Implementing logic functions by the method of decoders

Each decoder output is a function consisting of a single minterm of the inputs:

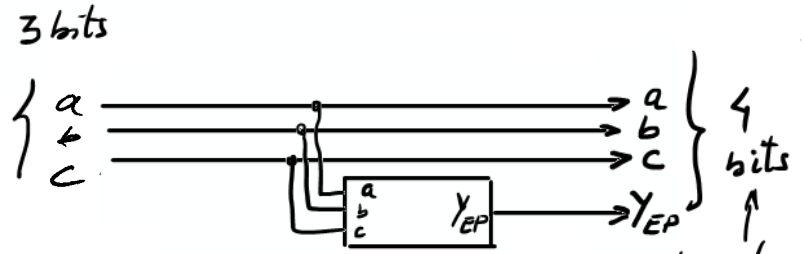
$$y_i = E \cdot m_i(x) \qquad 0 \le i \le 2^n - 1$$

In this way, it is easy to implement any logic function using a single OR

Ex: Even parity function

a b c	Y_{EP}	Y_{OP}
0 0 0	0	1
0 0 1	1	0
0 1 0	1	0
0 1 1	0	1
1 0 0	1	0
1 0 1	0	1
1 1 0	0	1
1 1 1	1	0

Odd parity function



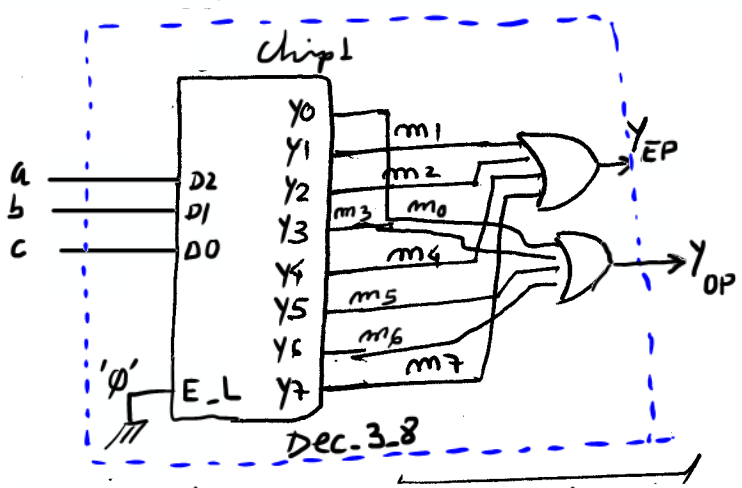
$$Y_{EP} = f(a,b,c) = \sum_3 m(1,2,4,7)$$

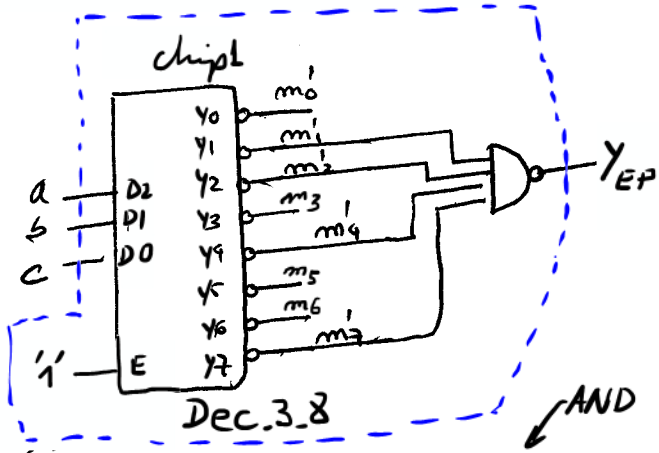
$$= m_1 + m_2 + m_4 + m_7$$

In the same way: $Y_{EP} = \left(\sum_3 m(1,2,4,7)\right)'' = \left(m_1' \cdot m_2' \cdot m_4' \cdot m_7'\right)'$

Thus, if we use a decoder with active-high outputs → OR

and if we use a decoder with active-low outputs → NAND





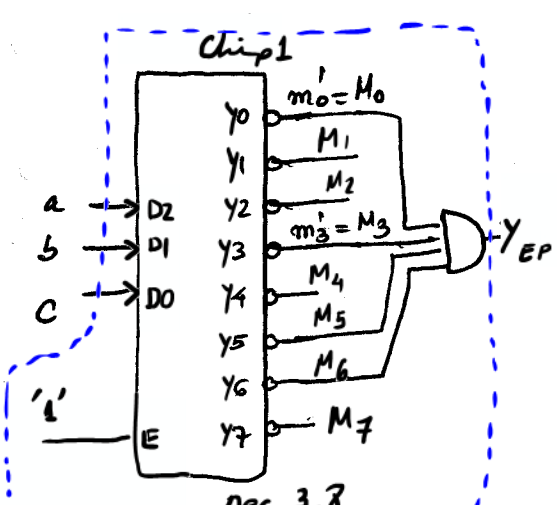
AND

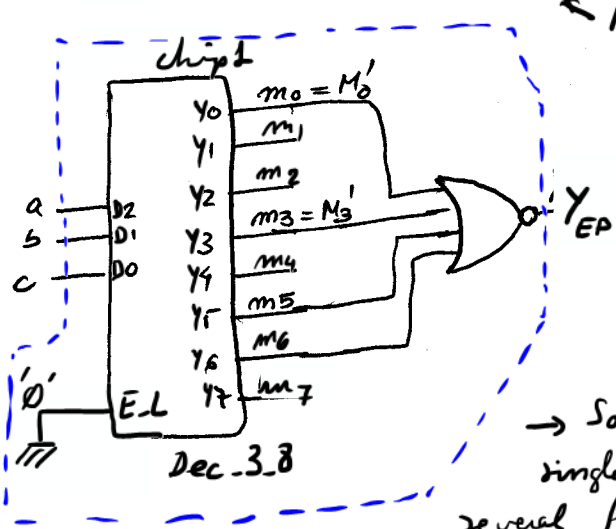
If we try to implement the function based on maxterms →

$$Y_{EP} = \prod_3 M(0,3,5,6) = M_0 \cdot M_3 \cdot M_5 \cdot M_6$$

$$Y_{EP} = (M_0' + M_3' + M_5' + M_6')'$$

NOR





→ So, with a single decoder several functions can be implemented