Multiplexer:

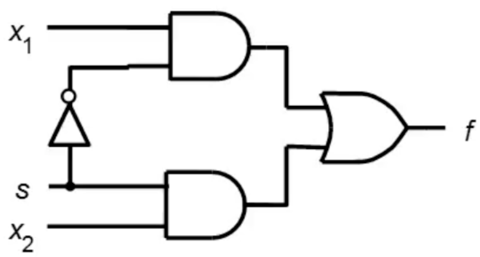
```
if s:
    return x2
else:
    return x1
```

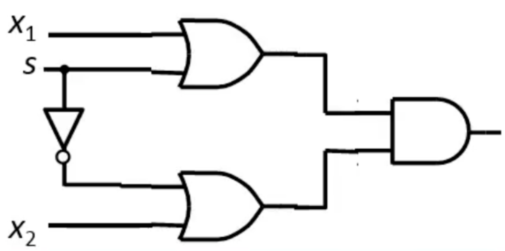
$s\ x_1\ x_2$	$f(s, x_1, x_2)$
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

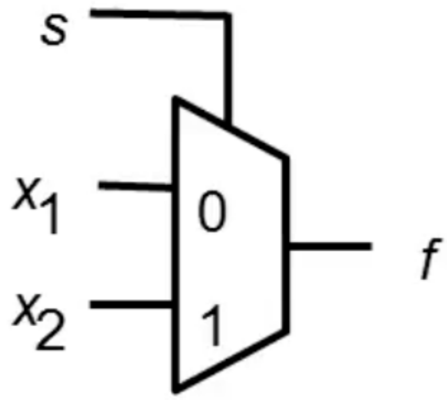
n (minterms) = n(maxterms)

$$f = \sum m(2,3,5,7) = !s\, x_1 + s\, x$$
$$= \prod M(0,1,4,6) = (s + x_1)(!s + x_2)$$

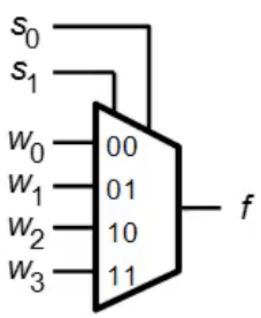
2-to-1 multiplexer:



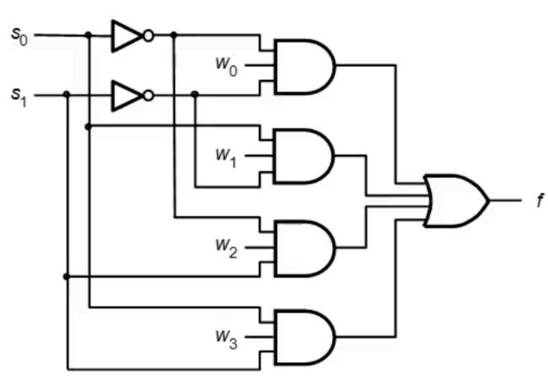


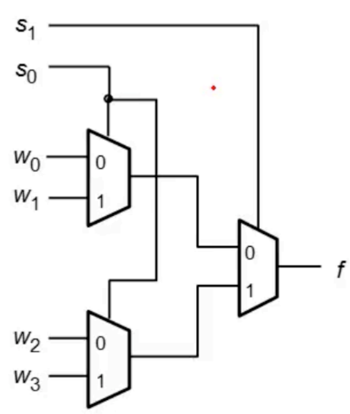


4-to-1 (s_1 , s_2)



s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

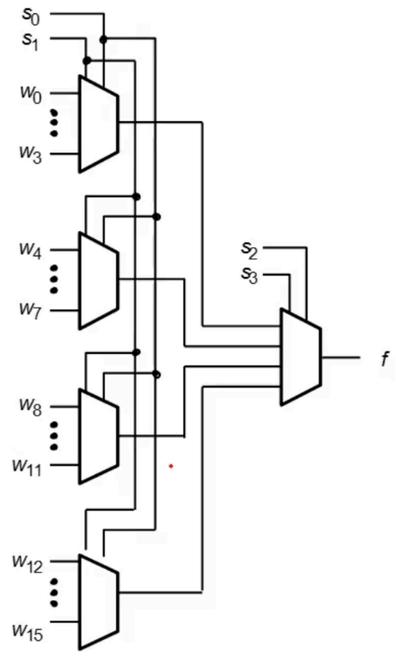




$n = 2^k$ for k in $\mathbb{N}$, n-to-1 multiplexer has:

- n inputs
- 1 output
- k selection control inputs

16-to-1



$$\begin{cases} f = x_1 \text{ if } s = 0 \\ f = x_2 \text{ if } s = 1 \end{cases}$$

s	$f(s, x_1, x_2)$
0	x_1
1	x_2

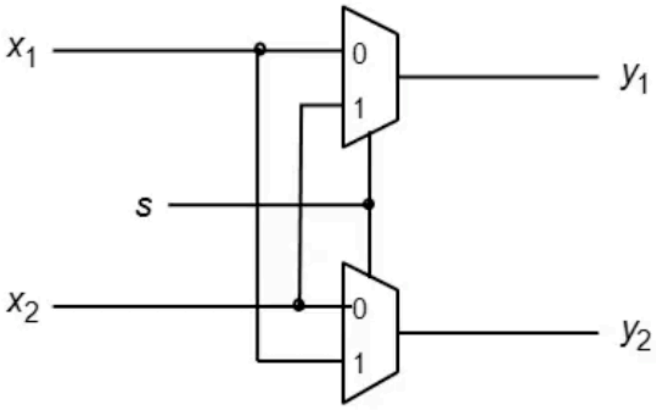
Crossbar switch

If s = 0, crossbar connects:

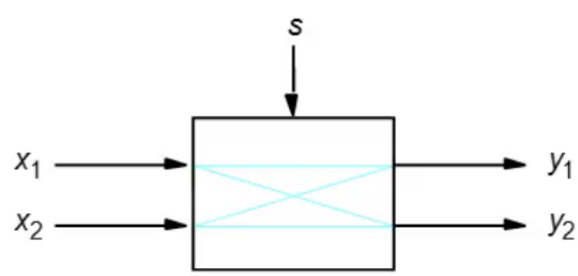
- x_1 to y_1
- x_2 to y_2

If s = 1, crossbar connects:

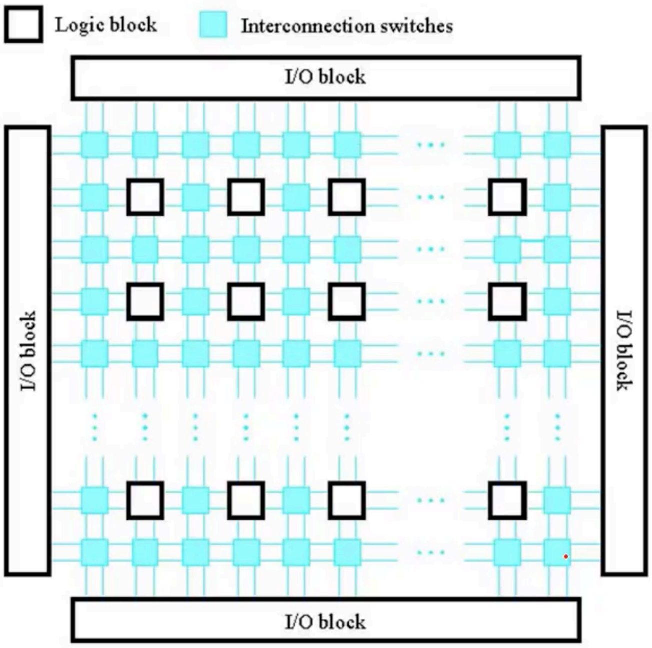
- x_1 to y_2
- x_2 to y_1



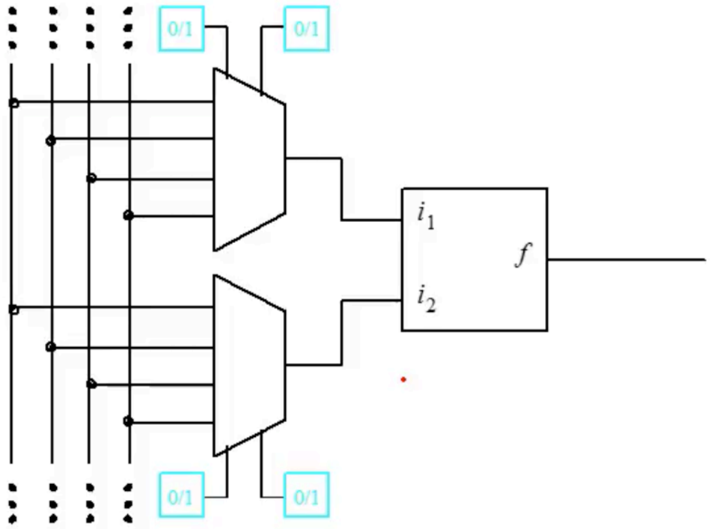
(b) Implementation using multiplexers



(a) A 2x2 crossbar switch



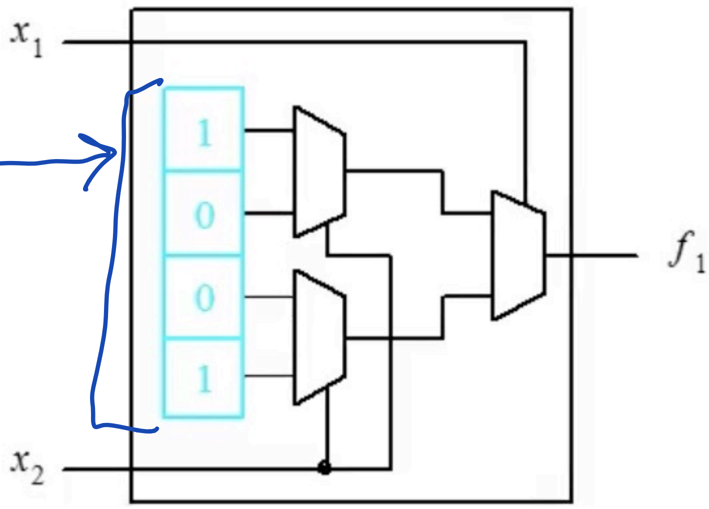
The interconnection between logic blocks can be implemented using 4-to-1 multiplexers.



Can implement any logic gate:

lookup table (LUT)

truth table



$$f(w_1, \dots, w_n) = \underbrace{!w_1 f(0, w_2, \dots, w_n)}_{\text{co factor of with respect to } !w_1} + \underbrace{w_1 f(1, w_2, \dots, w_n)}_{\dots \text{ to } w}$$

if term of f has $!w_1$ → goes to $!w_1(\ldots)$

w_1 → $w_1(\ldots)$

doesn't have w_1 → goes to both

factoring out all of the terms leads to a canonical SOP expression.