

7. Basic processor design – Sequential logic

EECS 370 – Introduction to Computer Organization – Winter 2007

Prof. Valeria Bertacco & Prof. Scott Mahlke

EECS Department
University of Michigan in Ann Arbor, USA

© V. Bertacco & S. Mahlke, 2007

The material in this presentation cannot be
copied in any form without our written permission

Announcements

- ❑ New professor today – Scott Mahlke (mall key)
 - For the next 3 classes, then Valeria for Exam 1 review and Exam 1, then me for the next 1 and a half months
- ❑ Homework 2 – Due today (Jan 30)
 - Homework 3 available on course website
- ❑ Project 1 – Due Friday (Feb 2)
- ❑ Exam 1
 - Review session – In class, Thursday Feb 8
 - Exam itself – In class, Tuesday Feb 13
 - Open book / open notes → Covers through finite state machines
- ❑ **Regular discussion sections this week – cancelled**
 - GSIs will have extra office hours during discussion slots in the Media Union

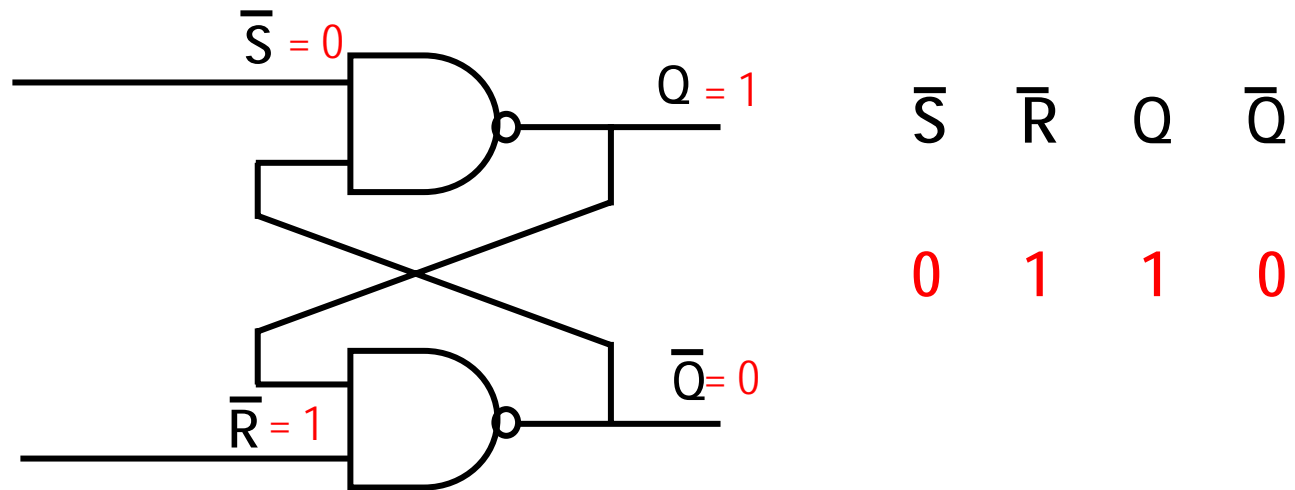
Lecture schedule

1. Combinational Logic:
 - Basics of electronics; logic gates, muxes, decoders
2. **Sequential Logic:**
 - **clocks and data storage**
3. ALU design
 - Building an adding circuit
4. State Machines
 - Building a simple processor

Levels of abstraction

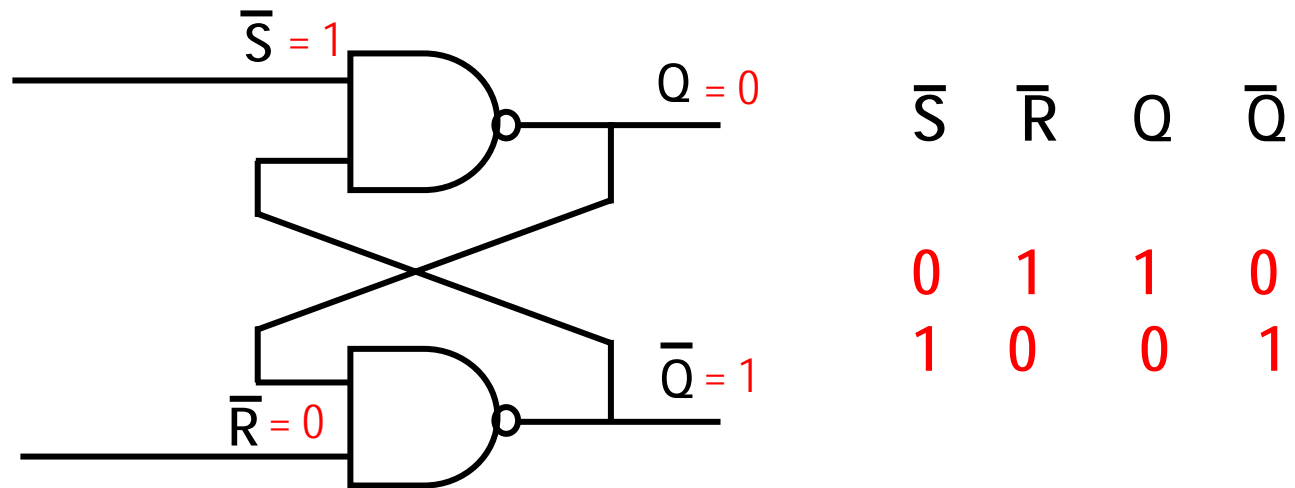
- ❑ Quantum level, **solid state physics**
- ❑ **Conductors, Insulators, Semiconductors.**
- ❑ **Doping silicon to make diodes and transistors.**
- ❑ **Building simple gates, boolean logic, and truth tables**
- ❑ **Combinational logic: muxes, decoders**
- ❑ **Clocks**
- ❑ **Sequential logic: latches, memory**
- ❑ **State machines**
- ❑ **Processor Control: Machine instructions**
- ❑ **Computer Architecture: Defining a set of instructions**

Let's look at the following circuit



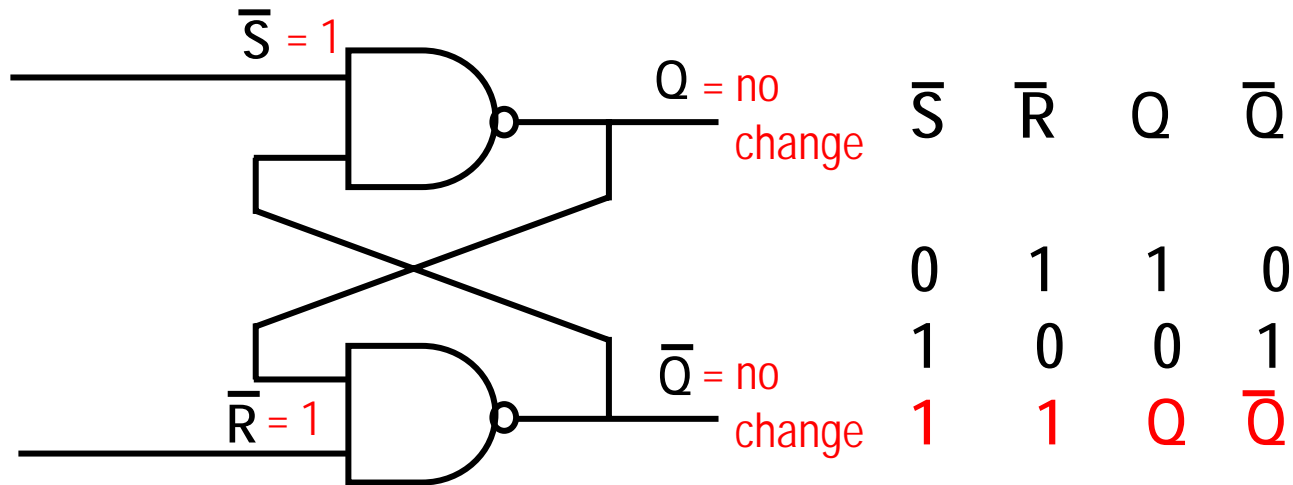
What is the value of Q if \bar{R} is 1 and \bar{S} is 0?

Building the Truth Table



What is the value of Q if \bar{R} is 0 and \bar{S} is 1?

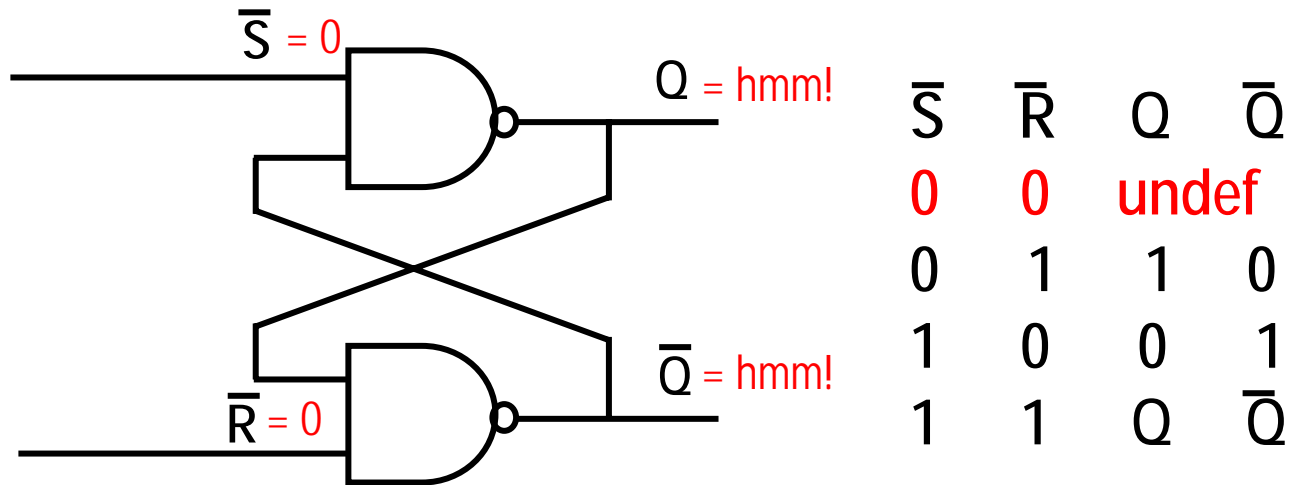
For a Basic Memory Cell



What is the value of Q if \bar{R} is 1 and \bar{S} is 1?

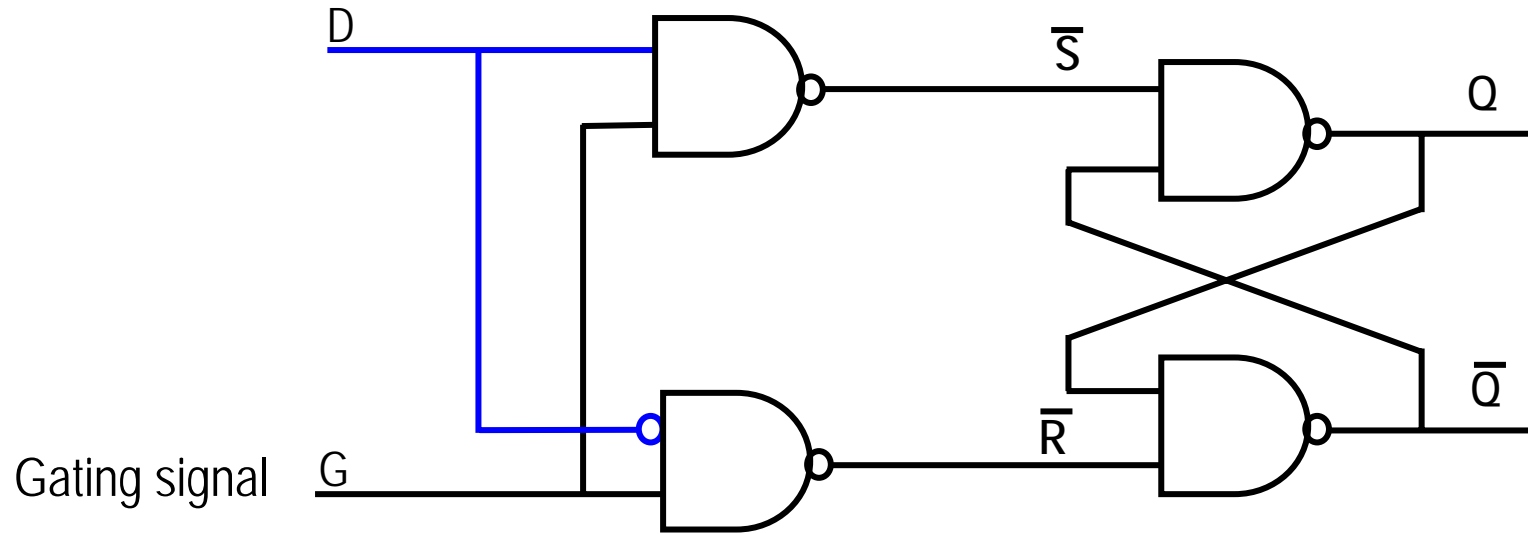
As long as R and S remain 1 then the value Q (and \bar{Q}) will remain unchanged. This value is **stored** in this circuit. This is a basic memory cell.

With unstable inputs 0,0



What is the value of Q if \bar{R} is 0 and \bar{S} is 0?

Transparent D Latch



D	G	Q	\bar{Q}
0	0	Q	\bar{Q}
0	1	0	1
1	0	Q	\bar{Q}
1	1	1	0

Next state is set

Set state is retained

Adding a clock to the mix

- ❑ We can design more interesting circuits if we have a clock signal.
- ❑ The use of a clock enables a sequential circuit to **predictably** change state (and store information).
- ❑ A clock signal alternates between 0 and 1 states at a fixed frequency (e.g. 100MHz)
- ❑ What should the clock frequency be?

Clocks

□ Clock signal

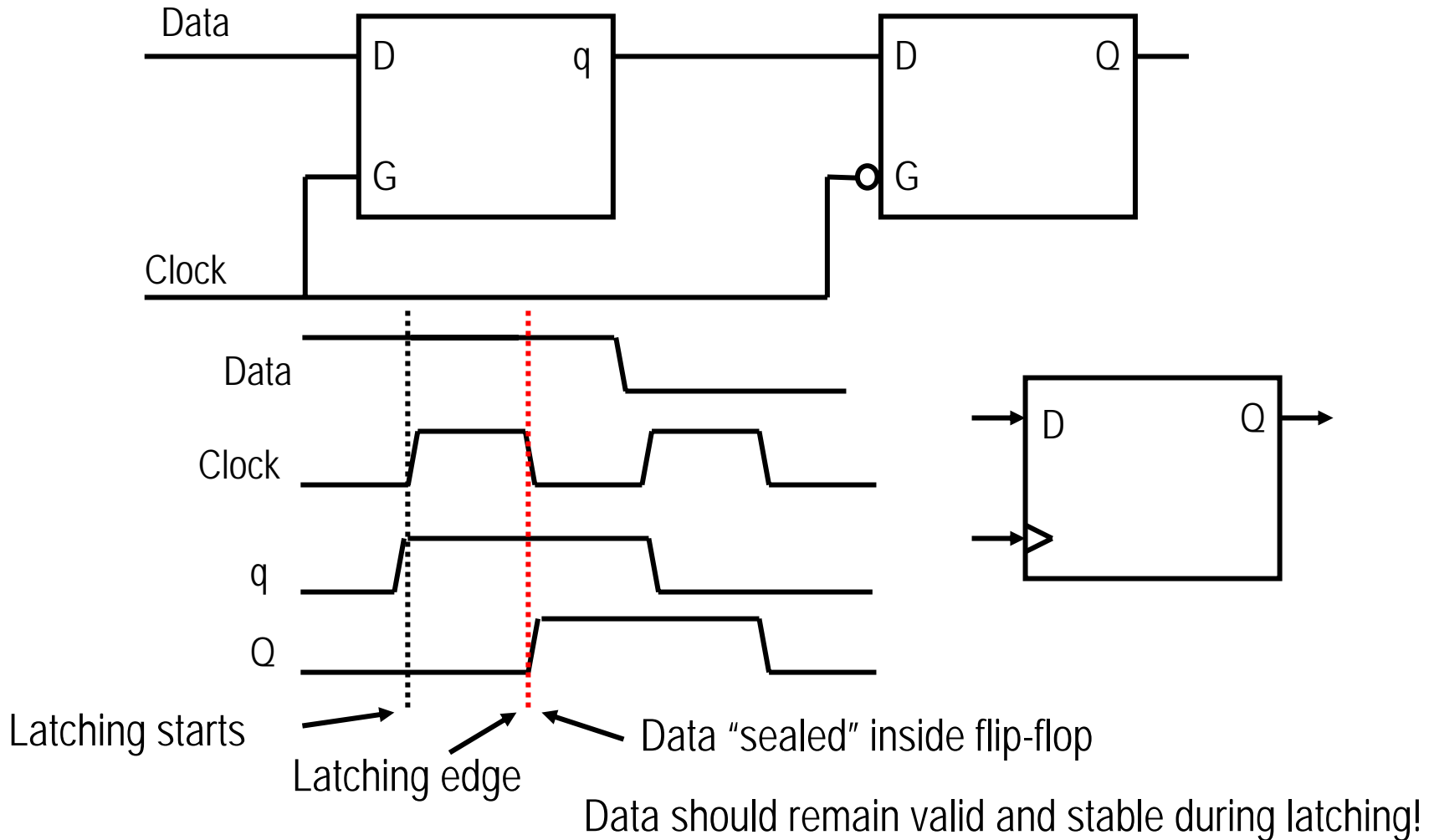
- Periodic pulse
- Generated using oscillating crystal or ring oscillator
- Distributed throughout chip using clock distribution net



□ With clock signals we can create a new class of circuits called **sequential**

- Output determined by inputs & **previous** state

Edge Triggered D Flip-flop



Q?

Data

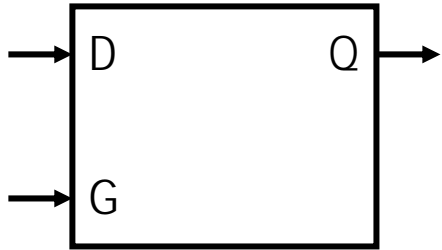


Clock

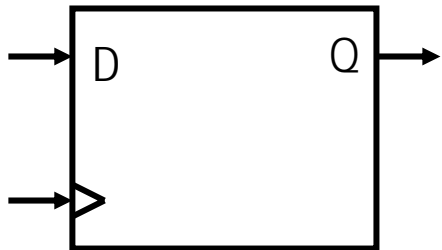


Q

Why edge-triggered flip-flops?



IS LIKE A TOLL-GATE

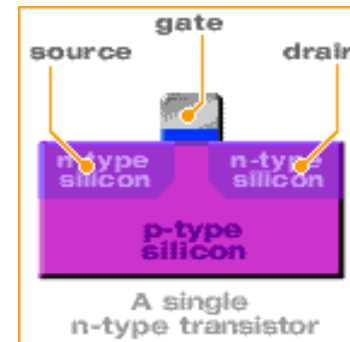
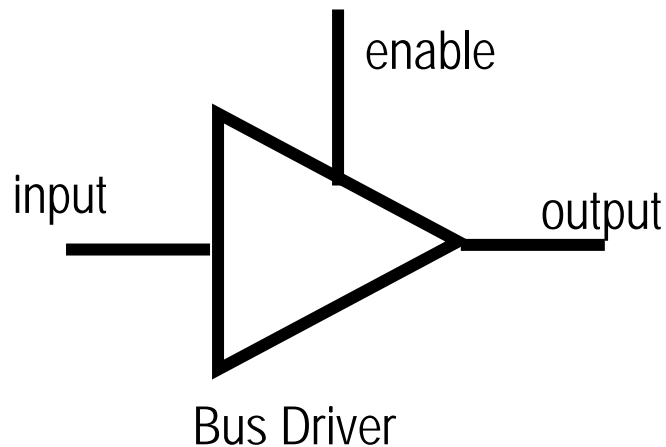


IS LIKE AN AIR-LOCK

In edge-triggered flip-flops, the latching edge provides convenient abstraction of “instantaneous” change of state.

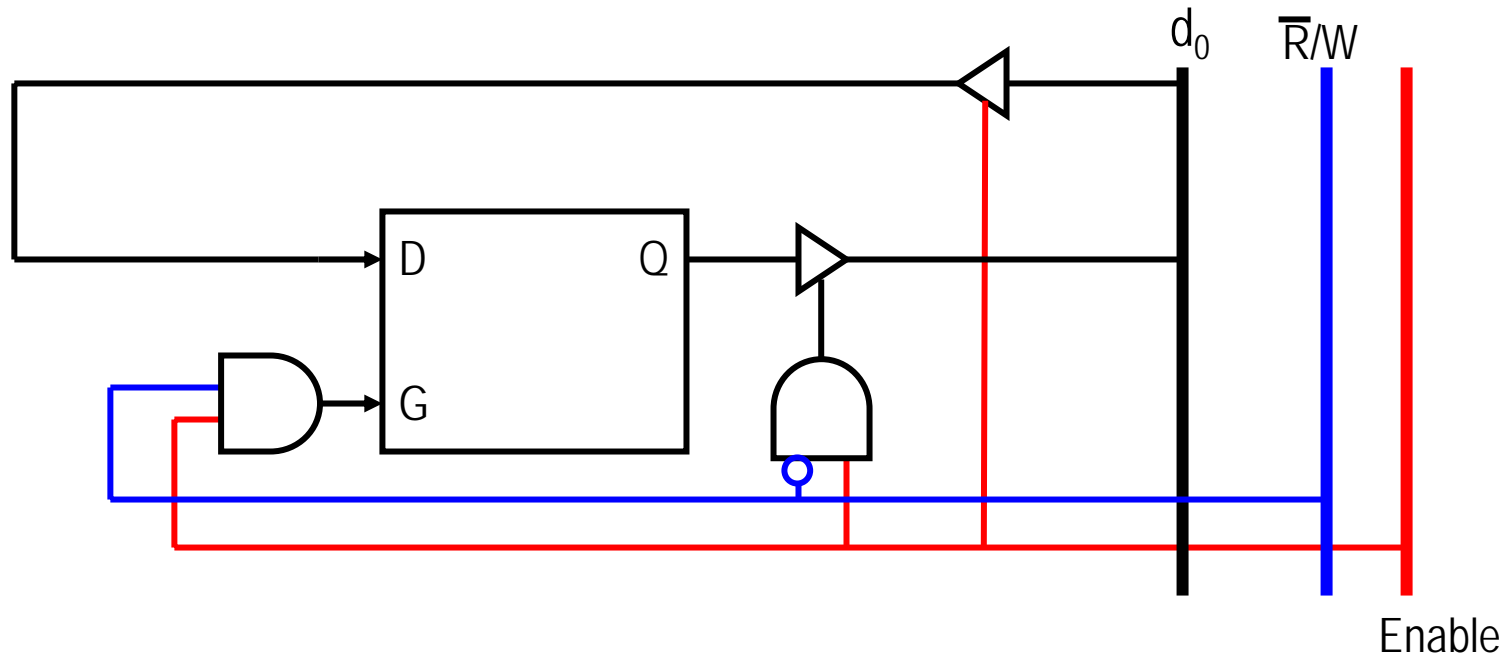
Tri-state Logic

- The output of a gate can be any of **three** different states: one, zero or not connected
 - Need to disconnect the circuit. How?

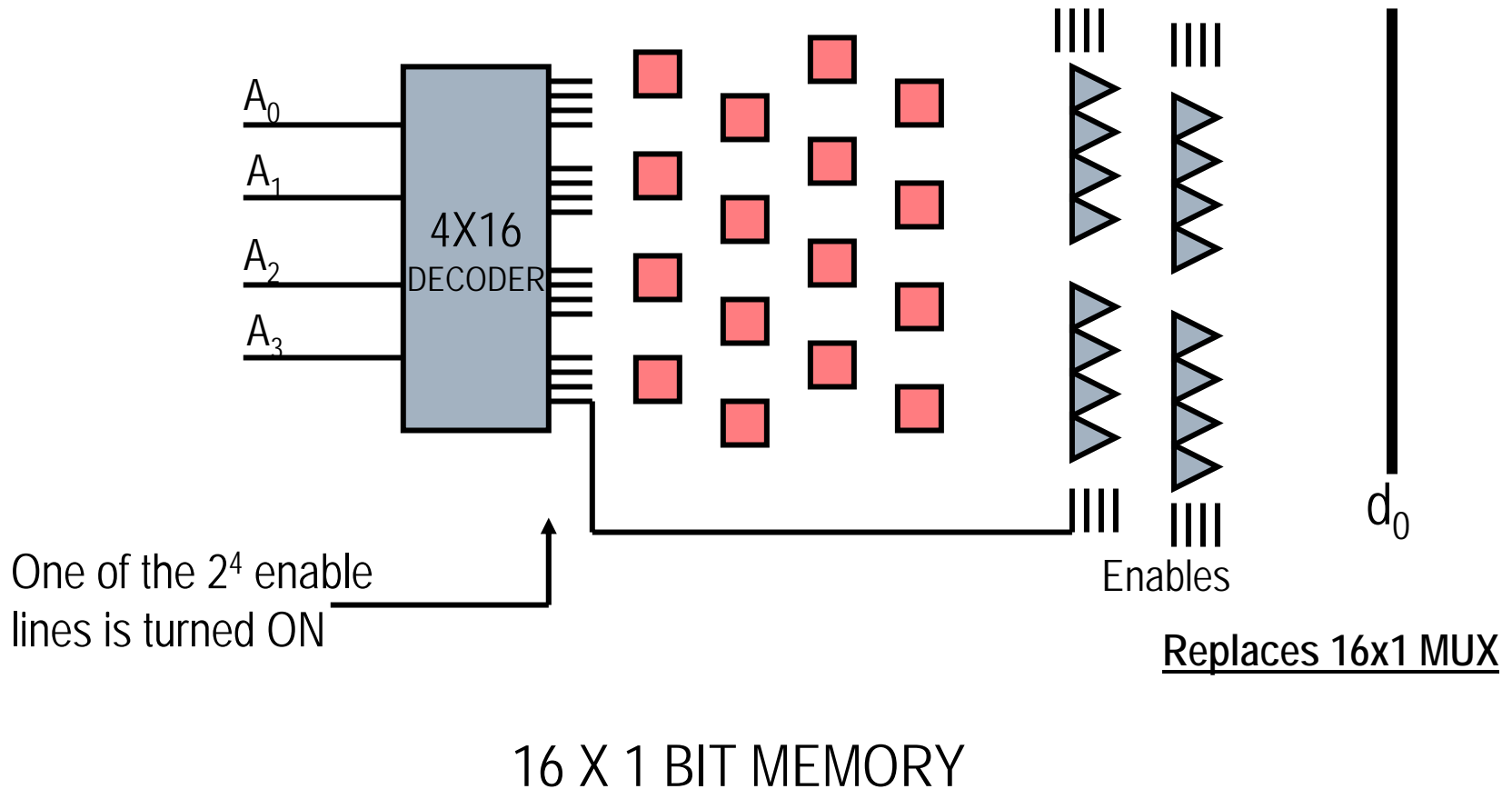


Implemented as
a single transistor

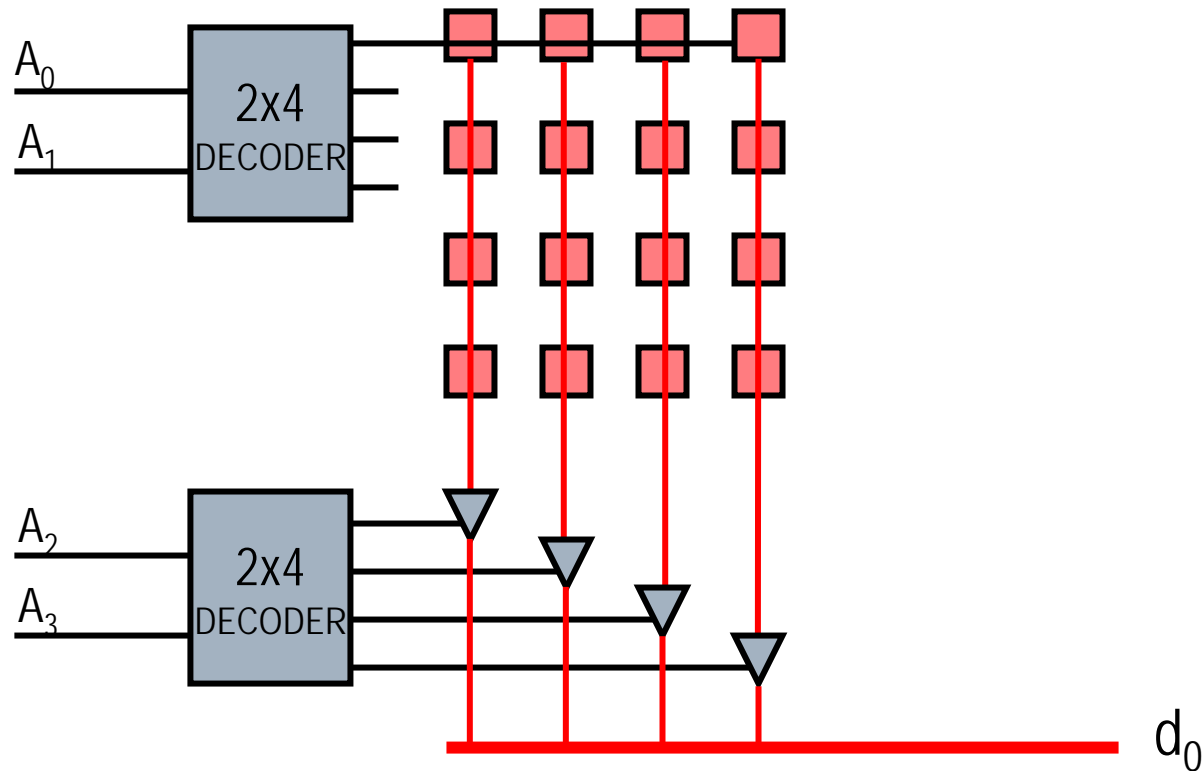
A Static Memory Cell



Addressing Memory Arrays



A Scheme with Fewer Components



16 X 1 BIT MEMORY

Putting it in a package



18-bit X 2M MEMORY (36 MBits)

Other Memories

❑ Static RAM

- Built from sequential circuits
 - Takes 4-6 transistors to store 1 bit
 - Fast access (< 1 ns access possible)

❑ Dynamic RAM

- Built using a single transistor and a capacitor
 - 1's must be refreshed often to retain value
 - Slower access than static RAM
 - Much more dense layout than static RAM

❑ ROM, PROM, Flash memory, etc. → Later

