We can also implement a function f using MUXES right from the Boolean Equation.

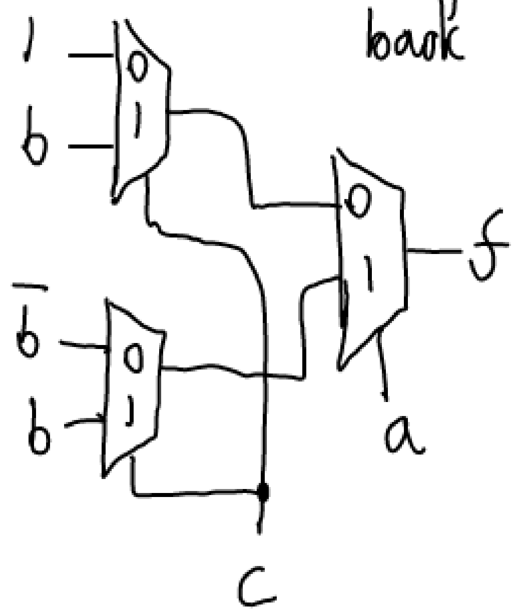
e.g., Implement f using only 2-input MUXES

$$f = abc + \bar{a}b + \bar{b}\bar{c} \quad \bar{b}\bar{c}(a+\bar{a})$$

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

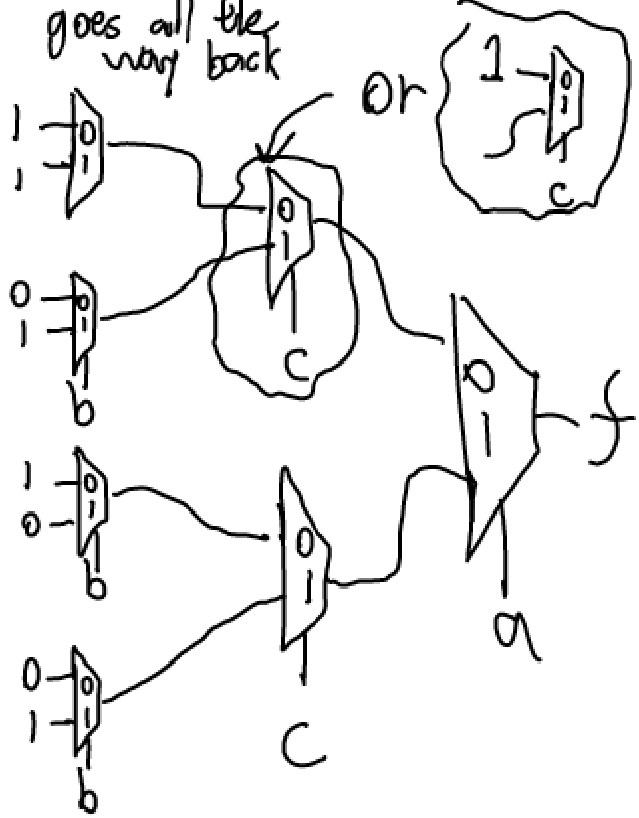
$$= \bar{a}(\bar{c}\underbrace{(\bar{b}+b)}_{1} + cb) + a(\bar{c}(\bar{b}) + c(b))$$

doesn't go all the way back



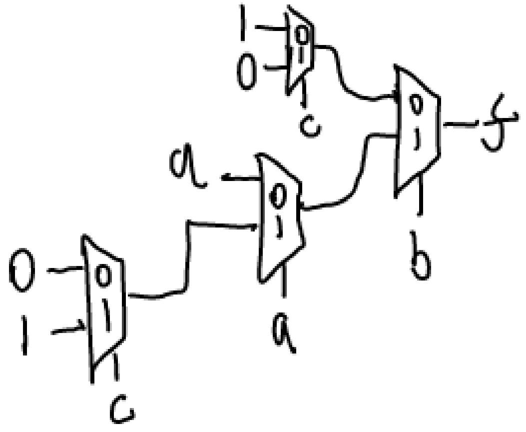
goes all the way back

or



$$f = \bar{b}(\bar{c}) + b(\bar{a} + a0)$$

$$= \bar{b}(\bar{c}(1) + c(0)) + b(\bar{a}(1) + a(0))$$



$$+ b(\bar{a}(1) + a(\bar{c}(0) + c(1)))$$

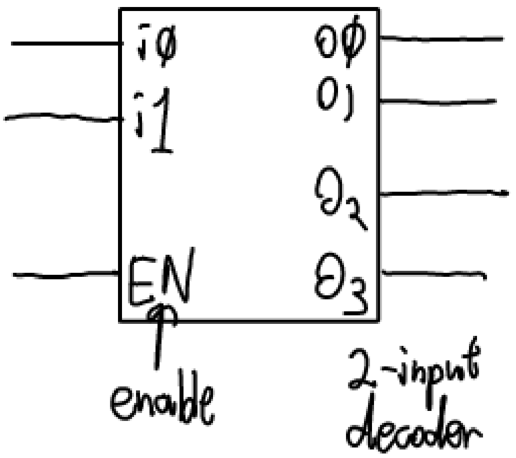
Shannon Decomposition (BDD)

Decoders

$\rightarrow 0 \ldots 2^n - 1$

* We have a binary pattern in n-bits and we want to set one output to 1 based on the input value.

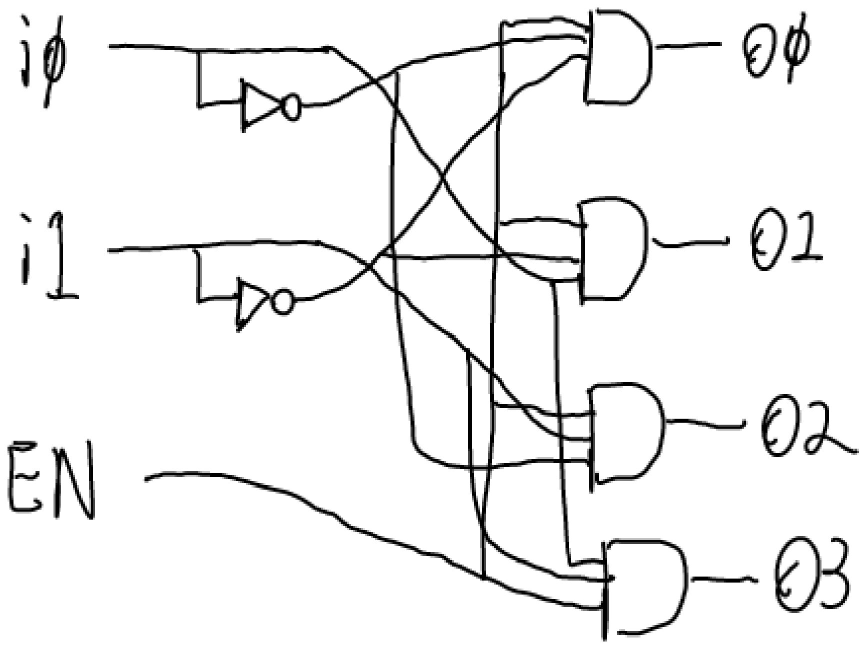
e.g.,



i1	i0	EN	O0	O1	O2	O3
x	x	0	0	0	0	0
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1

$O_0 = \overline{i1}\,\overline{i0}\,EN$

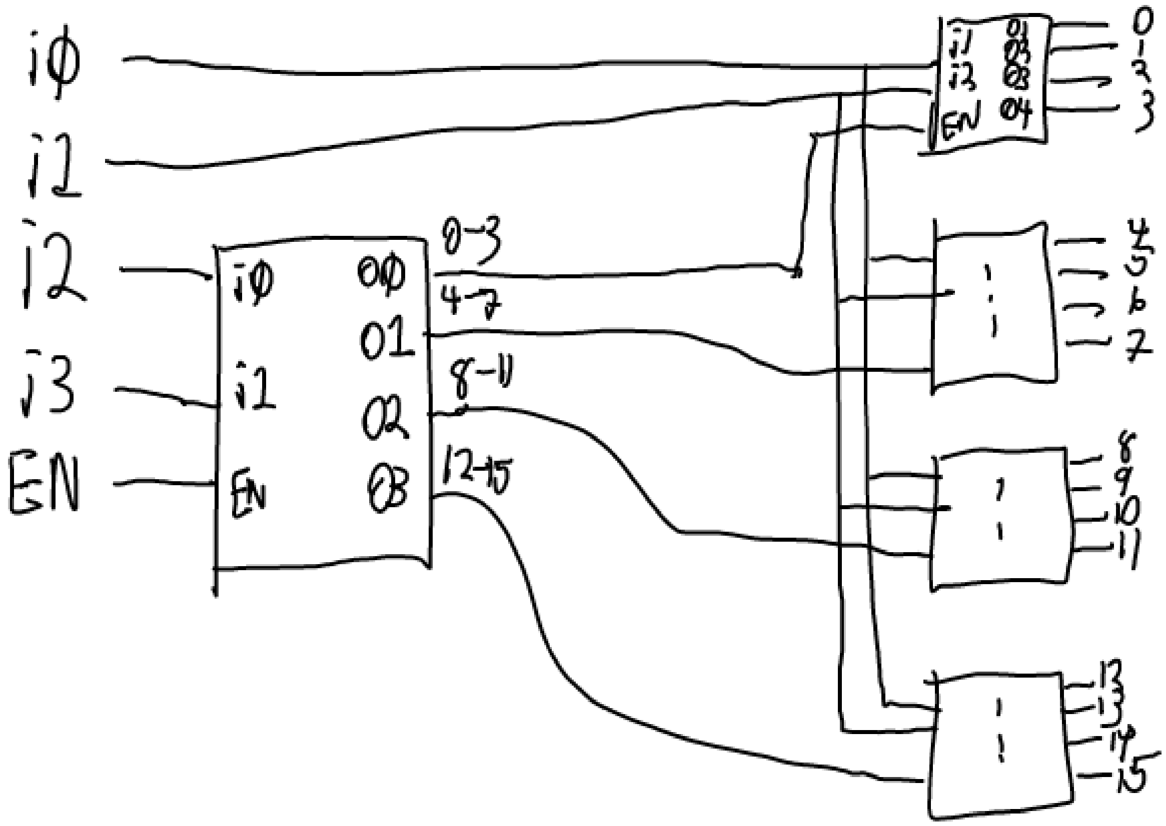
$O_2 = i1\,\overline{i0}\,EN$



Decoder Trees

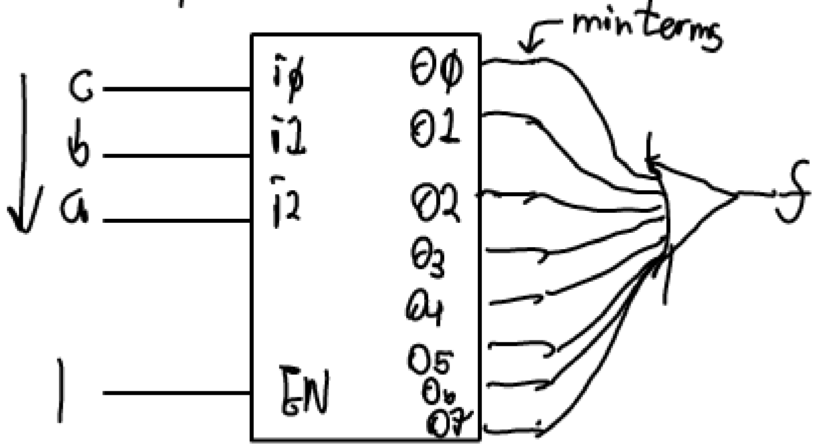
build a 4-to-16 decoder. Using 2-to-4 decoders.

i3	i2	i1 i0	
0	0	4	0-3
0	1	4	4-7
1	0	4	8-11
1	1	4	12-15



Make $f = abc + \bar{a}b + \bar{b}\bar{c}$ using a decoder.

$\Rightarrow$ 3 inputs $\Rightarrow \therefore$ 3-to-8 decoder.



a	b	c	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Encoder

* opposite to decoder
* has 2^n input lines and n outputs
* output is the binary value of the input line which is 1.

i_3	i_2	i_1	i_0	O1	O0	valid
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1

$O0 = i_1 + i_3$

$O1 = i_2 + i_3$

Problems

* What if multiple inputs are 1?
  ⇒ Ok... make it a priority encoder
  (highest #ed input is the one that matters)
  (derive equation for final exam)
* What if all zeroes?
  make another output to check for error.

i_3	i_2	i_1	i_0	valid
0	0	0	0	0