ESc201, Lecture 29: Intro to Digital

3-variable Karnaugh-map representation

x	y	z	min terms		x	y	z	f
0	0	0	$\bar{x}.\bar{y}.\bar{z}$	m0	0	0	0	0
0	0	1	$\bar{x}.\bar{y}.z$	m1	0	0	1	1
0	1	0	$\bar{x}.y.\bar{z}$	m2	0	1	0	0
0	1	1	$\bar{x}.y.z$	m3	0	1	1	1
1	0	0	$x.\bar{y}.\bar{z}$	m4	1	0	0	0
1	0	1	$x.\bar{y}.z$	m5	1	0	1	1
1	1	0	$x.y.\bar{z}$	m6	1	1	0	0
1	1	1	$x.y.z$	m7	1	1	1	1

Sum of Products form: f = Z

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

x \ yz	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Product of Sums form: f = $\bar{Z}$

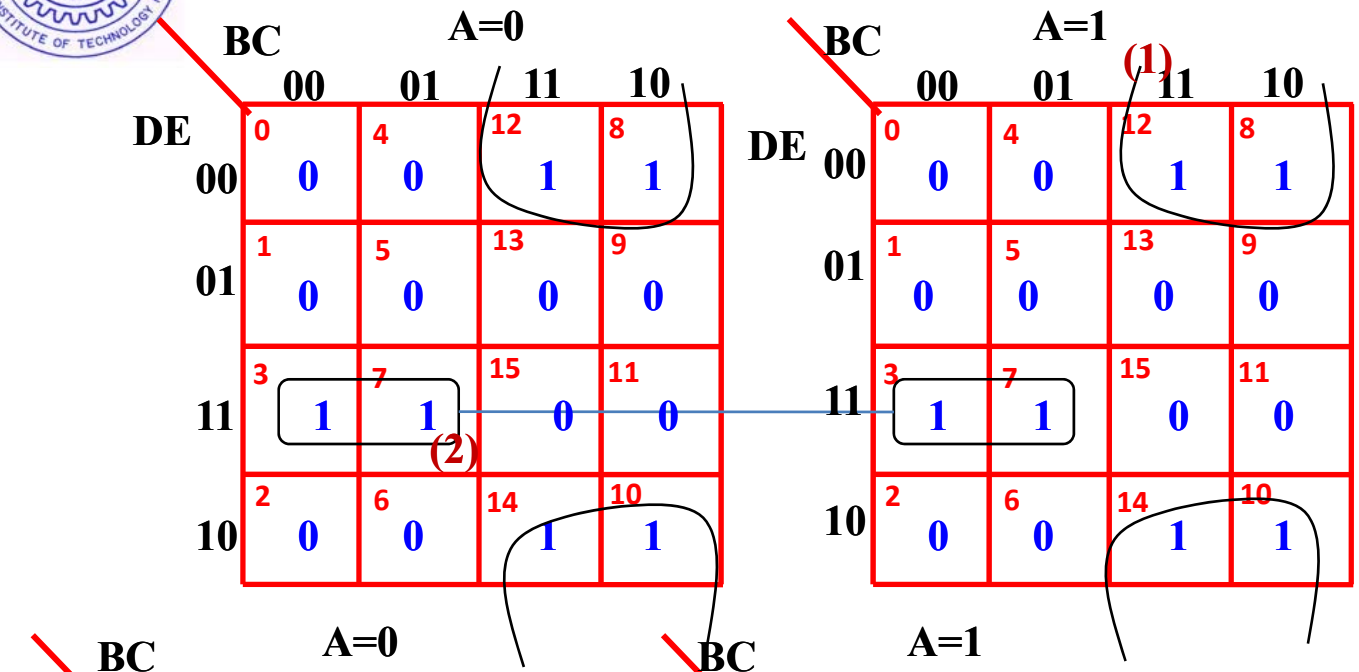
x \ yz	00	01	11	10
0	0	1	1	0
1	0	1	1	0

wx \ yz	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	0

$$f = \bar{w}.y.z + w.x.z + \bar{w}.x.\bar{z} + \bar{x}.\bar{y}.\bar{z} = \Pi(M_i)$$

ESc201, Lecture 29: Intro to Digital: Minimization of Boolean functions:

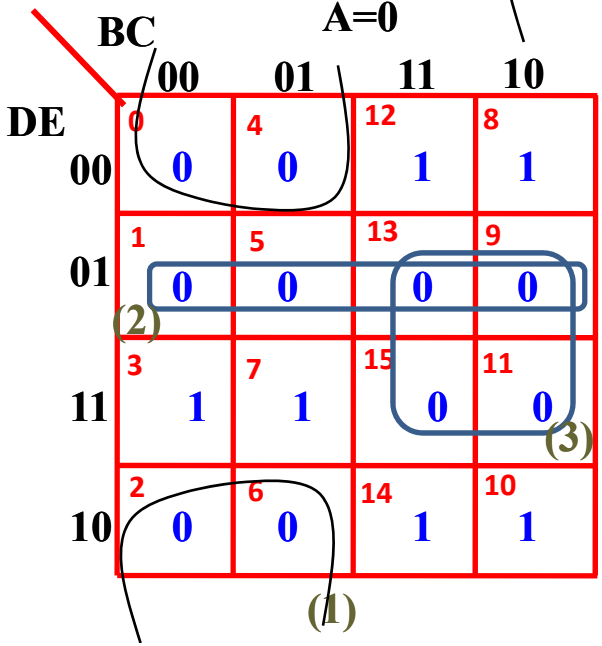
Let the K-map lof a five variable function (A,B,C,D,E) look like :

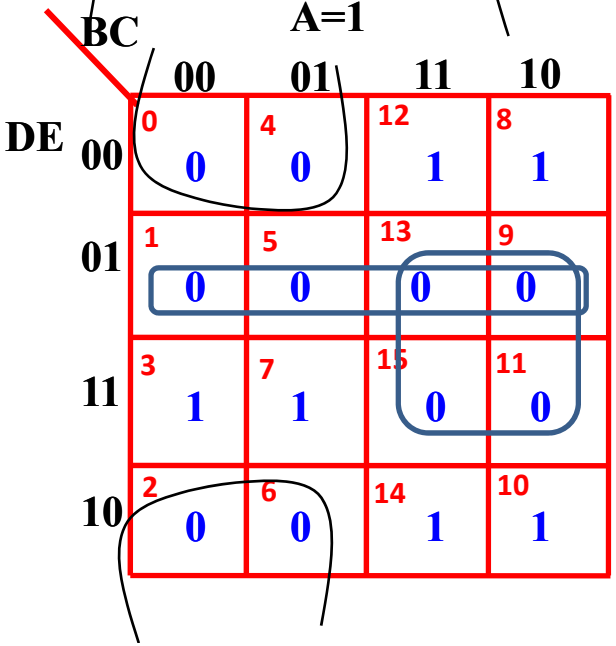


BCDE Decimal
0000 → 0
0001 → 1
0010 → 2
0011 → 3

Sum of Products (SOP) form:

$f = \bar{B}\,\bar{E} + \bar{B}DE$





Product of Sums (POS) form:

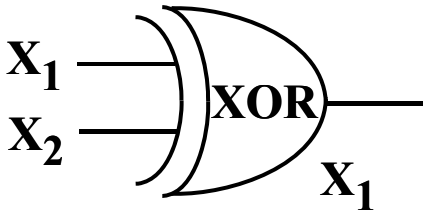
$f = (B + \bar{E})(\bar{D} + E)(B + E)$

When the number of variables become large, this kind of graphical means is no longer suitable.

One need to fall back on a tabular technique called the QUINE McCLUSKY method.

ESc201, Lecture 29: Intro to Digital

Exclusive OR Gate (XOR)



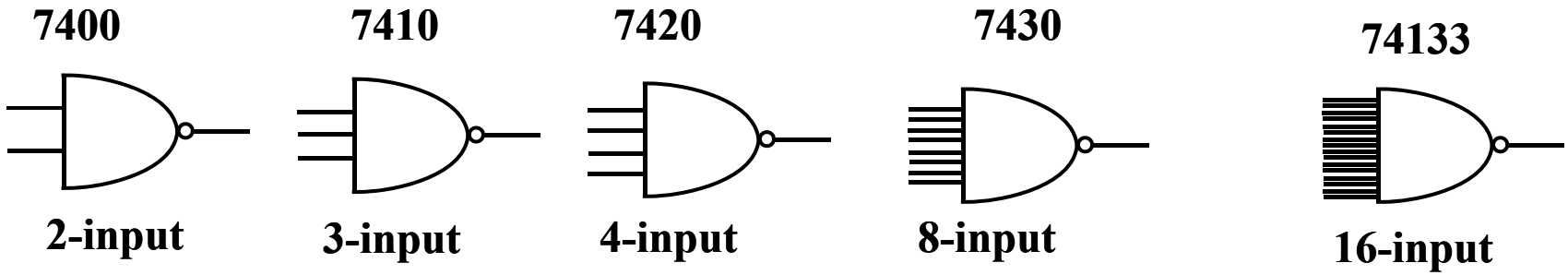
X_1	0	0	1	1
X_2	0	1	0	1
Y	0	1	1	0

$$X \oplus 0 = X$$
$$X \oplus 1 = \overline{X}$$
$$X \oplus X = 0$$
$$X \oplus \overline{X} = 1$$

$$X \oplus Y \oplus Z = X \oplus Z \oplus Y = Z \oplus X \oplus Y = Y \oplus Z \oplus X$$
$$X \bullet (Y \oplus Z) = (X \bullet Y) \oplus (X \bullet Z) = X \bullet Y \oplus X \bullet Z$$
$$X \bullet (\overline{X} \oplus Y) = X \bullet Y$$
$$X \oplus Y = \overline{X} \oplus \overline{Y} = \overline{X \odot Y}$$ ($\overline{XNOR}$, XNOR is dual of XOR)
$$\overline{X \oplus Y} = \overline{X} \odot \overline{Y} = X \odot Y$$

Fan-in and Fan-out

Fan-in of some TTL family chips are given



If the number of inputs are less than the number of Fan-in, it is not a good idea to keep them floating, as it would give rise to erroneous results due to noise. The result would be unaltered usually if two inputs are tied together. i.e. 8-inputs can be reduced to 4-inputs, etc.

Fan-out is decided by the capacity of the output stage of the chip (usually of a transistor) to drive current to the next stage. For every chip the maximum number of Fan-out is specified. Connecting more than this number would reduce the noise margin and ultimately result in erronous logic.

ESc201, Lecture 29: Intro to Digital: Binary numbers

Consider the number 7689 again and mention against each number the weight of its position and the position of each digit in the number as: $7x10^3+6x10^2+8x10^1+9x10^0$. That's why the lowest value is 0 and not one for the units column as $9x10^1$ will not give the correct value. Similarly for a decimal number 7689.5045 it is : $7x10^3+6x10^2+8x10^1+9x10^0+5x10^{-1}+0x10^{-2}+4x0^{-3}+5x10^{-4}$.

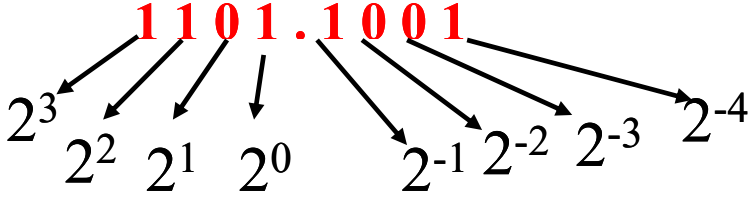
The blue values would give the digits after the decimal point.

The term 'binary' indicates any two-state device or signal and can be represented as 'on' or 'off ' , 'closed' or 'open', 'true' or 'false'. It is most common to represent the binary nature by '0' or '1'. Depending on the convention adopted, any meaning can be assigned to '0' and '1'. If '0' is assumed to be the 'off ' state, '1' is automatically the 'on' state, or it may be the other way around. From Philosophy by convention TRUE=1, FALSE=0 and hence this convention is used. Hence a binary counting system can only have two digits i.e. Only 0 and 1, very much like the decimal system which has 0, 1, 2, ..., 8, 9 for 10 states. Hence a binary number given using a BINARY POINT is let's say 101101001.011101

It represents a number but we are accustomed to decimal number so that it only gives us a feel if we convert it to a decimal number as: $101101001.011101_2 = (1x2^8+0x2^7+1x2^6+1x2^5+0x2^4+1x2^3+0x2^2+0x2^1+1x2^0+0x2^{-1}+1x2^{-2}+1x2^{-3}+1x2^{-4}+0x2^{-5}+1x2^{-6})_{10}$.

This is also how a binary is converted to a decimal number representation.

What about the reverse? Decimal to a binary representation !!



13. 5625

ESc201, Lecture 29: Intro to Digital

Converting decimal to binary number

Method of successive division by 2

147	remainder
73	1
36	1
18	0
9	0
4	1
2	0
1	0
0	1

LSB or Least significant bit

MSB or Most significant bit

1 0 0 1 0 0 1 1 =147.635=.1 0 1 0 0 0 1 1 0 1 0 _

Method of successive multiplication by 2

Carry over	635
1	270
0	540
1	080
0	160
0	320
0	640
1	280
0	560
1	120
0	240
0	480
0	960
1	120

No end to it ????
The sequence 120-960 continues and the number of significant digits depend on how accurately the number 635 is required.

Developing Fluency with Binary Numbers

11001 = ? 25

1100001 = ? 64+32+1=97

0.101 = ? 0.5+0.125=0.625

11.001 = ? 3+0.125=3.125

ESc201, Lecture 29: Intro to Digital Converting binary to decimal number

$(110100111.1000111)_2=[1x2^8+1x2^7+0x2^6+1x2^5+0x2^4+0x2^3+1x2^2+1x2^1+1x2^0+$
$1x2^{-1}+0x2^{-2}+0x2^{-3}+0x2^{-4}+1x2^{-5}+1x2^{-6}+1x2^{-7}]_{10}$
$= 423.5546875$

For big decimal numbers, conversion to Binay may lead to unusually very large number of bits, as an extra bit in the binary multiplies the number by only 2 whereas in the decimal the existing number becomes 10 time.

To remedy this problem the number may be expressed in higher OCTAL or HEXADECIMAL system.

An Octal number system has a base 8 and uses symbols (0,1,2,3,4,5,6,7). So three binary bits would be required to represent each number of the Octal number system.

$$(2007)_8 = 2\times8^3+0\times8^2+0\times8^1+7\times8^0=(1031)_{10}$$

Whereas a hexadecimal system has a base of 16 [Note that a minimum number of bits required would be 4-binary bits to represent each number in the hexadecimal system . 4-bits in digital parlance is called a byte].

However to represent 16 states (0-15) in a single digit is impossible. Hence 0-9 is used followed by A, B, C, D, E, as shown below.

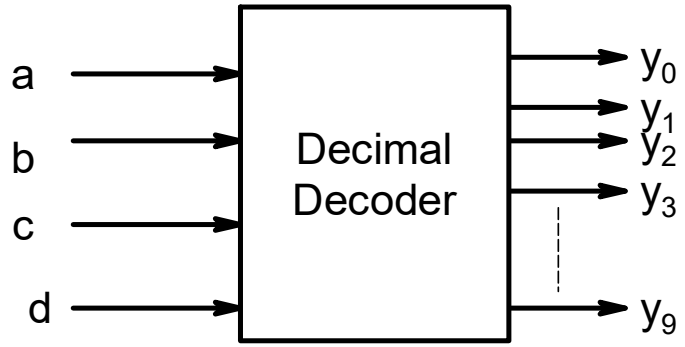
$$(2BC9)_{16} = 2\times16^3+B\times16^2+C\times16^1+9\times16^0=2\times4096+11\times256+12\times16^1+9\times16^0=(11209)_{10}$$

Representing each unit of a base of 16 number in Binary, one has to use 4-binary bits for each HEX letter. i.e. $(2BC9)_{16}=(0010)_2\ (1011)_2(1100)_2(1001)_2=(0010101111001001)_2$

Remember that gates do not understand this high level numbering (it only understands Binary), but is used for convenience of writing

ESc201, Lecture 29: Intro to Digital: Binary Coded Decimal (BCD)

A minimum of 4-bits are required to represent 0-9 in decimal.



a	b	c	d	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0	x	x	x	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x	x	x	x

Don't care States

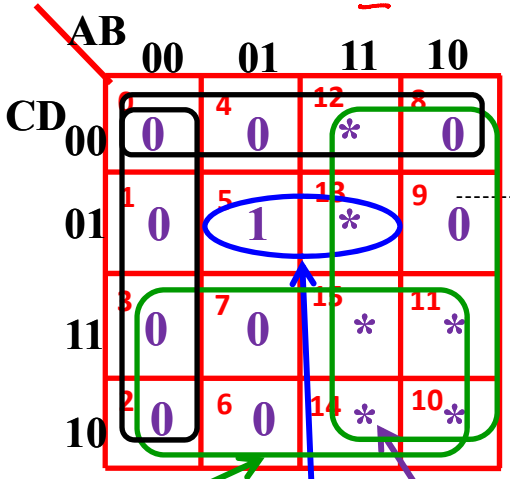
But with 4-bits the count would go from 0-15.
What happens to those above 9?

Seven Segment Display

y_5

Y_3

ab \ cd	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	x	x	x	x
10	0	0	x	x



$$y_3 = \bar{a}.\bar{b}.c.d$$

* Considered '0' for POS

* Considered '1' for SOP

Don't care States