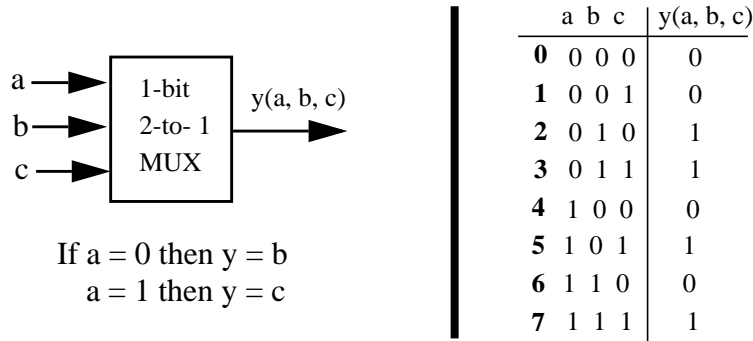


DIGITAL CIRCUIT DESIGN & ALGEBRAIC SIMPLIFICATION EXAMPLES

Example 1

1-bit 2-to-1 Multiplexer

A 1-bit 2-to-1 Multiplexer (MUX) is a selector which selects one of the two inputs based on a select signal. As seen below, it has three inputs and one output. Two inputs (b and c) are **data inputs**. The third input (a) is the **control input**, the select input. The single output is always equal to either b or c at any time. The input/output relationship (the operation, purpose) is given below in two different ways : A **textual** description and a **truth table**.

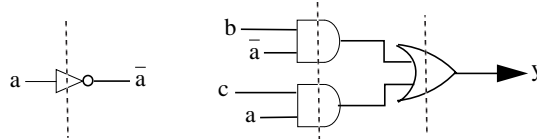


The MUX is a 1-bit MUX since when an input is selected, there is only one data line selected. The minterm list, canonical SOP and minimal SOP expressions are shown below.

$$y(a,b, c) = \sum m(2, 3, 5, 7) \left\{ \begin{array}{l} \overline{a} b \overline{c} + \overline{a} b c + a \overline{b} c + a b c \end{array} \right.$$

$$\begin{aligned} y(a, b, c) &= \overline{a} b (\overline{c} + c) + a c (\overline{b} + b) \longrightarrow k(m+p) = km + kp \\ &= \overline{a} b 1 + a c 1 \longrightarrow k + \overline{k} = 1 \\ &= \overline{a} b + a c \longrightarrow k1 = k \end{aligned}$$

As the gate network shows, the MUX has three gate delays :



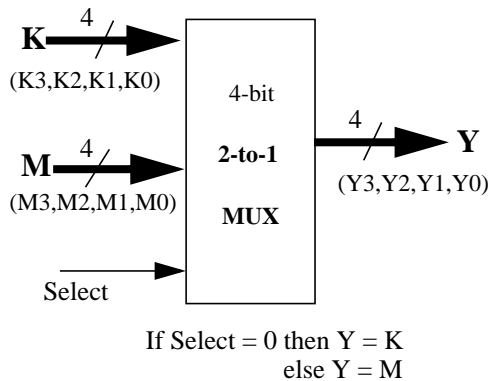
One can develop **k-bit** 2-to-1 MUXes from 1-bit 2-to-1 MUXes. For example, 2-bit 2-to-1 MUXes, 4-bit 2-to-1 MUXes, etc. are implemented by using 1-bit 2-to-1 MUXes. Example 2 shows how to implement a 4-bit 2-to-1 MUX by using four 1-bit 2-to-1 MUXes.

Note also that there are k-bit 4-to-1 MUXes, k-bit 8-to-1 MUXes, etc.

Example 2

4-bit 2-to-1 MUX

A 4-bit 2-to-1 MUX selects between two 4-bit inputs. As a black box, it has 9 inputs and 4 outputs as shown below. The single control input is Select and the 4-bit inputs, K and M, are data inputs. The 4-bit output is Y which is K if Select is 0 and M if Select is 1 :



Since there are 9 inputs, we need to partition it into simpler pieces ! We have to obtain the operation table of the 4-bit 2-to-1 MUX :

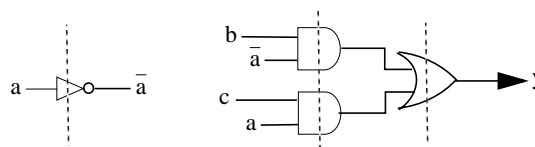
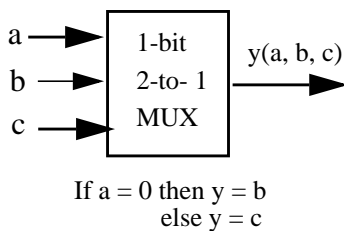
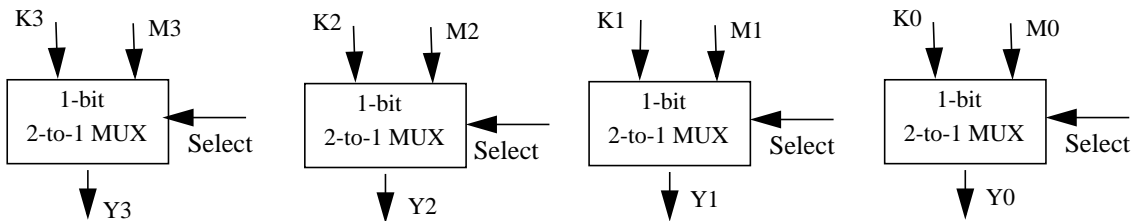
Select	Operation
0	Y = K
1	Y = M

} The major operations are not clear on this operation table. We need to get a **different**, more detailed operation table :

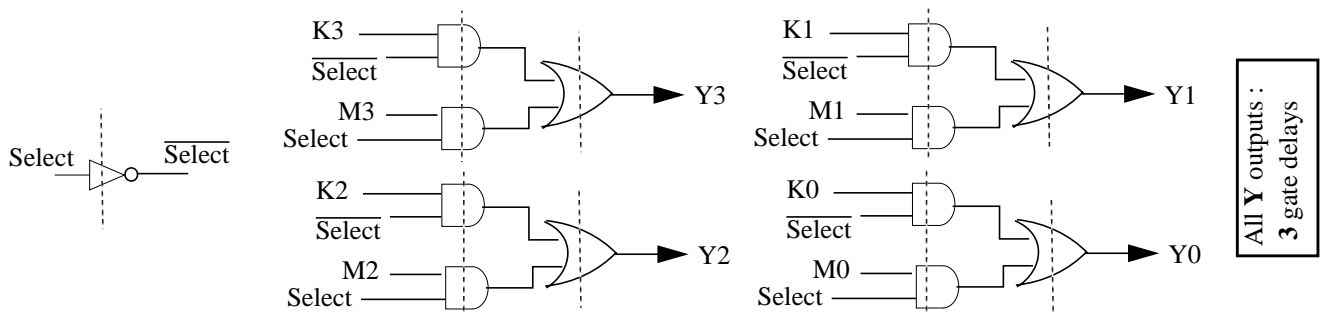
Select	Operation
0	Y3 = K3 ; Y2 = K2 ; Y1 = K1 ; Y0 = K0
1	Y3 = M3 ; Y2 = M2 ; Y1 = M1 ; Y0 = M0

There are four identical major operations : 1-bit 2-to-1 MUXing !

We partition the 4-bit 2-to-1 MUX into four blocks. Each block is a 1-bit 2-to-1 MUX which we have already designed by using Switching Algebra : It has three inputs and one output :



The 4-bit 2-to-1 MUX is then as follows :

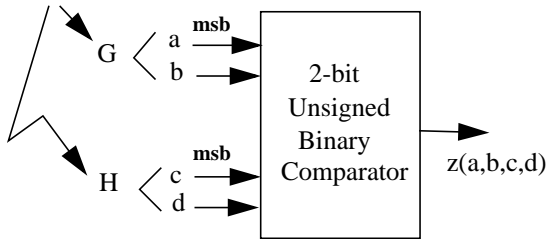


Example 3

$$\begin{aligned}
 z(a,b,c,d) &= a(b + bc) + a(\bar{d} + d\bar{c}) + \bar{c} \bar{d}(\bar{d} + (a+\bar{a})) + abc\bar{c} + \bar{a}b\bar{c} && \text{a nonminimal expression} \\
 &= a(b + bc) + a(\bar{d} + \bar{c}) + \bar{c} \bar{d}(\bar{d} + (a+\bar{a})) + abc\bar{c} + \bar{a}b\bar{c} && k + \bar{k}m = k + m \\
 &= a(b + bc) + a\bar{d} + a\bar{c} + \bar{c} \bar{d}(\bar{d} + (a + \bar{a})) + abc\bar{c} + \bar{a}b\bar{c} && k(m+n) = km + kn \\
 &= a(b + bc) + a\bar{d} + a\bar{c} + \bar{c} \bar{d}(\bar{d} + 1) + abc\bar{c} + \bar{a}b\bar{c} && k + \bar{k} = 1 \\
 &= a(b + bc) + a\bar{d} + a\bar{c} + \bar{c} \bar{d}1 + abc\bar{c} + \bar{a}b\bar{c} && k + 1 = 1 \\
 &= a(b + bc) + a\bar{d} + a\bar{c} + \bar{c} \bar{d} + abc\bar{c} + \bar{a}b\bar{c} && k1 = k \\
 &= a(b + bc) + a\bar{d} + a\bar{c} + \bar{c} \bar{d} + bc\bar{c}(a + \bar{a}) && k(m+n) = km + kn \\
 &= a(b + bc) + a\bar{d} + a\bar{c} + \bar{c} \bar{d} + bc\bar{c}1 && k + \bar{k} = 1 \\
 &= a(b + bc) + a\bar{d} + a\bar{c} + \bar{c} \bar{d} + bc\bar{c} && k1 = k \\
 &= ab + a\bar{d} + a\bar{c} + \bar{c} \bar{d} + bc\bar{c} && k + km = k
 \end{aligned}$$

$ab + a\bar{d} + a\bar{c} + \bar{c} \bar{d} + bc\bar{c}$ → the minimal SOP expression

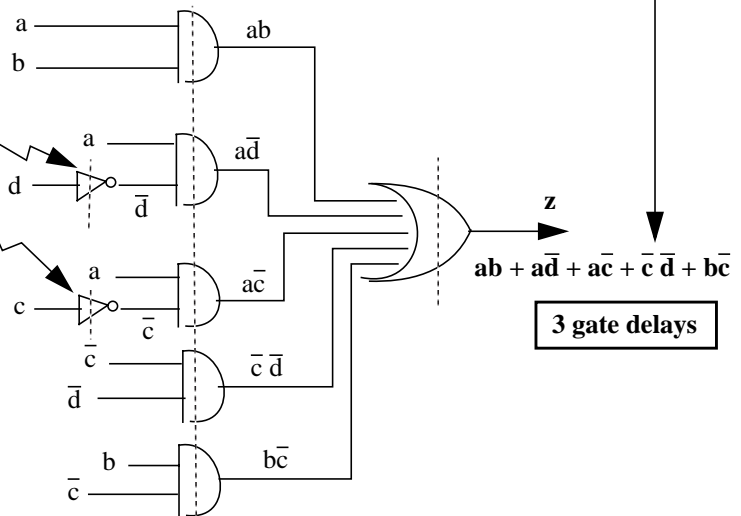
Single-rail inputs



G and H are 2-bit Unsigned Binary numbers

if G is > H then $z(a, b, c, d) = 1$
 else $z(a, b, c, d) = 0$

Single-rail inputs : Inverters needed



Example 4

$$w(a,b,c,d) = \mathbf{bcd} + \overline{a}\overline{b}\overline{c} + \mathbf{bd}(\overline{a}\overline{c} + \overline{a}\overline{c}) + c(\overline{a}\overline{b} + \overline{a}\overline{d} + d) \quad \text{a nonminimal expression}$$

$$= \mathbf{bd}(\mathbf{c} + \overline{a}\overline{c} + \overline{a}\overline{c}) + \overline{a}\overline{b}\overline{c} + c(\overline{a}\overline{b} + \overline{a}\overline{d} + d) \quad k(m+n) = km+kn$$

$$= \mathbf{bd}(\mathbf{c} + \mathbf{a} + \overline{\mathbf{a}}) + \overline{a}\overline{b}\overline{c} + c(\overline{a}\overline{b} + \overline{a}\overline{d} + d) \quad k + \overline{k}m = k + m$$

$$= \mathbf{bd}(\mathbf{c} + \mathbf{1}) + \overline{a}\overline{b}\overline{c} + c(\overline{a}\overline{b} + \overline{a}\overline{d} + d) \quad k + \overline{k} = 1$$

$$= \mathbf{bd1} + \overline{a}\overline{b}\overline{c} + c(\overline{a}\overline{b} + \overline{a}\overline{d} + d) \quad k + 1 = 1$$

$$= \mathbf{bd} + \overline{a}\overline{b}\overline{c} + \mathbf{c}(\overline{a}\overline{b} + \overline{a}\overline{d} + \mathbf{d}) \quad k1 = k$$

$$= \mathbf{bd} + \overline{a}\overline{b}\overline{c} + \overline{a}\overline{b}\mathbf{c} + \overline{a}\overline{c}\mathbf{d} + \mathbf{cd} \quad k(m+n) = km + kn$$

$$= \mathbf{bd} + \overline{a}\overline{b}(\overline{c} + \mathbf{c}) + \overline{a}\overline{c}\mathbf{d} + \mathbf{cd} \quad k(m+n) = km + kn$$

$$= \mathbf{bd} + \overline{a}\overline{b}\mathbf{1} + \overline{a}\overline{c}\mathbf{d} + \mathbf{cd} \quad k + \overline{k} = 1$$

$$= \mathbf{bd} + \overline{a}\overline{b} + \overline{a}\overline{c}\mathbf{d} + \mathbf{cd} \quad k1 = k$$

$$= \mathbf{bd} + \overline{a}\overline{b} + \mathbf{c}(\mathbf{d} + \overline{a}\overline{d}) \quad k(m+n) = km + kn$$

$$= \mathbf{bd} + \overline{a}\overline{b} + \mathbf{c}(\mathbf{d} + \overline{\mathbf{a}}) \quad k + \overline{k}m = k + m$$

$$\longrightarrow \mathbf{bd} + \overline{a}\overline{b} + \mathbf{cd} + \overline{a}\overline{c} \quad k(m+n) = km + kn \longrightarrow \text{It seems we cannot simplify more !}$$

$$\longrightarrow \mathbf{bd} + \overline{a}\overline{b} + \mathbf{cd}(\mathbf{a} + \overline{\mathbf{a}})(\mathbf{b} + \overline{\mathbf{b}}) + \overline{a}\overline{c} \quad k(m+\overline{m}) = k \longrightarrow \text{We make it more complex !}$$

$$= \mathbf{bd} + \overline{a}\overline{b} + (\mathbf{acd} + \overline{a}\overline{c}\mathbf{d})(\mathbf{b} + \overline{\mathbf{b}}) + \overline{a}\overline{c} \quad k(m+n) = km + kn$$

$$= \mathbf{bd} + \overline{a}\overline{b} + \mathbf{abcd} + \overline{a}\overline{b}\mathbf{cd} + \overline{a}\overline{b}\mathbf{cd} + \overline{a}\overline{b}\mathbf{cd} + \overline{a}\overline{c} \quad k(m+n) = km + kn$$

$$= \mathbf{bd} + \overline{a}\overline{b} + \mathbf{abcd} + \overline{\mathbf{a}}\overline{\mathbf{b}}\mathbf{cd} + \overline{\mathbf{a}}\overline{\mathbf{c}} \quad k + km = k$$

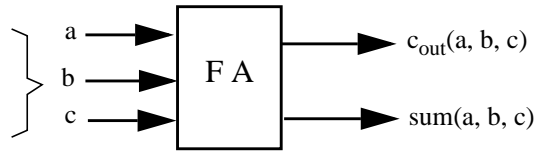
$$= \mathbf{bd} + \overline{a}\overline{b} + \overline{a}\overline{c} \quad k + km = k$$

the minimal SOP expression !

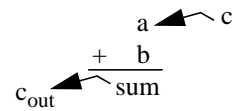
Example 5

1-bit ADDer, Full ADDer (FA)

The full adder has 3 inputs and 2 outputs. All the inputs are data inputs.



The 1-bit ADDer



We obtain the truth table from which we obtain the canonical SOP expressions :

a	b	c	$c_{out}(a, b, c)$	$sum(a, b, c)$
0	0	0	0	0
1	0	0	0	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	1	0
7	1	1	1	1

$$c_{out}(a,b,c) = \sum m(3, 5, 6, 7)$$

$$\begin{matrix} \overbrace{0\ 1\ 1}^3 & \overbrace{1\ 0\ 1}^5 & \overbrace{1\ 1\ 0}^6 & \overbrace{1\ 1\ 1}^7 \\ \bar{a}\ \bar{b}\ c & +\ a\ \bar{b}\ c & +\ a\ b\ \bar{c} & +\ a\ b\ c \end{matrix}$$

$$sum(a,b,c) = \sum m(1, 2, 4, 7)$$

$$\begin{matrix} \overbrace{0\ 0\ 1}^1 & \overbrace{0\ 1\ 0}^2 & \overbrace{1\ 0\ 0}^4 & \overbrace{1\ 1\ 1}^7 \\ \bar{a}\ \bar{b}\ c & +\ \bar{a}\ b\ \bar{c} & +\ a\ \bar{b}\ \bar{c} & +\ a\ b\ c \end{matrix}$$

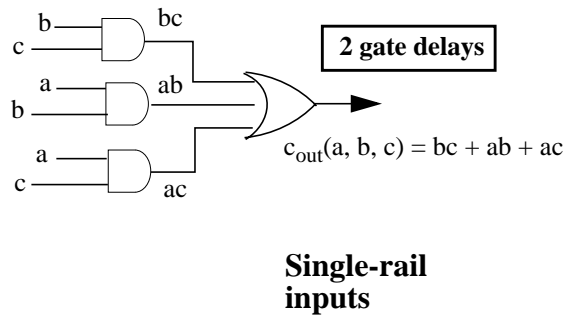
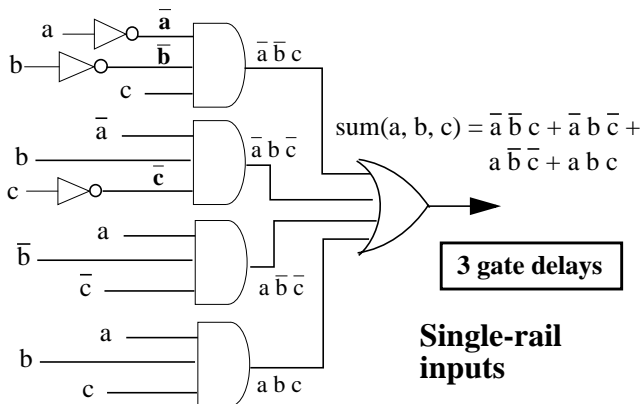
The canonical $sum(a, b, c)$ expression is also the minimal expression. It **cannot** be simplified :

$$sum(a, b, c) = \bar{a}\ \bar{b}\ c + \bar{a}\ b\ \bar{c} + a\ \bar{b}\ \bar{c} + a\ b\ c$$

$$\begin{aligned} c_{out}(a, b, c) &= \bar{a}\ b\ c + a\ \bar{b}\ c + a\ b\ \bar{c} + a\ b\ c \\ &= bc(\bar{a} + a) + a\ \bar{b}\ c + a\ b\ \bar{c} \\ &= bc + a\ \bar{b}\ c + a\ b\ \bar{c} \\ &= c(b + a\bar{b}) + a\ b\ \bar{c} \\ &= c(b + a) + a\ b\ \bar{c} \\ &= bc + ac + a\ b\ \bar{c} \\ &= b(c + a\bar{c}) + ac \\ &= b(c + a) + ac \\ &= bc + ab + ac \end{aligned}$$

$$\begin{aligned} k(m+p) &= km + kp \\ k+\bar{k} &= 1 \ \& \ k1 = k \\ k(m+p) &= km + kp \\ k + \bar{k}m &= k + m \end{aligned}$$

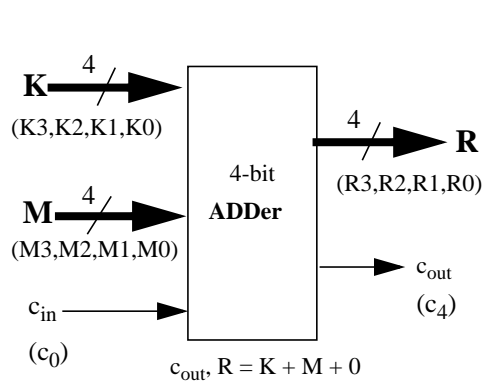
$$\begin{aligned} k(m+p) &= km + kp \\ k(m+p) &= km + kp \\ k + \bar{k}m &= k + m \\ k(m+p) &= km + kp \end{aligned}$$



Example 6

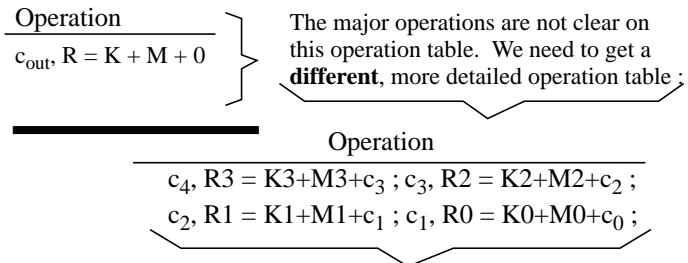
4-bit ADDer

The 4-bit adder adds two 4-bit numbers. It has 9 inputs and 5 outputs as shown below. The 1-bit input is carry_{in} and the 4-bit inputs are K and M. The outputs are a 4-bit R and a 1-bit carry_{out} :



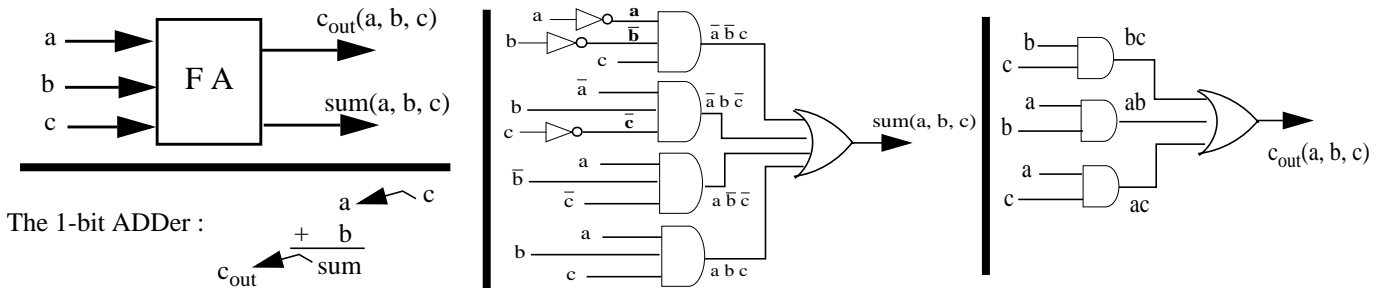
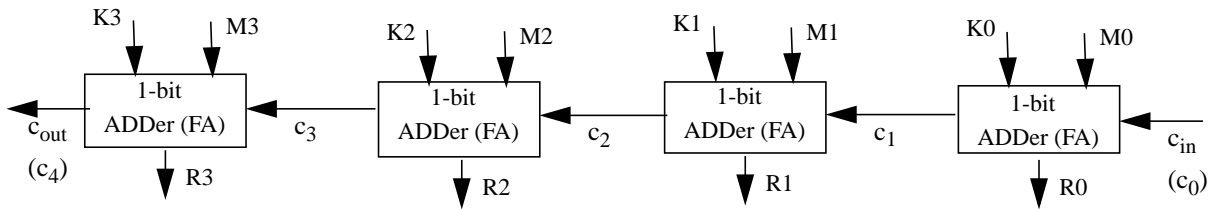
Since there are 9 inputs, we need to partition it into simpler pieces !

We have to obtain the operation table of the 4-bit ADDer :

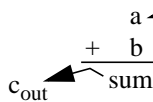


There are four identical major operations : 1-bit ADDitions !

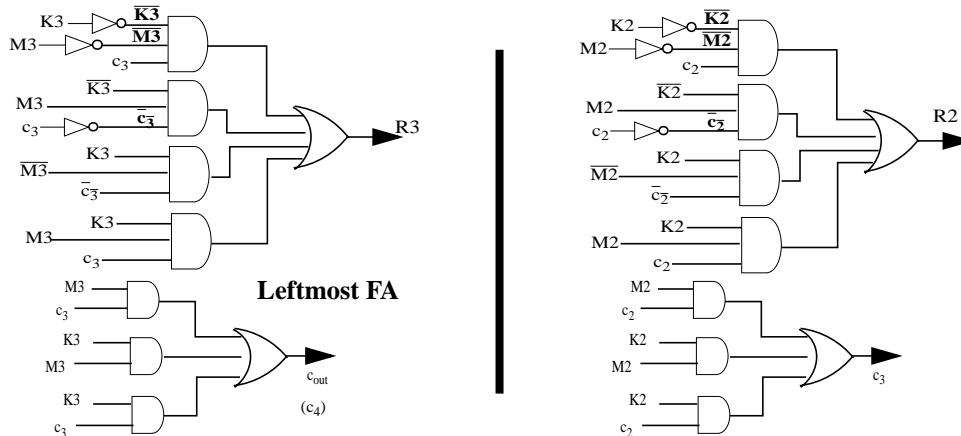
Each 1-bit ADDer is a full adder (FA). A FA has three inputs and 2 outputs ! Its design is on page 5 :



The 1-bit ADDer :



The leftmost two FAs of the 4-bit ADDer are shown below :

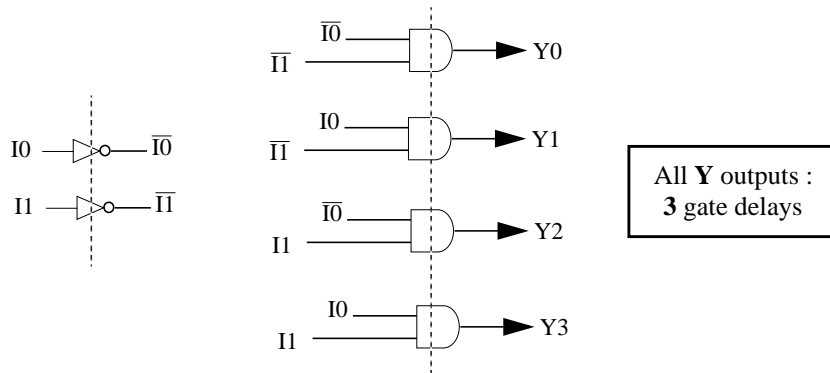
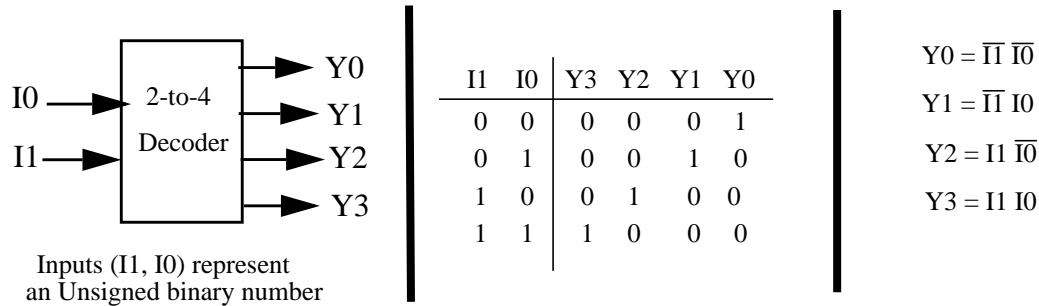


Example 7

2-to-4 Binary Decoder

The most common decoder type is the binary decoder type which has “k” data inputs and 2^k outputs. If the decoder is a 2-to-4 decoder, then “k” is 2 and so there are $2^2 = 4$ outputs.

The “k” inputs represent an Unsigned binary number. The outputs decode the unsigned number represented by the “k” inputs. For example if the inputs represent $(3)_{10}$, Output line 3 is 1 and the other outputs lines are 0. Below the development of the 2-to-4 decoder is shown.



The decoder with four outputs require 4 AND gates. Each input, I1 and I0, is connected to two AND gates. The outputs have at most two gate levels to generate the outputs. Therefore, the above decoder is fast.

Note that there are 3-to-8, 4-to-16, etc. decoders whose operation and implementation follow similarly. We will use a 3-to-8 binary decoder when we implement hardwiring in the Control Unit (Block 1) of the term project later in the semester. Today’s memory chips (DRAM, SRAM, ROM, etc.) have **very** large binary decoders.

For a k-input decoder, there are 2^k AND gates to generate the 2^k outputs. Each input is connected to half the number of AND gates. For small size decoders, this is not a major problem. But, for large decoders, it is a problem which is called “**fan-out**.”

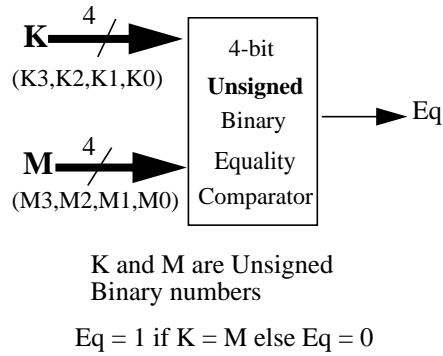
The fan-out of a line is a number which indicates how many inputs can be connected to it. If the number is exceeded, electrically, there are problems and so the circuit may not work.

It is because of this reason that the decoders of memory chips have their gate networks with more than two levels to reduce the fan-out requirement. However, with more levels, the decoder is slower, therefore, the memory chip is slower.

Example 8

4-bit Unsigned Binary Equality Comparator

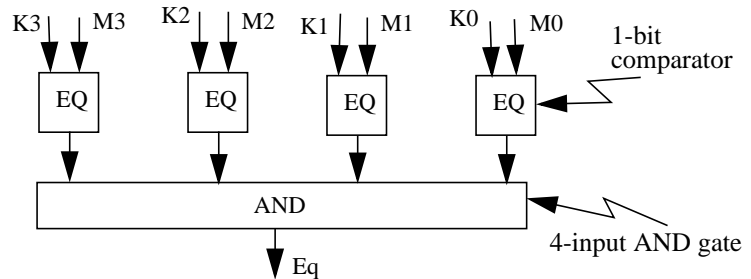
The 4-bit Unsigned Binary comparator compares two 4-bit Unsigned Binary numbers. It has 8 inputs and 1 output. All eight inputs are data inputs. The 4-bit inputs are K and M and the output is Eq. It checks if K is equal to M :



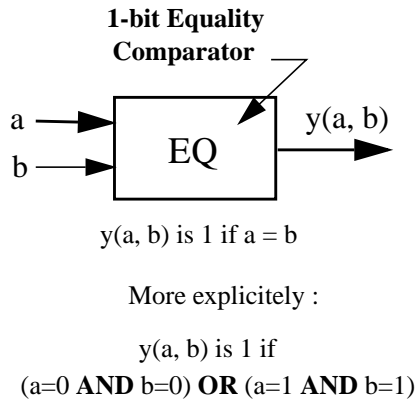
Since there are 8 inputs, we need to partition it into simpler pieces !

Output Eq is 1 only if all respective bits are equal to each other :

$$K_3=M_3 \text{ AND } K_2=M_2 \text{ AND } K_1=M_1 \text{ AND } K_0=M_0$$



Each 1-bit comparator has two inputs and 1 output ! We design it by using Switching Algebra :



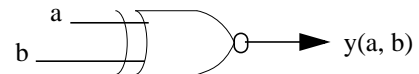
a	b	y(a, b)
0	0	1
1	0	0
2	1	0
3	1	1

$$y(a,b) = \sum m(0,3)$$

$$\overline{a} \overline{b} + a b$$

We realize this is an XNOR function :

$$y(a, b) = \overline{a \oplus b}$$



The 4-bit unsigned equality comparator that checks for equality is then as follows :

