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.

n-bit selector

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, $n = 2^k$.
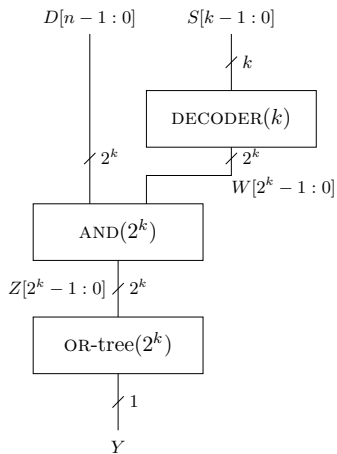
Example

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

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

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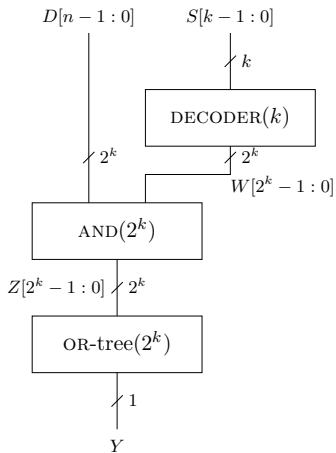- tree based.

decoder based (n:1)-MUX

$D[n-1:0]$ $S[k-1:0]$

k

DECODER(k)

2^k 2^k

$W[2^k-1:0]$

AND(2^k)

$Z[2^k-1:0]$ 2^k

OR-tree(2^k)

1

Y

Claim

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

decoder based (n:1)-MUX - cost

$D[n-1:0]$ $S[k-1:0]$

k

DECODER(k)

2^k 2^k

$W[2^k-1:0]$

AND(2^k)

$Z[2^k-1:0]$ 2^k

OR-tree(2^k)

1

Y

Claim

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

$D[n-1:0]$ $S[k-1:0]$

k

DECODER(k)

2^k 2^k

$W[2^k-1:0]$

AND(2^k)

$Z[2^k-1:0]$ 2^k

OR-tree(2^k)

1

Y

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.

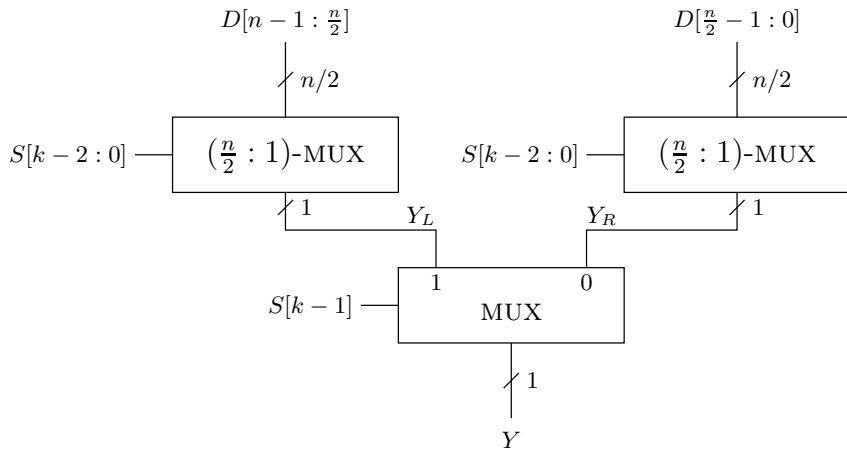Consider combinational circuits with gates of constant fan-in.

Corollary

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

Corollary

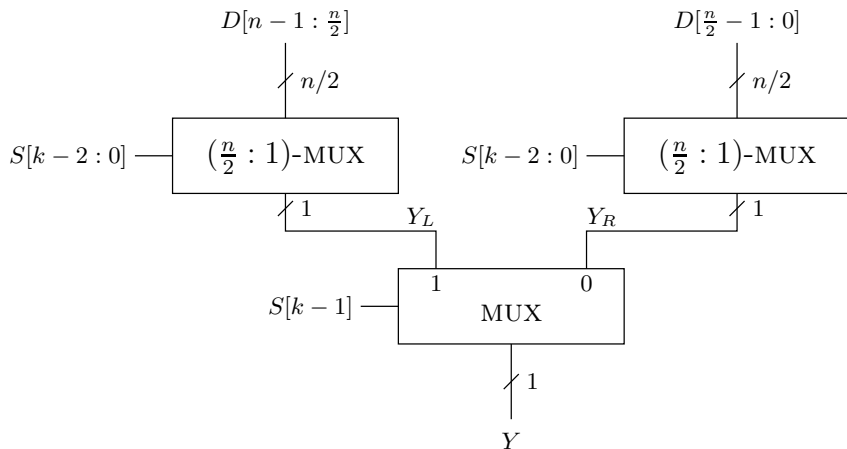*The delay of the (n:1)-MUX design is asymptotically optimal.*
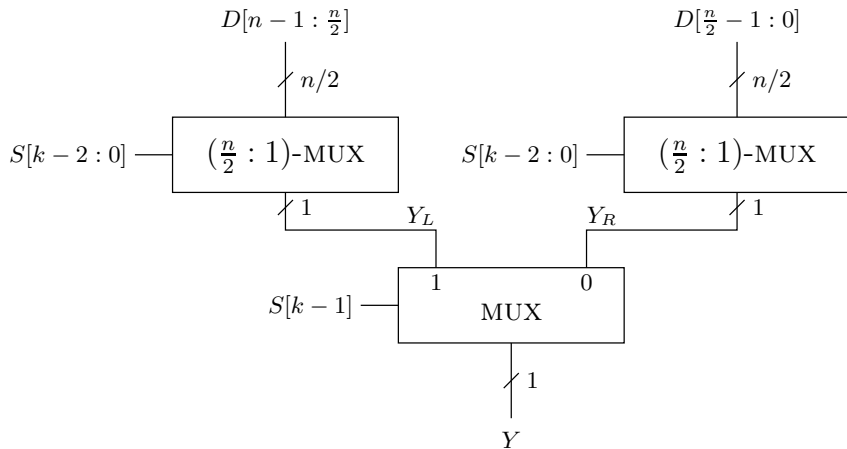
Claim

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

The diagram shows:

$D[n-1:\frac{n}{2}]$ with $n/2$ into $\left(\frac{n}{2}:1\right)$-MUX, with input $S[k-2:0]$, output 1 labeled Y_L.

$D[\frac{n}{2}-1:0]$ with $n/2$ into $\left(\frac{n}{2}:1\right)$-MUX, with input $S[k-2:0]$, output 1 labeled Y_R.

Y_L connects to input 1 and Y_R connects to input 0 of MUX, with select $S[k-1]$, output 1 labeled Y.

Claim

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

$D[n - 1 : \frac{n}{2}]$

$n/2$

$S[k - 2 : 0]$ —— $\left(\frac{n}{2} : 1\right)$-MUX

1 Y_L

$D[\frac{n}{2} - 1 : 0]$

$n/2$

$S[k - 2 : 0]$ —— $\left(\frac{n}{2} : 1\right)$-MUX

Y_R 1

1 0

$S[k - 1]$ —— MUX

1

Y

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.