline{x}_1 \; x_2 x_3$ ← literal

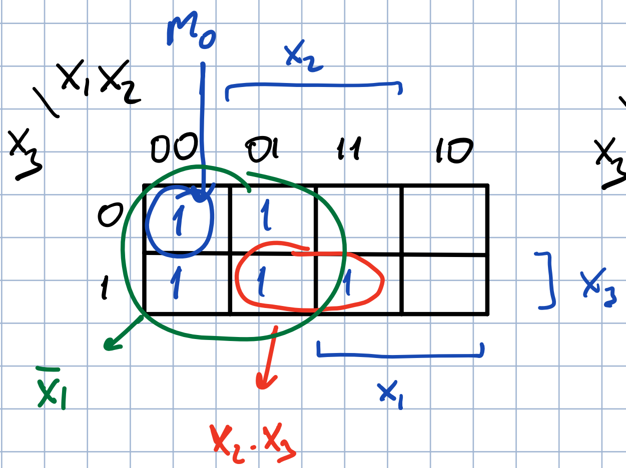
literal

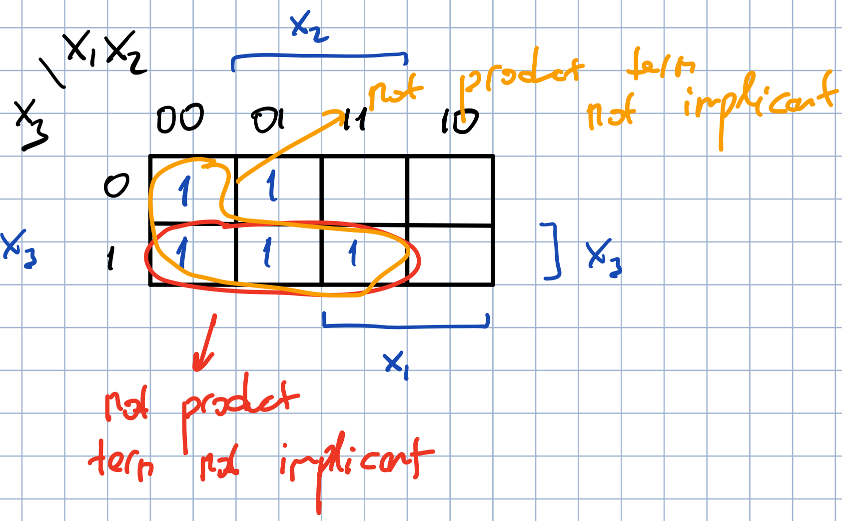
product term:

ex: $x_1\overline{x}_2$, $x_1x_2x_3$, x_3 , $\overline{x}_1$

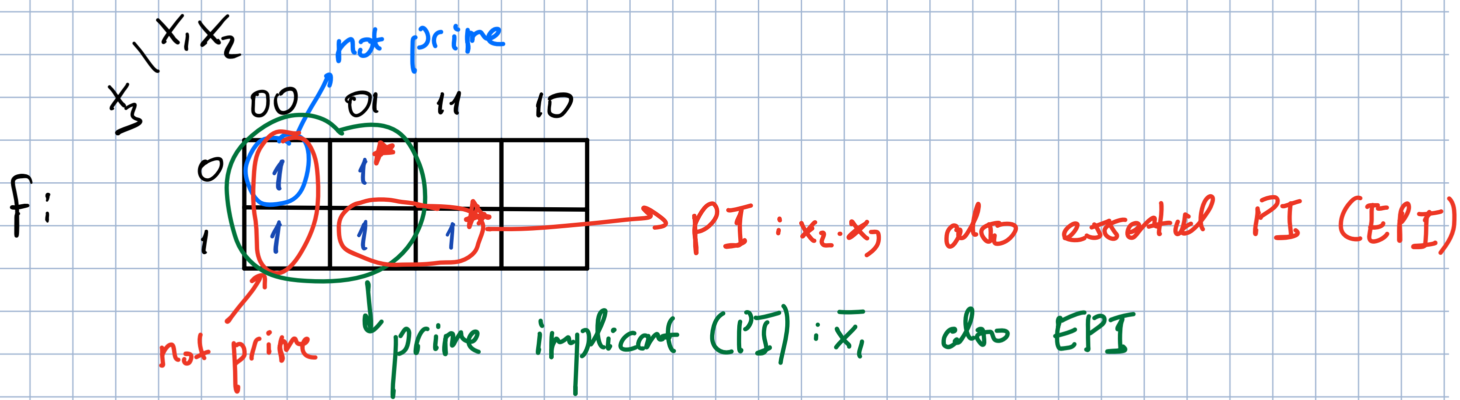
implicant: product term that covers a subset of 1s on KMAP but not any 0s

* depends on the logic fn.





Prime implicant: Implicant that cannot be combined into an implicant with fever literals



Cover: a collection of implicants that cover all 1s on KMAP

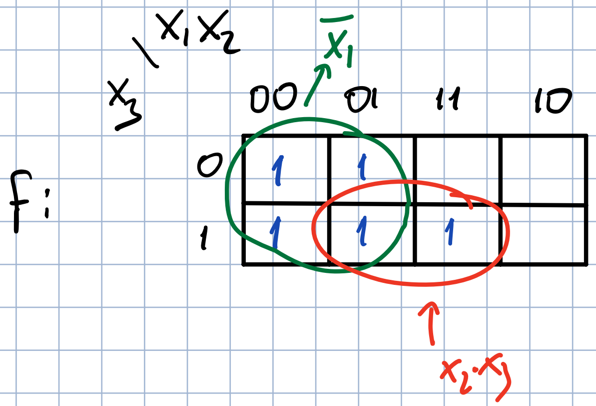
ex: $\{m_0, m_1, m_2, m_3, m_7\}$, $\{\bar{x}_1, x_2x_3\}$

Essential prime implicant: A prime implicant that covers a 1 on KMAP that is not covered by any other prime implicant

Algorithm for SOP Minimization

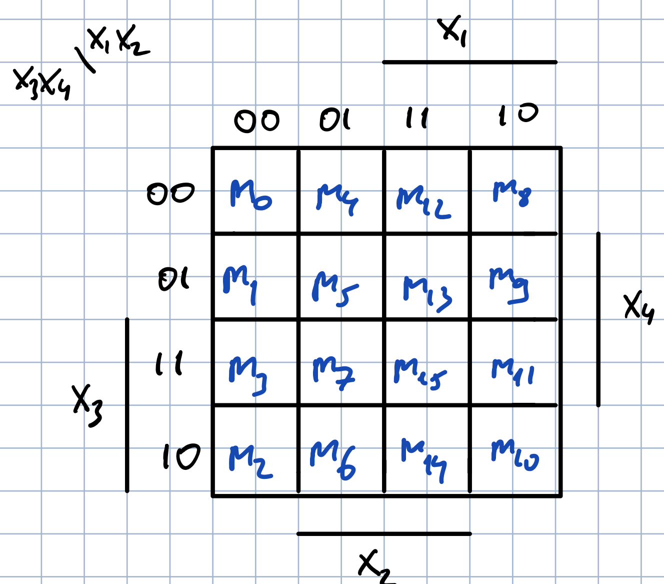
- Find all prime implicants (PI) $\bar{x}_1$, $x_2 \cdot x_3$
- Find all essential prime implicants (EPI) $\bar{x}_1$, $x_2 x_3$
- Form a cover such that
  - It includes all EPIs $[\bar{x}_1 , x_2 \cdot x_3]$ stopped.
  - If some 1s remain uncovered, include PIs that cover them.

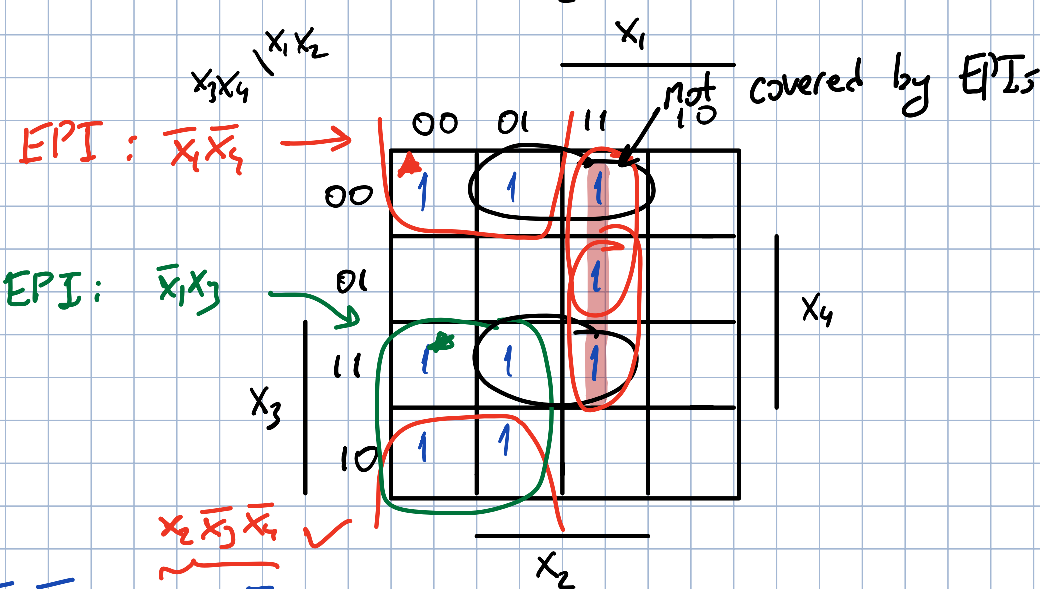
$f = \bar{x}_1 + x_2 \cdot x_3$ minimal SOP form of f.



4 variable K-MAP

	x_1	x_2	x_3	x_4	g
m_0	0	0	0	0	1
m_1	0	0	0	1	0
m_2	0	0	1	0	1
m_3	0	0	1	1	1
m_4	0	1	0	0	1
m_5	0	1	0	1	0
m_6	0	1	1	0	1
m_7	0	1	1	1	1
m_8	1	0	0	0	0
m_9	1	0	0	1	0
m_{10}	1	0	1	0	0
m_{11}	1	0	1	1	0
m_{12}	1	1	0	0	1
m_{13}	1	1	0	1	1
m_{14}	1	1	1	0	0
m_{15}	1	1	1	1	1





$g = \bar{x}_1 x_3 + x_1 x_4 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_4$ ✓

$x_2 x_3 x_4$

3 minimal SOP forms

$g = \bar{x}_1 \bar{x}_4 + \bar{x}_4 x_3$

Other exercises:

f:

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	1	1	1	
01	1	1	1	
11			1	1
10			1	1

x_1 (columns 11, 10); x_2 (columns 01, 11); x_3 (rows 11, 10); x_4 (rows 01, 11)

f:

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	1			1
01	1	1	1	1
11				
10	1			1

x_1 (columns 11, 10); x_2 (columns 01, 11); x_3 (rows 11, 10); x_4 (rows 01, 11)

$\bar{x}_3 x_4$

$\bar{x}_2 \bar{x}_4$

$f = \bar{x}_2 \bar{x}_4 + \bar{x}_3 x_4$

POS Minimization

g

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	1	1	1	
01			1	
11	1	1	1	
10	1	1		

$\bar{g}$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00				1
01	1	1		1
11				1
10			1	1

$\rightarrow x_1 \cdot \bar{x}_2$

$\bar{x}_1 \bar{x}_3 x_4 \rightarrow$

$\rightarrow x_1 x_3 \bar{x}_4$

$g = \bar{\bar{g}}$

* Find minimal SOP of $\bar{g}$. Use DeMorgan's theorem to get minimal POS of g.

$$\bar{g} = x_1\bar{x}_2 + \bar{x}_1\bar{x}_3x_4 + x_1x_3\bar{x}_4$$

$$g = \bar{\bar{g}} = \overline{x_1\bar{x}_2 + \bar{x}_1\bar{x}_3x_4 + x_1x_3\bar{x}_4} = (\bar{x}_1 + x_2).(x_1 + x_3 + \bar{x}_4).(\bar{x}_1 + \bar{x}_3 + x_4)$$

Alternative :

g

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	1	1	1	0
01	0	0	1	0
11	1	1	1	0
10	1	1	0	0

$\rightarrow \bar{X}_1 + X_2$

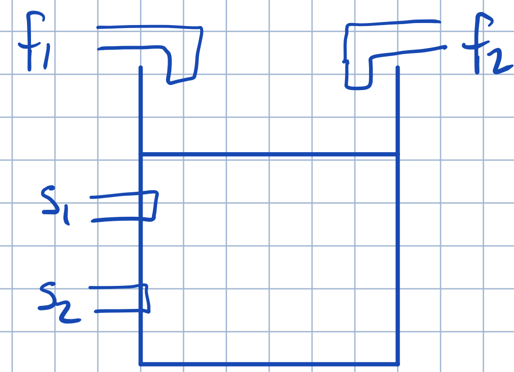
$X_1 + X_3 + X_4 \rightarrow$

$\rightarrow \bar{X}_1 + \bar{X}_3 + X_4$

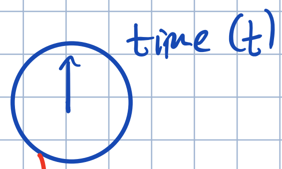
Incompletely Specified Functions

Design Example

- when time = 1
  - both faucets are turned on if pool is empty
  - only faucet 1 is turned on when pool is half empty
  - both faucets are off when pool is full
- when time = 0
  - faucets are always off



sensor =1 if contacts with water, 0 otherwise.



inputs / outputs

t	s_1	s_2	f_1	f_2	f_1	f_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	d	d	0	0
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	0	1	1	0	1	0
1	1	0	d	d	0	0
1	1	1	0	0	0	0

(safest options)

impossible

don't care

without don't care $f_2 = t\bar{s}_2\bar{s}_1$

f_1:

$s_2 \backslash ts_1$	00	01	11	10
0		d	d	1
1				1

f_2:

$s_2 \backslash ts_1$	00	01	11	10
0		d	d	1
1				

$f_1 = t\bar{s}_1$

$f_2 = t\bar{s}_2$

$\in \{0, 1, d\}$

```
entity pool is
   port(s1, s2, t: in std_logic;
        f1, f2: out std_logic);
end pool;

architecture pool_arch of pool is
begin
   f1 <= t and (not s1);
   f2 <= t and (not s2);
end pool_arch;
```