Memory

Philipp Koehn

14 February 2018
D-Type Level-Triggered Latch

DATA → AND → NOR → Q

CLOCK → AND → NOR → Q

Q → NOT → AND → NOR → Q

Q → NOT → AND → NOR → Q
Operations

• Circuit latches on one bit of memory and keeps it around

• Truth table

<table>
<thead>
<tr>
<th>Data-In</th>
<th>Write</th>
<th>Data-Out</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>Data</td>
</tr>
</tbody>
</table>

• Can write 1 bit and read content
multi-bit storage
1 Bit Memory

DATA IN

WRITE

AND

NOR

AND

NOR

DATA OUT
8 Bit Memory
Output Selector

- 8 Bit Latch contains 8 bits
- Now: only read 1 bit at a time
- Select the bit with an address
- Input: address
- Output: bit value
Output Selector

WRITE

DATA IN

8-Bit Selector

ADDRESS

OUT
### Output Selector

- **Truth table**

<table>
<thead>
<tr>
<th>Address</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2 A1 A0</td>
<td>OUT</td>
</tr>
<tr>
<td>0 0 0</td>
<td>D0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>D1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>D2</td>
</tr>
<tr>
<td>0 1 1</td>
<td>D3</td>
</tr>
<tr>
<td>1 0 0</td>
<td>D4</td>
</tr>
<tr>
<td>1 0 1</td>
<td>D5</td>
</tr>
<tr>
<td>1 1 0</td>
<td>D6</td>
</tr>
<tr>
<td>1 1 1</td>
<td>D7</td>
</tr>
</tbody>
</table>

- **What Boolean operation returns the correct value for address 000?**

\[(\text{NOT A2}) \land (\text{NOT A1}) \land (\text{NOT A0}) \land \text{D0}\]
Output Selector

- Full Boolean formula

\[
\begin{align*}
( \text{NOT } A2) \text{ AND } (\text{NOT } A1) \text{ AND } (\text{NOT } A0) \text{ AND } D0 & \text{ OR} \\
( \text{NOT } A2) \text{ AND } (\text{NOT } A1) \text{ AND } A0 \text{ AND } D1 & \text{ OR} \\
( \text{NOT } A2) \text{ AND } A1 \text{ AND } (\text{NOT } A0) \text{ AND } D2 & \text{ OR} \\
( \text{NOT } A2) \text{ AND } A1 \text{ AND } A0 \text{ AND } D3 & \text{ OR} \\
A2 \text{ AND } (\text{NOT } A1) \text{ AND } (\text{NOT } A0) \text{ AND } D4 & \text{ OR} \\
A2 \text{ AND } (\text{NOT } A1) \text{ AND } A0 \text{ AND } D5 & \text{ OR} \\
A2 \text{ AND } A1 \text{ AND } (\text{NOT } A0) \text{ AND } D6 & \text{ OR} \\
A2 \text{ AND } A1 \text{ AND } A0 \text{ AND } D7 & \\
\end{align*}
\]
Output Selector

D0
D1
D2
D3
D4
D5
D6
D7
A0
A1
A2
AND
AND
AND
AND
AND
AND
AND
AND
AND
AND
AND
AND
AND
OR
DATA OUT
Input Decoder

• 8 Bit Latch allows 8 bits to be written at the same time

• Now: only write 1 bit at a time

• Select the bit with an address

• Input
  – address
  – write flag
  – data bit
Input Decoder

3-to-8 Decoder

WRITE

ADDRESS

DATA IN

DI W
DO

DATA OUT

DI W
DO

DI W
DO

DI W
DO

DI W
DO

DI W
DO

DI W
DO

DI W
DO

DI W
DO
## Input Decoder

- **Truth table**

<table>
<thead>
<tr>
<th>Address</th>
<th>W7</th>
<th>W6</th>
<th>W5</th>
<th>W4</th>
<th>W3</th>
<th>W2</th>
<th>W1</th>
<th>W0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>WRITE</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>WRITE 0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>WRITE 0 0</td>
<td></td>
</tr>
<tr>
<td>0 1 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>WRITE 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>WRITE 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 1</td>
<td>0</td>
<td>0</td>
<td>WRITE 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 0</td>
<td>WRITE 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1</td>
<td>WRITE 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- What Boolean operation returns the correct value for output W0?

\[(\text{NOT A2}) \text{ AND} (\text{NOT A1}) \text{ AND} (\text{NOT A0}) \text{ AND WRITE}\]
Input Decoder

WRITE

A0 A1 A2

AND

W0

AND

W1

AND

W2

AND

W3

AND

W4

AND

W5

AND

W6

AND

W7
8 Bit RAM

WRITE

ADDRESS

DATA IN

8-Bit Selector

DATA OUT

3-to-8 Decoder

DI
DO
W

DI
DO
W

DI
DO
W

DI
DO
W

DI
DO
W

DI
DO
W

DI
DO
W

DI
DO
W

DI
DO
W

Philipp Koehn
Computer Systems Fundamental: Memory
14 February 2018
8 Bit RAM

• 8 Bit Random Access Memory (RAM)

• Input
  - address
  - write flag
  - data bit

• Output
  - data bit
8x2 Bit RAM

• 8x1 bit RAM allows read/write of 1 bit at a time

• What if we want to read/write 2 bits at a time? (and ultimately 8 bits (1 byte) and more)

⇒ Arrange them together
8x2 Bit RAM

DATA-IN0  WRT  ADDR  DATA-IN1

8x1 Bit RAM

DATA-OUT0  DATA-OUT1
8x2 Bit RAM

![Diagram of a 8x2 Bit RAM module with inputs DI, W, A and output DO.](image-url)
8x2 Bit RAM

DI  W  A

DO

8x2 Bit RAM
• 64KB = 65,536 bytes

• 16 bit address space \(2^{16} = 65536\)

• Common memory size in the 1980s: we will use it with 6502 assembly
Control Panel

64-KB RAM Control Panel

A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

Write

Takeover
Early 1980s: 64 KB RAM, 16 bit address space
Bigger Memories

• Early 1980s: 16 bit address space, up to 64 KB

• 1990s: 32-bit address space, up to 4 GB

• Today: 64-bit address space, up to 16 EB (exa-byte)

• Actually supported by Intel/AMD 64-bit processors
  - 52 bits for physical memory: 4 peta-byte
  - 48 bits for virtual memory: 256 tera-byte

• Actually existing RAM: my lab biggest RAM machine: 768 GB
  (doubles every ~2 years)