Memory Elements (II)
Overview

Memory Elements (II)

• registers
• register files
The Big Picture
(Clocked) Sequential Circuit
Write Enabling

- This flip-flop writes the data each time the clock rises.
- Most of the time in a computer, we only write a few flip-flops.
- How to make sure that a flip-flop does not change?
1-bit Register

- Same functionality as a flip-flop when nothing happens when the write signal is 0.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Q(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>0</td>
<td>Q(t)</td>
</tr>
</tbody>
</table>
1-bit Register

- Gating the clock
1-bit Register

- **Gating the clock**
  - Similar timing issues as latches
  - The write signal should be stable

- **Basic rule in CS-031**
  - Do not gate the clock
1-bit Register

<table>
<thead>
<tr>
<th>D</th>
<th>W</th>
<th>Q(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>0</td>
<td>Q(t)</td>
</tr>
</tbody>
</table>
2-bit Register

The register is written

- on the clock edge
- whenever the write signal is enabled
Register Files

A register file is a set of registers

Construct a 4 register file with 8-bit registers

- size of address?  
  2 bits addressing 4 registers
- size of input data?  
  8-bits to write to the 8-bit registers
- size of output data?  
  8-bits, also from the 8-bit registers
1-bit Register File

We want each register storing one bit

Proceed in two parts
  • Read Part
  • Write Part
1-bit Register File

We want each register storing one bit

Proceed in two parts

• Read Part
• Write Part
Read Part

Diagram showing a multiplexer (MUX) with four inputs labeled 'D' and one input labeled 'address'. The outputs of the 'D' inputs are connected to the inputs of the MUX, while the 'address' input selects which 'D' input is connected to the output of the MUX.
1-bit Register File

We want each register storing one bit

Proceed in two parts

• Read Part
• Write Part
Write Part

data  clock  address  write

decoder 3 2 1 0

Diagram showing connections between data, clock, address, and write signals.
The Whole Thing

data  clock  address  write

decoder
3 2 1 0

multiplexer

output
8-bit Register File

We want each register storing 8 bits

Proceed in two parts

• Read Part
• Write Part
8-bit Register File with 4 Registers
### Motivation

- Many devices need to communicate with each other
- Many wires to connect them all
- A bus is a way to share the wires

### Problem to solve

- Making sure that no two devices write on the bus at the same time
Tri-state

Basic Idea: three-states

• 1, 0, high impedance

Behavior

• when enable is asserted, the output is the input
• when enable is deasserted, the output has a high impedance value
• a high impedance value means that the output can be driven by some other device
High Impedance

When an output is in the high impedance state, it is as though there were no connection

- Think of a very high, essentially infinite, resistance between the power supply, ground lines and the output
- Think of a water pipe
  1. Value 1 is 10 gallons/sec (high)
  2. Value 0 is 2 gallons/sec (low)
  3. High impedance is 0 gallons/sec (off)
At any time

- only one of the tri-states can be enabled
Control Bits

- Output bit: controls whether the data should be driven on Dout; useful to connect to a bus
- Write bit: controls whether the input should be written at the address
RAM Implementation

Too large to build a multiplexer
  • use tri-states instead
RAM Implementation

Basic Idea

- Integrate the tri-state in the flip-flop so that it can be connected to a shared wire
- One shared wire per output
- One Decoder

Main difference from a register file

- No multiplexer
Multiplexer & Tristates

Can I build a multiplexer with tristates

- Integrate the tri-state in the flip-flop so that it can be connected to a shared wire
- One shared wire per output
- One Decoder

In practice

use multiplexers unless for RAM or for connecting to a bus
The Good News

We have everything we need to build a computer now!