Multiple-valued Function

Multiple-valued Function

A multiple-valued function

$$f : P \Rightarrow \{0,1,2\} \text{ where } P = \mathop{X}_{i=1}^{n} p_i$$

Each p_i is a set of integers $\{1,2,...,p_i\}$ that the ith variable can assume

Ex : $n = 3$, $(p_1\ p_2\ p_3) = (3\ 5\ 2)$

$V = (2\ 4\ 1)$ is a minterm

$V = (4\ 4\ 1)$ is illegal

Multiple-valued Function

- To represent multiple valued variable using two-valued variables : one hot encoding

For a variable which can take pi values, we associate pi Boolean variables

Ex:

$(p_1\ p_2\ p_3) = (3\ 5\ 2)$

(__ __ __) (__ __ __ __ __) (__ __)

Ex:

n = 3 $(p_1\ p_2\ p_3) = (3\ 5\ 2)$

V= (2 4 1) will be represented

(0 1 0)(0 0 0 1 0)(1 0)

This should be thought of as a "minterm"

Product Term

- A general product term

  C = (1 1 0)(0 1 1 0 1)(1 0) means

  (V_1 = 1 or 2) and (V_2 = 2 , 3, or 5) and

  (V_3 =1)
- The problem of multi-valued logic minimization is to find an above form of minimized number of product term

Multi-valued Logic Minimization

Three questions to answer:

1. When to use a multiple-valued logic?

2. How to use a two-valued logic minimizer to minimize multiple valued logic?

3. How to realize a multiple-valued logic?

Multi-valued Logic Minimization

1. When to use a multiple-valued logic ?
   - State assignment to find adjacency relations
   - Allowing bit pairing to minimize logic

Multi-valued Logic Minimization

2. How to use a two-valued logic minimizer to minimize multiple valued logic?

   Ex:

   (1 0 0)(1 0 1 0)(1 0)

   \+ (0 1 0)(1 0 1 0)(1 0)

   => (1 1 0)(1 0 1 0)(1 0)

   A product term involves both “AND” and “OR”. But in two-valued logic, a product term involves “AND” only.

Multi-valued Logic Minimization

- How to change "OR" relations to "AND" relations?

Solution => *don't care*

Ex:

Consider the second 4-valued variable

$V_2 = (1\ 0\ 1 0)$

- means $V_2 = 1$ or $V_2 = 3$
- use "AND" to represent the meaning

" V_2 not 2 AND not 4"

Two valued logic:

$V_2 = (2\ 0\ 2\ 0\ )$

=> (0 0 0 0) don't care

(0 0 1 0)

(1 0 0 0)

(1 0 1 0) don't care

Multi-valued Logic Minimization

- The *don't care* set?

	x_1	x_2	x_3
care set =	1	0	0
	0	1	0
	0	0	1

	x_1	x_2	x_3
don't care:	1	1	2
	1	2	1
	2	1	1
	0	0	0

No pair of two x_i are both on and x_i are never all off.

Multi-valued Logic Minimization

- Use two-valued logic minimizer to minimize a multi-valued logic:

  step (1) Create $\sum$ Pi Boolean variables

  step (2) For each multi-valued variable, we associate the Don't care set

  step(3) Espresso

  step(4) Convert the result back to multiple-valued function

Example: Input File

8-valued 4-valued

100000001000 | 10000000 - - -

001000001000 | 10000000 - - -

....

010000001000 | - - - - - - - - -000

000010001000 | - - - - - - - - -000

....

001000000100 | 10000000 - - -

000000100100 | 10000000 - - -

....

010000000100 | - - - - - - - - -000

....

001000000001 | - - - - - - - - -101

000000100001 | - - - - - - - - -101

Multi-valued Input Version of PLA

DK17(Don’t-cares not shown)

1. Adding don’t care for each variable
2. Call Espresso to minimize the two-valued logic

Example: Output of Espresso

0. 00. .0. 1 | - -1 - - - - - - -1

0000. .0. . . .1 | - -1 - - - - - - 1-

. 0. 00000. .1. | - 1- - - - - - - - -

. . 00. 000. 1.. | - - - 1- - - - - - -

....

. 00. 0000. . .1| - - - -1- - - - 1

00. 000. 0. 1. .| 1- - - - - - - -1 -

. . . . . 1. . . 1. .| - - - - - - - - 1- -

....

. 1 . . . .1| - - - - - 1- - - -

2 2 1 2

0 0 1 0
0 0 1 1
0 1 1 0
0 1 1 1
1 0 1 0
1 0 1 1
1 1 1 0
1 1 1 1

20020000

00000000
00010000
10000000
10010000

Multi-valued Input Minimization of DK17

Example: Converting Back to Multi-valued Logic

010011010010 | - - 1 - - - - - - - 1 -

000011010001 | - - 1 - - - - - 1 - -

101000000010 | - 1 - - - - - - - - - 1

110010000100 | - - - 1 - - - - - - - -

....

100100000001 | - - - - 1 - - - - 1 -

001000100100 | 1 - - - - - - - - 1 - 1

000001000100 | - - - - - - - - 1 - - -

....

000000101000 | - - - 1 - - - - - 1 -

000001001000 | - - - - - - - 1 - - - -

010000000001 | - - - - - 1 - - - - - -

MINI Representation of Minimized DK17

Multi-valued Logic Minimization

1. When to use a multiple-valued logic ?
   - State assignment to find adjacency relations
   - Allowing bit pairing to minimize logic

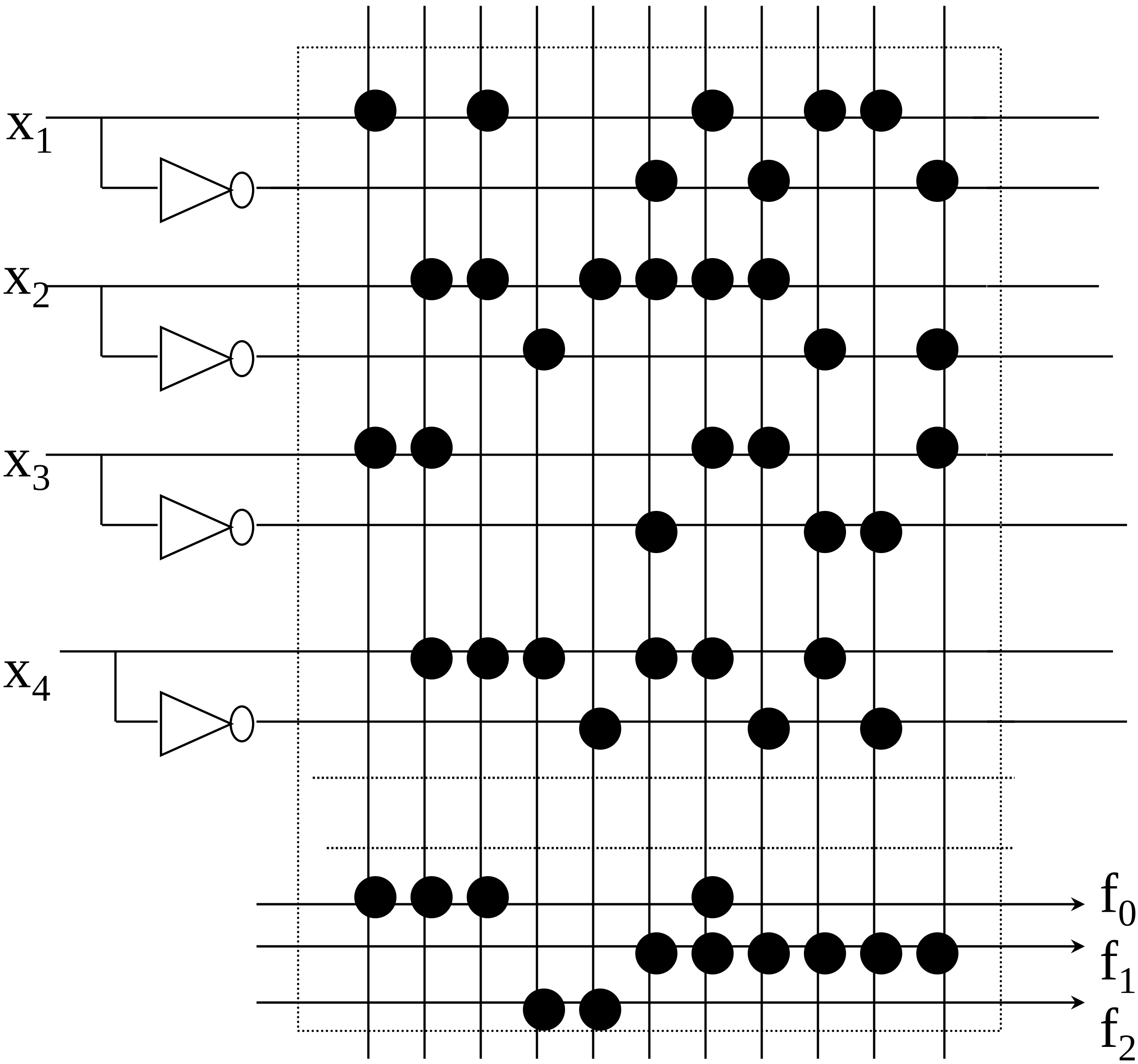
Example

$$\begin{array}{r} x_1\ x_2 \\ +\ x_3\ x_4 \\ \hline f_0\ f_1\ f_2 \end{array}$$

Two-Bit ADDER(ADR2)

x_1	x_2	x_3	x_4	f_0	f_1	f_2
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Example



11 products

Example

pair TABLE IV

$X_1=(x_1\ x_2)$ Two-Bit ADDER(ADR2)

$X_2=(x_3\ x_4)$

pair	x_1	x_2	x_3	x_4		f_0	f_1	f_2
	0	0	0	0	**1**	0	0	0
1	0	0	0	1	**2**	0	0	1
	0	0	1	0	**3**	0	1	0
	0	0	1	1	**4**	0	1	1
	0	1	0	0		0	0	1
2	0	1	0	1		0	1	0
	0	1	1	0		0	1	1
	0	1	1	1		1	0	0
	1	0	0	0		0	1	0
3	1	0	0	1		0	1	1
	1	0	1	0		1	0	0
	1	0	1	1		1	0	1
	1	1	0	0		0	1	1
4	1	1	0	1		1	0	0
	1	1	1	0		1	0	1
	1	1	1	1		1	1	0

Example: One-hot Encoding

X_1	X_2	$f_0f_1f_2$
1000----	0100----	0 0 1
1000----	0010----	0 1 0
1000----	0001----	0 1 1
0100----	1000----	0 0 1
0100----	0100----	0 1 0
0100----	0010----	0 1 1
0100----	0001----	1 0 0
0010----	1000----	0 1 0
0010----	0100----	0 1 1
0010----	0100----	1 0 0
0010----	0001----	1 0 1
0001----	1000----	0 1 1
0001----	0100----	1 0 0
0001----	0010----	1 0 1
0001----	0001----	1 1 0

Example: After Optimization

$$\begin{bmatrix} \text{0111----0001----1 0 0} \\ \text{0011----0011----1 0 0} \\ \text{0001----0111----1 0 0} \\ \text{1010----0101----0 0 1} \\ \text{0101----1010----0 0 1} \\ \text{1100----0010----0 1 0} \\ \text{0110----0100----0 1 0} \\ \text{1001----0001----0 1 0} \\ \text{0011----1000----0 1 0} \end{bmatrix}$$

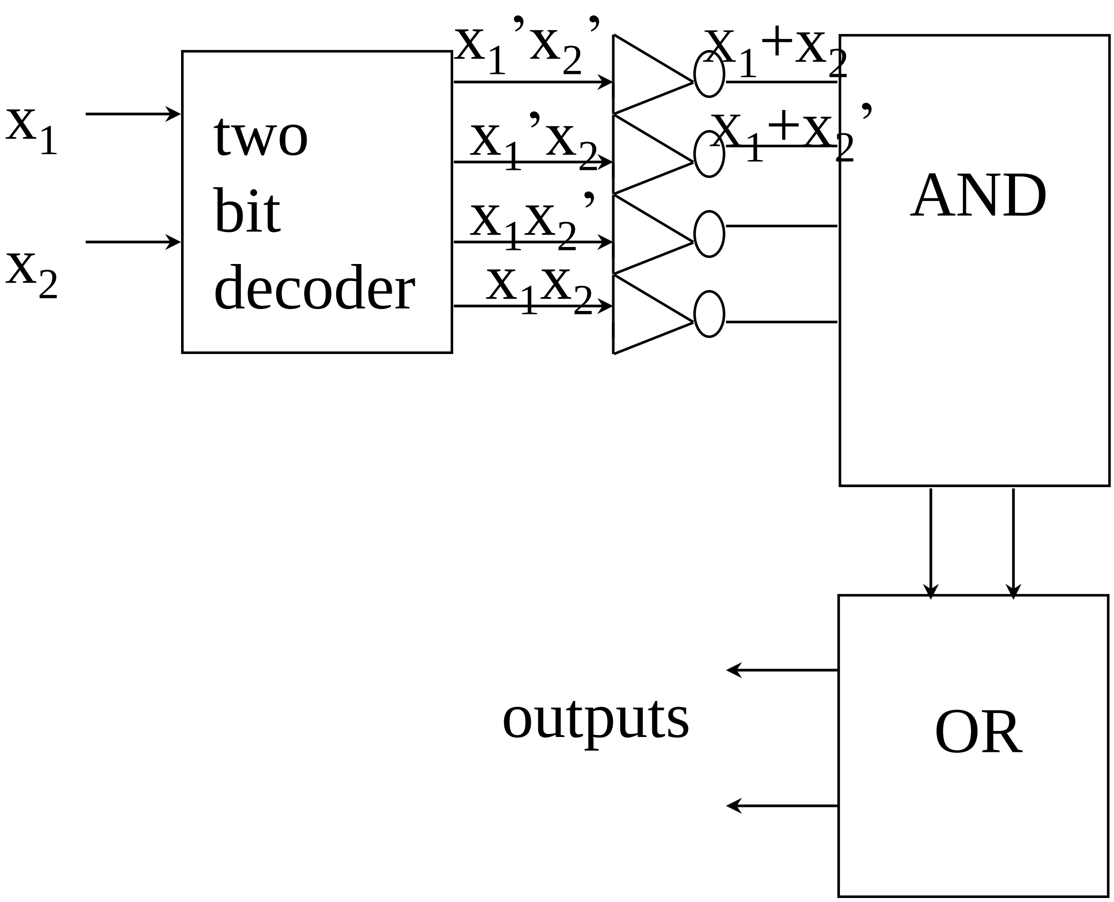
9 products < 11 products

Multiple-valued Logic Minimization

3. How to realize a multiple-valued logic?
   - Using decoder to generate multiple value logic

   Ex : Using a two-bit decoder to generate

   4-value logic



Multi-valued Logic Minimization

		$x_0\ x_1$	f		
$x_0'x_1'$	m_0	0 0	0	M_0	x_0+x_1
$x_0'x_1$	m_1	0 1	0	M_1	x_0+x_1'
x_0x_1'	m_2	1 0	1	M_2	$x_0'+x_1$
x_0x_1	m_3	1 1	0	M_3	$x_0'+x_1'$

Implement f:

- Use minterms $f = x_0x_1' = \Sigma\ (m_2)$
- Use maxterms

$$f' = (x_0'x_1')+(x_0'x_1)+(x_0x_1)$$

$$f = (x_0+x_1)\ \ (x_0+x_1')\ \ (x_0'+x_1')$$

$$= M_0 \cdot M_1 \cdot M_3$$

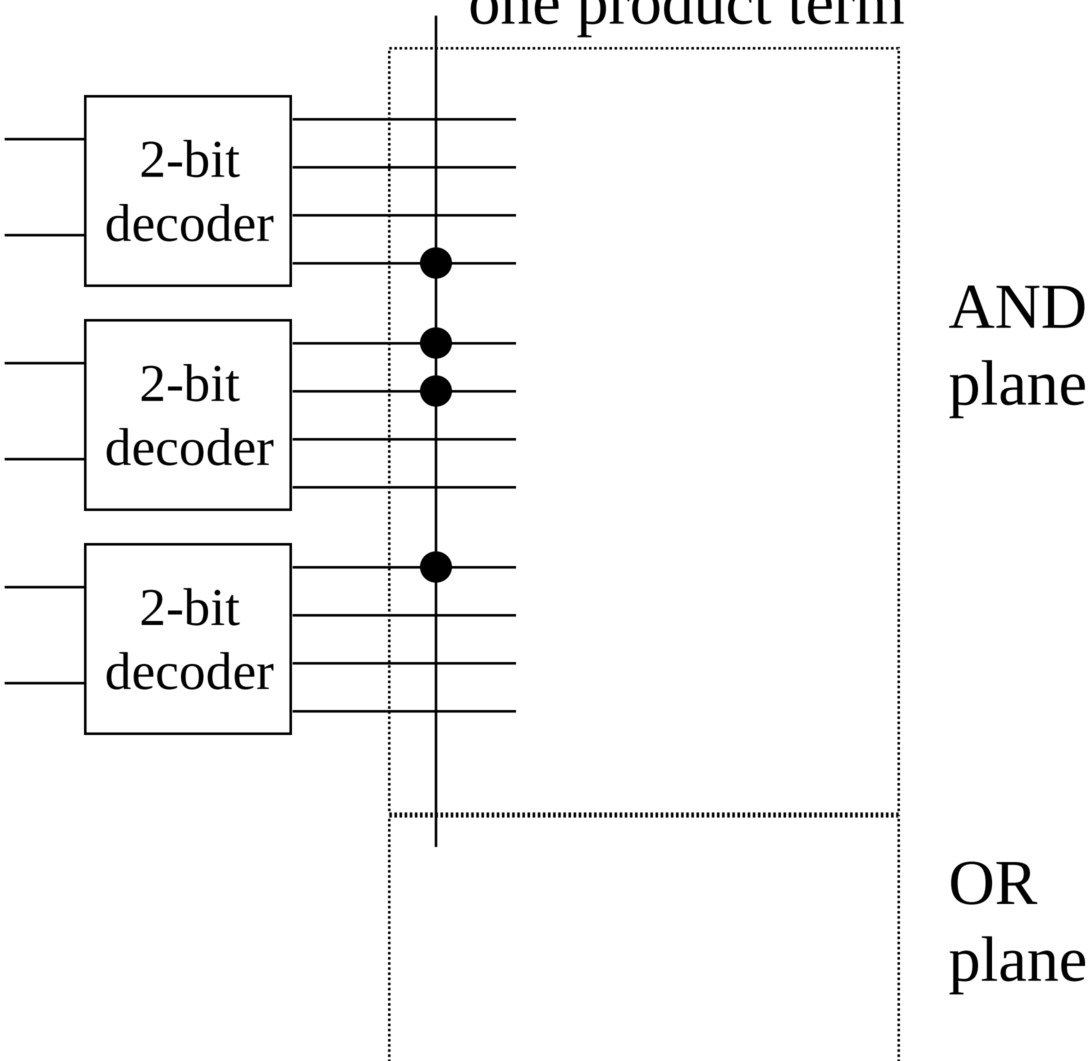
$$= \Pi\ (M_0\ , M_1\ , M_3\ )$$

Example

Ex : multiple output function

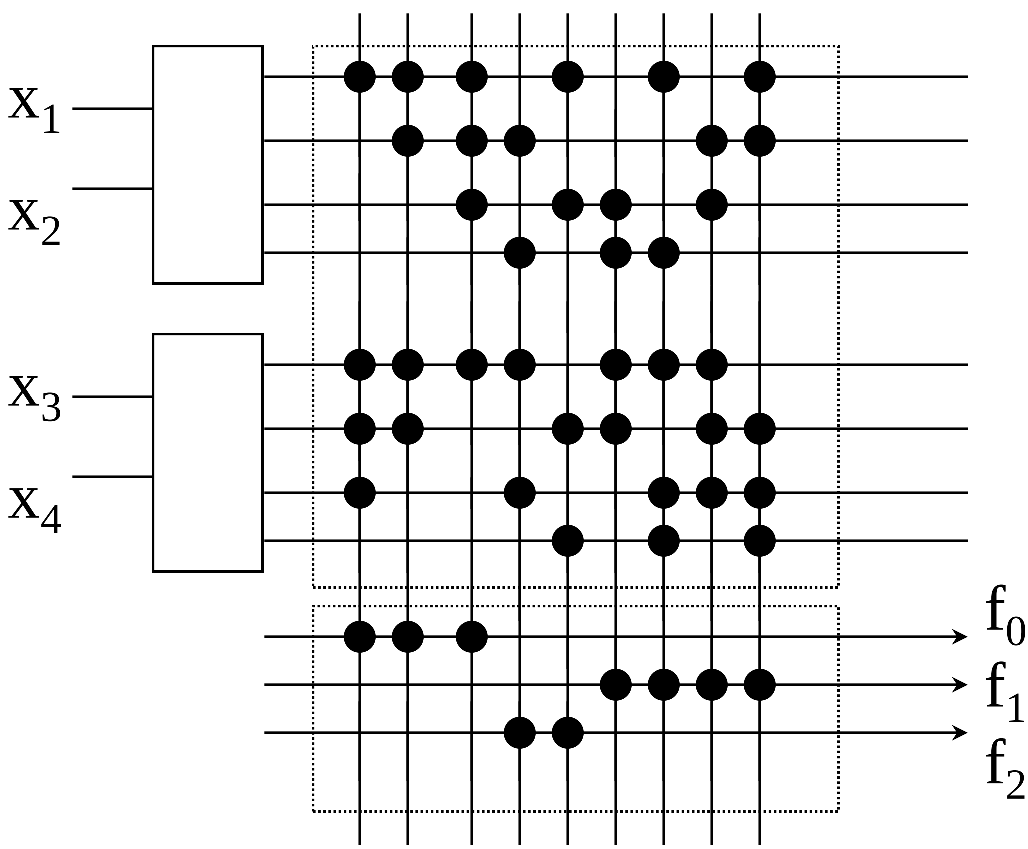
$f = (p_1\ p_2\ p_3) = (1110)(0011)(0111)$

$p_1\ p_2\ p_3 \in \{1,2,3,4\}$



Example

$$
\begin{pmatrix}
0111\text{----}0001\text{----}1\ 0\ 0 \\
0011\text{----}0011\text{----}1\ 0\ 0 \\
0001\text{----}0111\text{----}1\ 0\ 0 \\
1010\text{----}0101\text{----}0\ 0\ 1 \\
0101\text{----}1010\text{----}0\ 0\ 1 \\
1100\text{----}0010\text{----}0\ 1\ 0 \\
0110\text{----}0100\text{----}0\ 1\ 0 \\
1001\text{----}0001\text{----}0\ 1\ 0 \\
0011\text{----}1000\text{----}0\ 1\ 0
\end{pmatrix}
$$



PLA for ADR2(input variable assignment nonoptimized).