Encode this table as a circuit

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(Your response)
Encode this table as a circuit

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>1 0 0 0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0 0 1</td>
</tr>
</tbody>
</table>

2-bit decoder

(Your response)
Levels of abstraction

- Stored-program computers
- Random-access memory (RAM)
- Registers
- 1-bit memory: flip-flops
- Logic gates
- Transistors / switches

Today: Tuesday
Next week: Tuesday
Pass-through

I’m (not) pressing the button right now.
Phase behavior: set

I (haven’t yet) pressed the button. (toggle on)
Phase behavior: set / reset

I can toggle the button on and off.

[Diagram of S/R latch with waveforms for reset, set, and output over time t0 to t7]
Pass-through via set / reset

I’m (not) pressing the button right now.
Pass-through *or* set / reset

I can toggle between pass-through and remembering the state from a moment in time.

![D latch diagram]

**Strobe** | **Data** | **Output** | **Time**
--- | --- | --- | ---
0 | 0 | 0 | t0
0 | 1 | 0 | t1
0 | 0 | 1 | t2
0 | 1 | 0 | t3
0 | 1 | 0 | t4
0 | 1 | 0 | t5
0 | 1 | 0 | t6
0 | 1 | 0 | t7
Random-access memory (RAM)

A 512K x 8 RAM (About 4.2 million bits)
A small piece of RAM

Interface: we can read or write one of four rows of memory, and each row stores three bits.
A small piece of RAM
A *small* piece of RAM

Addressing: select which “line”
A small piece of RAM

Write mode
A small piece of RAM
A small piece of RAM

Wire data bits to corresponding memory bits
A *small* piece of RAM

Wire data bits to corresponding memory bits
A *small* piece of RAM

Wire data bits to corresponding memory bits