Basic Memory Circuits

Latch

Single-bit memory, level-triggered

Flip-Flop

Also single-bit, but edge-triggered

SR (Set Reset) Latch

SR latch uses 2 cross-coupled NOR gates:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>(A+B)'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

What does unstable/non-deterministic mean? Notice in a NOR gate, if either input = 1 to a gate, its output = 0 (deterministic input)

You wouldn't usually try to set and reset at the same time (it doesn't make sense), but if you did, Q = Q' = 0. This is predictable.

However, when you go back to the remember state (S=R=0) from S=R=1, Q and Q' do not stay at 0 (the circuit does not 'remember' the previous output). Instead, Q and Q' change to the complement of one another.

But, you cannot predict whether Q will be 1 or 0! The final state depends on which of the inputs (S or R) being set to 0 is sensed first in the circuit. It is for this reason that the S=R=1 state is called non-deterministic, or unstable.
Here is an equivalent circuit for the SR latch, using NAND gates only:

![NAND gate circuit for SR latch]

Works the same, except when $S = R, Q = Q' = 1$ (instead of 0 as in the NOR gate configuration). Why?

For the NAND gate, if either input is 0, the output is 1 (so 0 is the deterministic input)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>(AB)'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Clocked SR Latch** incorporates a clock input

![Clocked SR latch circuit]

Change occurs only on a high clock level.
3 circuits which avoid non-deterministic state

D Latch

\[ \begin{array}{c|c}
D & Q_{\text{next}} \\
\hline
0 & 0 \\
1 & 1 \\
\end{array} \]

T Latch

\[ \begin{array}{c|c}
Q_{\text{prev}} & Q_{\text{next}} \\
\hline
0 & 1 \\
1 & 0 \\
\end{array} \]

JK Latch

\[ \begin{array}{c|c|c}
J & K & Q \\
\hline
0 & 0 & Q_0 \text{ (remembers)} \\
0 & 1 & 0 \\
1 & 0 & 1 \\
1 & 1 & \text{toggles} \\
\end{array} \]
Re-visiting the Binary Counter

Now that you understand how flip-flops work, you can explain the operation of the binary counter from Laboratory 7.

Four T flip-flops were interconnected, with the output of each flip-flop tied into the input of the next flip-flop.

QA serves as the clock to QB. So, QB only changes when QA falls from 1 to 0 (on the negative edge). QB therefore only changes half as frequently as QA.

A similar relationship exists for QB to QC, and QC to QD.

The pattern of outputs then represents the binary numbers, since that is exactly how the digits change as the numbers increment.
Register: n-bit memory, shared clock and clear inputs
Register File – set of registers

- **Write** is the write control signal.
- **Write register** is the single register to be written to at a time
- **Read register number 1 and 2** indicate which 2 registers can be read at data ports **Read data 1** and **Read data 2**
- clear and clock (**CLR** and **CLK**) are shared by all the 16 registers.
- **CLR** is active low
LogicWorks register file
Internally:

- 2 sets of 16 x 1 multiplexers select which 2 registers are currently being output at the two read ports.

- A 4x16 decoder uses the write register number to select which of the 16 registers will receive a new value on a write.
Memory Devices

- Access time is the same for any location in memory

- RAM (random access) memory:
  - volatile (loses its contents if power is lost)
  - Static (SRAM) uses flip-flops as an underlying technology
  - Dynamic (DRAM) uses capacitors as an underlying technology
  - SRAM is faster but more expensive than DRAM

- ROM (read only) memory:
  - non-volatile
  - Program or data are pre-set and do not change.