Digital Logic Design: a rigorous approach ©

Chapter 14: Selectors

Guy Even Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

May 14, 2020

Book Homepage:
http://www.eng.tau.ac.il/~guy/Even-Medina

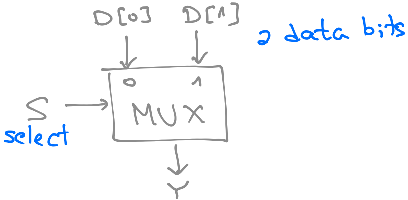
Multiplexer (MUX)

Definition

A MUX-gate is a combinational gate that has three inputs $D[0], D[1], S$ and one output Y. The functionality is defined by

$$Y = \begin{cases} D[0] & \text{if } S = 0 \\ D[1] & \text{if } S = 1. \end{cases}$$

Note that we could have used the shorter expression $Y = D[S]$ to define the functionality of a MUX-gate.



Definition

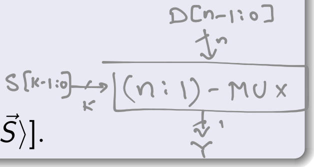
An (n:1)-MUX is a combinational circuit defined as follows:

Input: data input $D[n-1:0]$ and select input $S[k-1:0]$ where $k = \lceil \log_2 n \rceil$.

Output: $Y \in \{0,1\}$.

Functionality:

$$Y = D[\langle \vec{S} \rangle].$$



To simplify the discussion, we will assume in this chapter that n is a power of 2, namely, $n = 2^k$.

Example

3 2 1 0

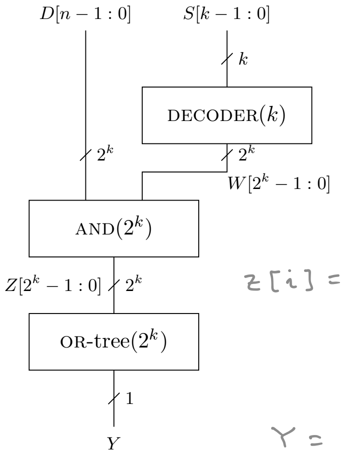
Let $n = 4$ and $D[3:0] = 0101$. If $S[1:0] = 00$, then $Y = D[0] = 1$. If $S[1:0] = 01$, then $Y = D[1] = 0$.

Implementation

We describe two implementations of (n:1)-MUX.

- translate the number $\langle \vec{S} \rangle$ to 1-out-of-n representation (using a decoder).
- tree based.

decoder based (n:1)-MUX



$w[i] = 1 \iff \langle s \rangle = i$

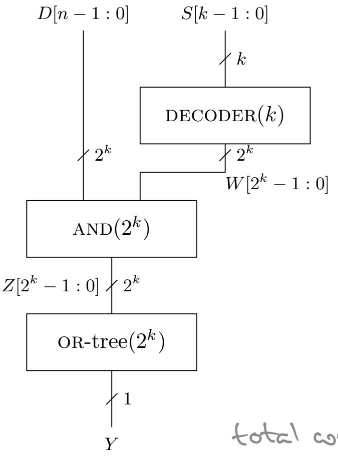
$$z[i] = \begin{cases} 0 & \text{if } i \neq \langle S \rangle \\ D[i] & \text{if } i = \langle S \rangle \end{cases}$$

$Y = D[\langle S \rangle]$

Claim

The (n:1)-MUX design is correct.

decoder based (n:1)-MUX - cost



cost $= \Theta(2^k)$

cost $= \Theta(2^k)$

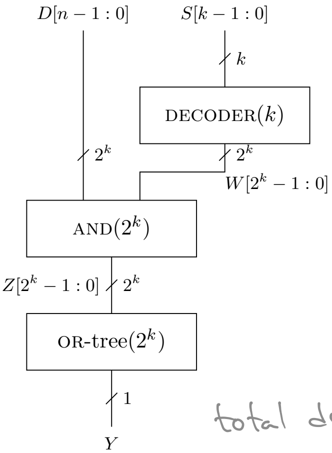
cost $= \Theta(2^k)$

total cost $= \Theta(2^k) = \Theta(n)$

Claim

The cost of the (n:1)-MUX design is $\Theta(n)$.

decoder based (n:1)-MUX - delay



delay = $\Theta(k)$

delay = $\Theta(1)$

delay = $\Theta(k)$

total delay = $\Theta(k) = \Theta(\log n)$

Claim

The delay of the (n:1)-MUX design is $\Theta(\log n)$.

(n:1)-MUX - lower bounds

Claim

The cone of the Boolean function implemented by a $(n:1)$-MUX circuit contains at least n elements.

Consider combinational circuits with gates of constant fan-in.

Corollary

The cost of the (n:1)-MUX design is asymptotically optimal.

Corollary

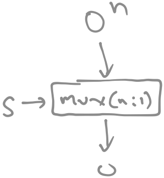
The delay of the (n:1)-MUX design is asymptotically optimal.

$$|\mathrm{cone}(\mathrm{mux})| \geq n$$

<u>proof</u>: fix $i \in \{0, \dots, n-1\}$.

let $\langle S \rangle = i$

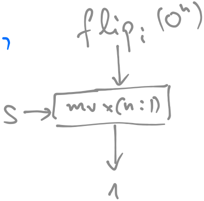
consider $D[n-1:0] = 0^n$



then, output $Y = 0$

but for $\mathrm{flip}_i(D[n-1:0])$,

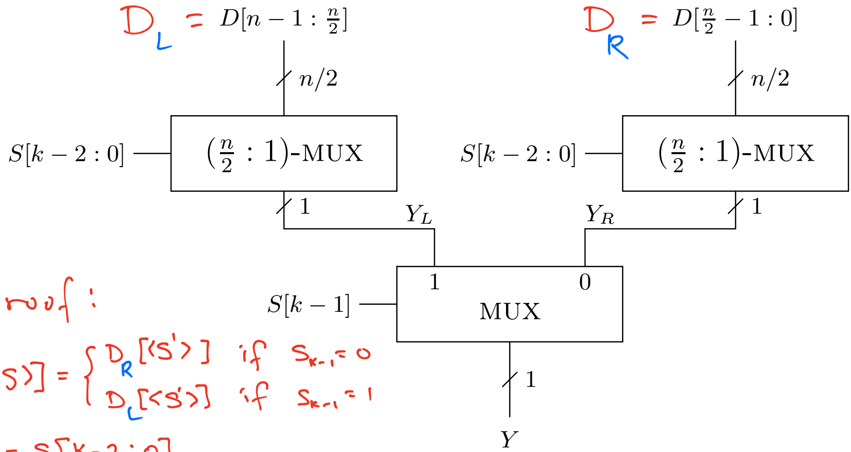
output $Y = 1$.



$\boxtimes$

tree based (n:1)-MUX

(recursive design)



proof:

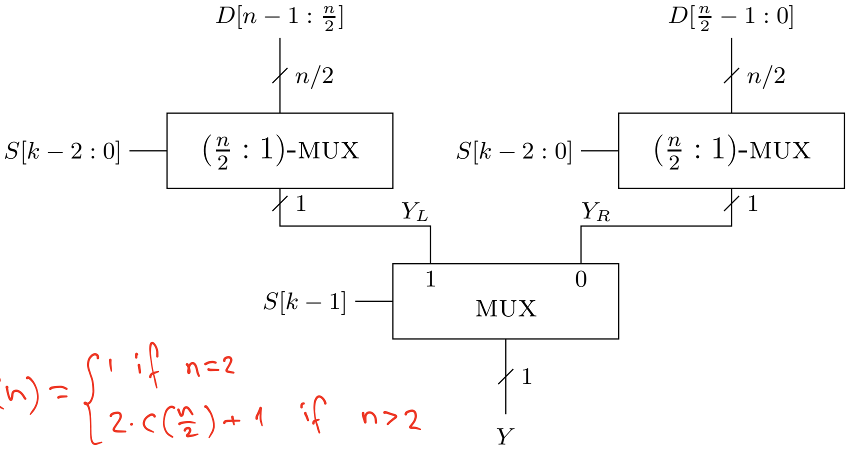
$$D[\langle S\rangle] = \begin{cases} D_R[\langle S'\rangle] & \text{if } S_{k-1} = 0 \\ D_L[\langle S'\rangle] & \text{if } S_{k-1} = 1 \end{cases}$$

$S' = S[k-2:0]$,

Claim

The (n:1)-MUX design is correct.

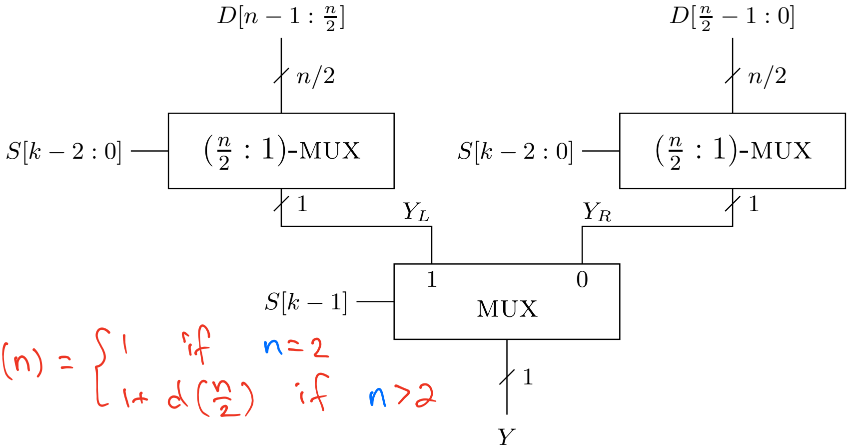
tree based (n:1)-MUX - cost



$$c(n) = \begin{cases} 1 & \text{if } n=2 \\ 2\cdot c(\frac{n}{2})+1 & \text{if } n>2 \end{cases}$$

Claim

The cost of the (n:1)-MUX design is $\Theta(n)$.



$$d(n) = \begin{cases} 1 & \text{if } n=2 \\ 1+d\left(\frac{n}{2}\right) & \text{if } n>2 \end{cases}$$

Claim

The delay of the (n:1)-MUX design is $\Theta(\log n)$.

Comparison

- Both implementations are asymptotically optimal with respect to cost and delay.
- The cost/delay table suggests that the tree-like implementation is cheaper and faster.
- Fast and cheap implementations of MUX-gates in CMOS technology (called "pass transistors") do not restore the signals well. This means that long paths consisting only of such MUX-gates are not allowed (must interleave with invertors to restore the signals).
- What about physical layout? Which design has a smaller "drawing"? (beyond the scope of this course)
- Conclusion: our simplified model cannot be used to deduce conclusively which multiplexer design is better.

both are asymp. optimal.