5 - combinational circuits - decoders, encoders, multiplexers

decoders

- used to implement arbitrary functions
- abstraction and modularity, as building blocks
- you can implement arbitrary functions with decoders

$n \rightarrow$ [in every output, there is only one 1 among Os] $\rightarrow 2^n$

minterm generator

S_1	S_0	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

decodes a binary number into a one of four code

$Q_0 = \overline{S_1} \cdot \overline{S_0}$
$Q_1 = \overline{S_1} \cdot S_0$
$Q_2 = S_1 \cdot \overline{S_0}$
$Q_3 = S_1 \cdot S_0$

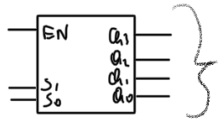
enable input used to activate or deactivate the device

$EN = 1 \Rightarrow$ decoder is active, behaves as specified
$EN = 0 \Rightarrow$ decoder is deactive, all outputs are 0.

EN	S_1	S_0	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

EN = 0, S_1 = x, S_0 = x — don't care

blocks and abstraction encapsulated decoders



2 to 4 decoder.

$Q_0 = \overline{S_1} \cdot \overline{S_0}$
$Q_1 = \overline{S_1} \cdot S_0$
$Q_2 = S_1 \cdot \overline{S_0}$
$Q_3 = S_1 \cdot S_0$

- decoder blocks provide abstraction
- simpler diagrams and hardware reuse
- like functions in programming

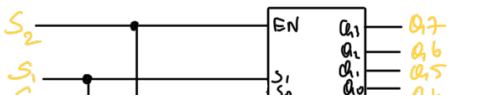
3 to 8 decoder

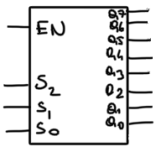
S_2	S_1	S_0	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

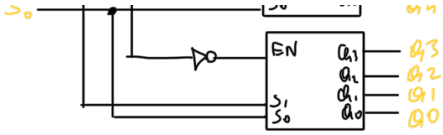
or two 2 to 4 decoders

(S_2 like enable input)

when $S_2 = 0$ outputs: $Q_0 - Q_3$
$S_2 = 1$ outputs: $Q_4 - Q_7$







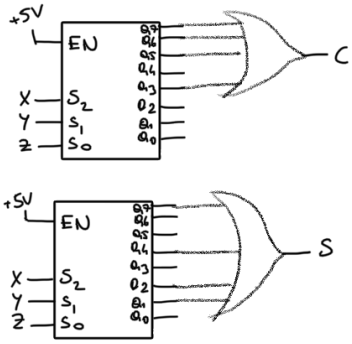
decoder-based adder

X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

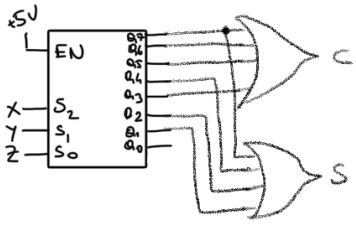
carry ↵ ↓ sum (x+y+z)

$C(x,y,z) = \sum m(3,5,6,7)$

$S(x,y,z) = \sum m(1,2,4,7)$



Or using one decoder



Variation of the standard decoder

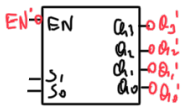
active-high decoder (normal one)

EN	S_1	S_0	Q_0	Q_1	Q_2	Q_3
0	×	×	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

$Q_3 = S_1 S_0$ $Q_2 = S_1 S_0'$ $Q_1 = S_1' S_0$ $Q_0 = S_1' S_0'$

↳ generate minterms

active-low decoder



EN'	S_1'	S_0'	Q_0'	Q_1'	Q_2'	Q_3'
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	×	×	1	1	1	1

$Q_3' = S_1' + S_0'$ $Q_2' = S_1' + S_0$ $Q_1' = S_1 + S_0'$ $Q_0' = S_1 + S_0$

↳ generate maxterms

converting $\sum m(4,5,7) = \prod M(1,2,3,6)$

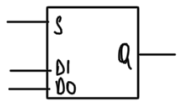
multiplexers / demultiplexers / muxes

↳ used to choose between devices
↳ also used to implement arbitrary functions

inputs =
① 2^n data input lines
② S select line (to choose one of the 2^n data input lines)

$2^n \rightarrow$ ▭ $\rightarrow 1$

Output = one of the 2^n data inputs



$Q = S' D_0 + S D_1$

2 to 1 mux

S	DI	DO	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S	Q
0	D_0
1	DI

if S = 0 → the output is D_0

S=1 → the output is D_1

4 to 1 multiplex

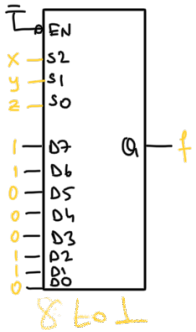
EN	S1	S0	Q
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	x	x	1

$Q = S_1' S_0' D_0 + S_1' S_0 D_1 + S_1 S_0' D_2 + S_1 S_0 D_3$ + EN (active-low one)

functions with multiplexers

ex: func
0
1
1
0
0
0
1
1

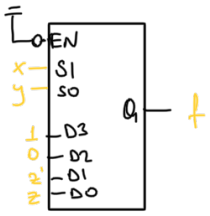
selectors x (S2)	y (S1)	z (S0)
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



A 2^n to 1 multiplexer routes one of 2^n input lines to a single output line

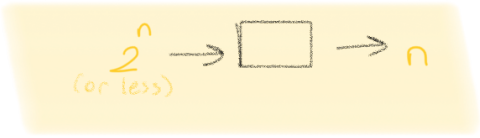
more efficient way

when
- xy = 00 → f = z (function/output)
- xy = 01 → f = z'
- xy = 10 → f = 0
- xy = 11 → f = 1



encoders

↳ the inverse operation of a decoder



inputs D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	outputs x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

ex: octal to binary

examples of all = 7-segment display

ex inputs outputs

A B C D → a b c d e f g

0	→	a+b+c+d+e+f
1	→	b+c
9	→	a+b+c+d+f+g
[0,9]		{a,b,c,d,e,f,g}
(4 bits)		
ABCD		

0	0	0	0	0	1	1	0
1	0	0	0	1	0	1	0
2	0	0	1	0	1	0	1
3	0	0	1	1	1	1	1
4	0	1	0	0	0	1	1
5	0	1	0	1	1	0	1
6	0	1	1	0	1	0	1
7	0	1	1	1	1	1	0
8	1	0	0	0	1	1	1
9	1	0	0	1	1	1	1
0	1	0	1	0	0	0	0
	1	0	1	1	0	0	0
	1	1	0	0	0	0	0
	1	1	0	1	0	0	0
	1	1	1	0	0	0	0
	1	1	1	1	0	0	0

$a = \sum m(0,2,6,7,8,9)$

$b = \sum m(0,1,2,3,4,7,8,9) \rightarrow$ active low $\sum M(5,6,10,11,12,13,14,15)$

$c = \sum m(0,1,3,4,5,6,7,8,9)$

$d = \sum m(0,2,3,5,6,8,9)$

$e = \sum m(0,2,6,8)$

$f = \sum m(0,4,5,6,8,9)$

$g = \sum m(2,3,4,5,6,8,9) \rightarrow$ active low $\sum M(0,1,7,10,11,12,13,14,15)$