Digital Logic Design: a rigorous approach ©

Chapter 18: Memory Modules

Guy Even   Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

January 2, 2014

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

The Zero Delay Model

1. Transitions of all signals are instantaneous.
2. Combinational gates: $t_{pd} = t_{cont} = 0$.
3. Flip-flops satisfy:

$$\begin{aligned} t_{su} &= t_{i+1} - t_i, \\ t_{hold} &= t_{cont} = t_{pd} = 0. \end{aligned}$$

4. Simplified model for specifying and simulating the functionality of circuits with flip-flops.

The Zero Delay Model - cont

1. The clock period, in the delay model, equals 1.
2. $t_{i+1} - t_i = 1$, for every i.
3. the duration of the ith clock cycle is the interval $[t_i, t_{i+1}) = [i, i+1)$.
4. All transitions are instantaneous, so we may assume that each signal is stable during each clock cycle.
5. Let X_i denote the digital value of the signal X during the i'th clock cycle.

The Zero Delay Model - functionality of a FF

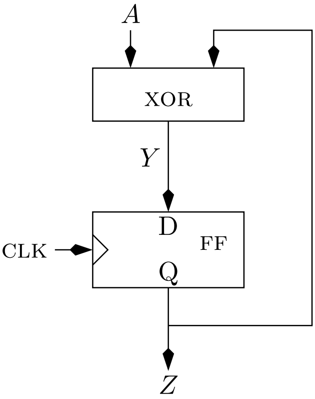
1. The functionality of a flip-flop is specified as follows:

   $$Q(t) = D(t-1).$$

   Since each signal is stable during each clock cycle, we could also write $Q_i = D_{i-1}$.

2. meaning:
   - The critical segment C_i equals $[t_{i-1}, t_i)$.
   - The value of $D(t)$ is stable during the critical segment $[t_{i-1}, t_i)$.
   - This value, denoted by D_{i-1}, is sampled by the flip-flop during the clock cycle $(i-1)$.
   - In the next clock cycle $[t_i, t_{i+1})$, the flip-flop's output $Q(t)$ equals the value of the input sampled during the previous cycle.

Example: Sequential XOR



i	A_i	Y_i	Z_i
0	0	0	0
1	0	0	0
2	1	1	0
3	0	1	1
4	0	1	1
5	1	0	1
6	0	0	0
7	1	1	0
8	0	1	1

Registers

A term register is used to define a memory device that stores a bit or more. There are two main types of register depending on how their contents are loaded.

1. Parallel Load Register
2. Shift Register (also called a serial load register)

Parallel Load Register - specification

Definition

An n-bit *parallel load register* is specified as follows.

Inputs:
- $D[n-1:0](t)$,
- $\textsc{ce}(t)$, and
- a clock $\textsc{clk}$.

Output: $Q[n-1:0](t)$.

Functionality:

$$Q[n-1:0](t+1) = \begin{cases} D[n-1:0](t) & \text{if } \textsc{ce}(t) = 1 \\ Q[n-1:0](t) & \text{if } \textsc{ce}(t) = 0. \end{cases}$$

An n-bit parallel load register is simply built from n clock enabled flip-flops.

Parallel Load Register - design

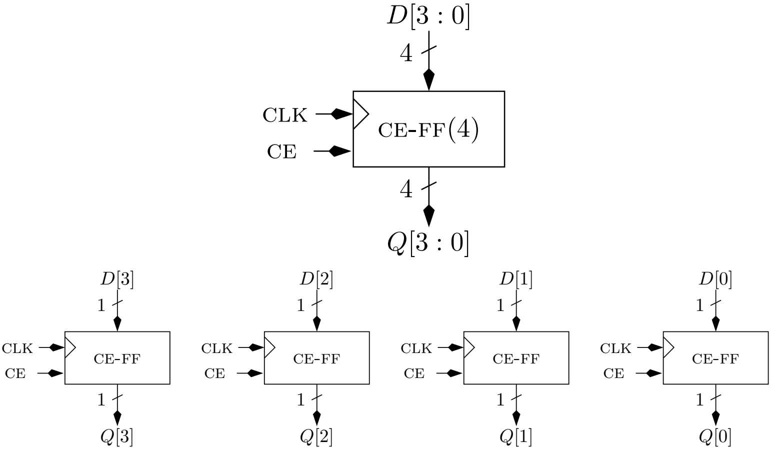
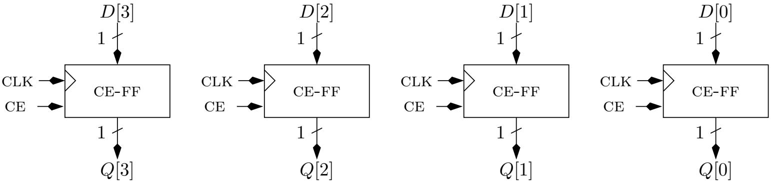


Figure: A 4-bit parallel load register

Parallel Load Register - simulation



i	$D[3:0]$	CE	$Q[3:0]$
0	1010	1	0000
1	0101	1	1010
2	1100	0	0101
3	1100	1	0101
4	0011	1	1100

Shift Register - definition

Definition

A *shift register* of n bits is defined as follows.

Inputs: $D[0](t)$ and a clock $\textsc{clk}$.

Output: $Q[n-1](t)$.

Functionality: $Q[n-1](t+n) = D[0](t)$.

Shift Register - design

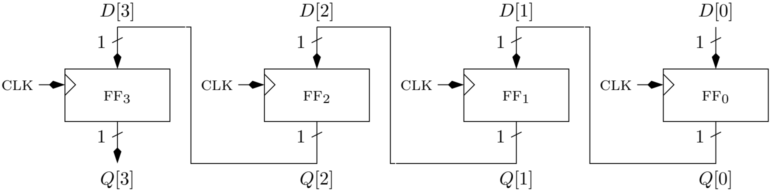
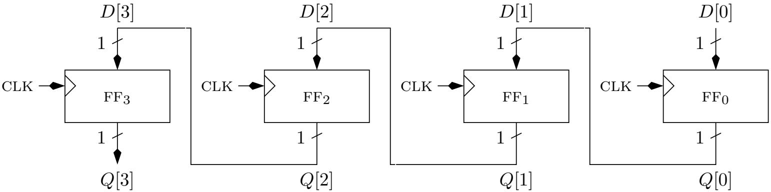


Figure: A 4-bit shift register. Functionality: $Q[3](t+4) = D[0](t)$

Shift Registers - simulation



i	$D[0]$	$Q[3:0]$
0	1	0000
1	1	0001
2	1	0011
3	0	0111
4	1	1110

Random Access Memory (RAM)

1. Array of memory cells.
2. Each memory cell stores a single bit.
3. in each cycle, a single memory cell is accessed.
4. Two operations are supported: read and write.
    - read operation: the contents of the accessed memory is output.
    - write operation: a new value is stored in the accessed memory.
5. The number of memory cells is denoted by 2^n.
6. Each cell has a distinct address between 0 and $2^n - 1$.
7. The cell to be accessed is specified by an n-bit string called *Address*.
8. The array of memory cells is denoted by $M[2^n - 1 : 0]$. Let $M[i](t)$ denote the value stored in the ith entry of the array M during clock cycle t.

RAM - definition

Definition

A RAM(2^n) is specified as follows.

Inputs: $Address[n-1:0](t) \in \{0,1\}^n$,$D_{\text{in}}(t) \in \{0,1\}$, $R/\overline{W}(t) \in \{0,1\}$ and a clock CLK.

Output: $D_{\text{out}}(t) \in \{0,1\}$.

Functionality :

1. data: array $M[2^n-1:0]$ of bits.
2. initialize: $\forall i : M[i] \leftarrow 0$.
3. For $t = 0$ to ∞ do
    1. $D_{\text{out}}(t) = M[\langle Address\rangle](t)$.
    2. For all $i \neq \langle Address\rangle$: $M[i](t+1) \leftarrow M[i](t)$.
    3. $$M[\langle Address\rangle](t+1) \leftarrow \begin{cases} D_{\text{in}}(t) & \text{if } R/\overline{W}(t) = 0 \\ M[\langle Address\rangle](t) & \text{else.} \end{cases}$$

RAM - schematic

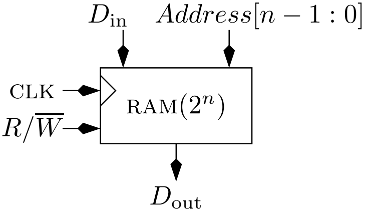
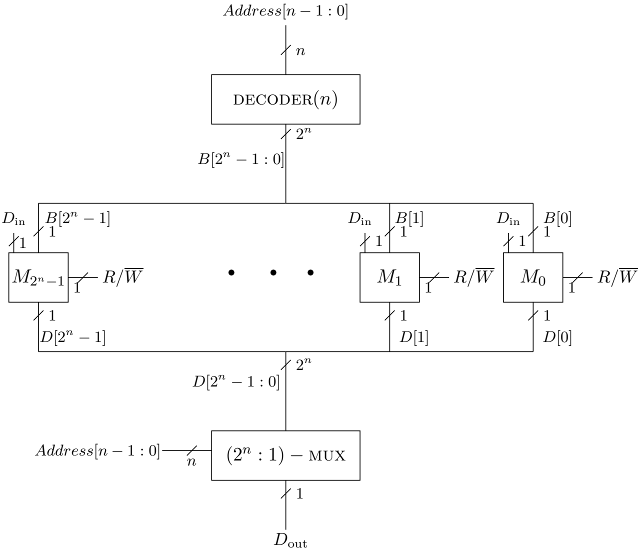


Figure: A schematic of a RAM(2^n).

RAM -design



$Address[n-1:0]$
n
$\text{DECODER}(n)$
2^n
$B[2^n-1:0]$
D_{in}
$B[2^n-1]$
1
1
M_{2^n-1}
1
$R/\overline{W}$
1
$D[2^n-1]$
D_{in}
$B[1]$
1
1
M_1
1
$R/\overline{W}$
1
$D[1]$
D_{in}
$B[0]$
1
1
M_0
1
$R/\overline{W}$
1
$D[0]$
2^n
$D[2^n-1:0]$
$Address[n-1:0]$
n
$(2^n:1)-\text{MUX}$
1
D_{out}

Memory Cell - specification

Definition

A single bit *memory cell* is defined as follows.

Inputs: $D_{\text{in}}(t)$, $R/\overline{W}(t)$, $sel(t)$, and a clock $\textsc{clk}$.

Output: $D_{\text{out}}(t)$.

Functionality:

Let $S(t) \in \{0,1\}$ denote the state of memory cell in cycle t. Assume that the state is initialized to be $S(0) = 0$. The functionality is defined according to the following cases.

1. $S(t) \leftarrow \begin{cases} D_{\text{in}}(t) & \text{if } sel(t) = 1 \text{ and } R/\overline{W}(t) = 0 \\ S(t-1) & \text{otherwise.} \end{cases}$
2. $D_{\text{out}}(t) \leftarrow S(t-1)$.

Memory Cell - design

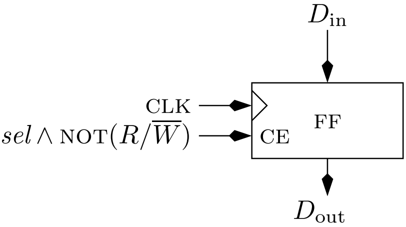


Figure: An implementation of a memory cell.

Read-Only Memory (ROM)

- The module called Read-Only Memory (ROM) is similar to a RAM, except that write operations are not supported.
- This means that the contents stored in each memory cell are preset and fixed.
- ROMs are used to store information that should not be changed.
- For example, the ROM stores the program that is executed when the computer is turned on.

ROM - definition/design

Definition

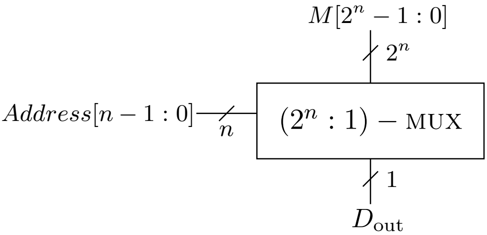
A ROM(2^n) that implements a Boolean function $M : [0..2^n - 1] \to \{0,1\}$ is defined as follows.

Inputs: $Address[n-1:0](t)$.

Output: $D_{\text{out}}(t)$.

Functionality :

$$D_{\text{out}} = M[\langle Address \rangle] \ .$$


$M[2^n - 1 : 0]$
2^n
$Address[n-1:0]$
n
$(2^n : 1) - \text{MUX}$
1
D_{out}