

ECE 2020: Latches and Flip-Flops

Instructor: Samuel Talkington

October 22, 2024

Logistics

- **Exam 2:** You did it!
- **Lab report 1:** Grades now available; you all did really good
- **Old notes for this module** from a previous semester are now available on Canvas; our focus may differ, but I want you to have as many resources as possible.
- **We'll discuss the mid-semester survey** this week

Coming soon

- **Problem set 4:** Released around the end of the week.
- **Prelab 2:** Released around the end of the week.
- **Lab 2:** Planning for October 31st, 2024.
- I will try to bring candy

Redemption

I believe engineering is an iterative process¹, so we will have these redemption opportunities on the following big-ticket assignments:

- Exam 2
- Lab reports—final lab portfolio due by the end of the semester.

¹and I'm passionate enough about this job to spend hours regrading for you all

Exam 2 revisions

- I will do everything in my power to grade Exam 2 before the drop deadline.
- After grades are released, you will have the opportunity to make a revised submission with up to **one** teammate—containing the problems you **both** got wrong—for partial credit towards Exam 2.
- Be prepared to indicate your teammate during today's participation puzzle.
- Collaboration is **not allowed** outside of you and your teammate.
- If resubmitting, you must include a ≈ 1 page reflection statement.

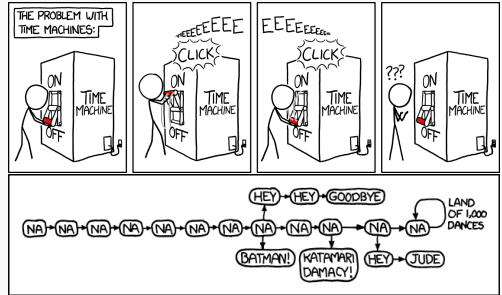
Reflection statement guidelines

- Prepare a 1-2 paragraph summary of what we learned in Exam 2. I do not want just a bulleted list of topics, I want you to use complete sentences and establish context (Why is what we have learned relevant? How does it connect with other classes?). *The more insight you give, the better.*
- Prepare a 1 paragraph statement that **explicitly** describes the contributions, improvements, and learning achieved by each teammate.
- Each teammate must explicitly describe their contributions and learning to the new submission in the reflection to receive credit.
- Submissions with unclear disclosures of both teammates contributions will receive zero partial credit.

Next up: the core of computers

Agenda: next 2 weeks

- Sequential logic
- Latches
- Flip flops
- State machines



Source: xkcd

Sequential logic

Class discussion

So far, we have assumed that an input to a system *completely determines the output*.

- Would you plug $2 + 2$ into a calculator over and over expecting a different result?
- Is this kind of thing ever realistic? If so, when? When is it not?
- What about memory—e.g., food delivery vehicles, surveillance cameras
- What are some “stateless” systems—e.g.: virtual machines, functional programs

combinational logic

- Input combination completely determines the output
- The system behaves the same way at all times
- No **storage or memory**; systems cannot remember anything
- No delay
- Described with truth tables and Boolean algebra
- Gates, building blocks

sequential logic

- Outputs can vary with the same inputs
- The system behavior can evolve over time
- Systems can have **“state”**; with this, they can remember things
- Can have delay
- Described with timing diagrams and transition tables
- Latches and flip-flops

Sequential logic

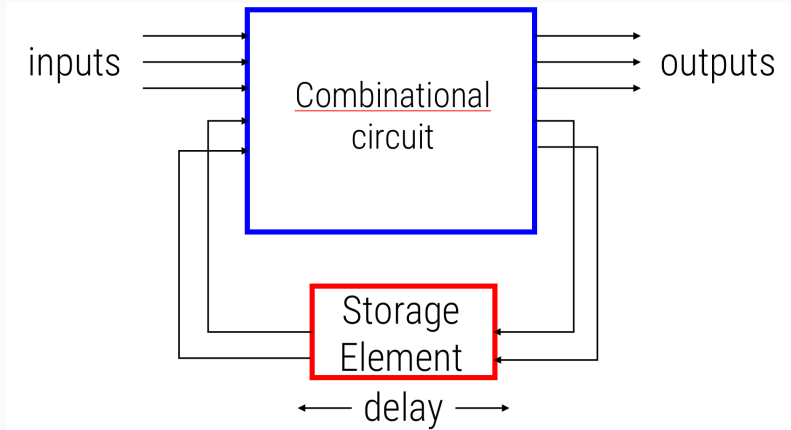


Figure 1: *It's alive!* Sequential circuit = combinational circuit + state and delay

Latches

Can we make memory with...time delays?

Basic latch

What it does

- It's a "hacky" approach to making 1 bit of **read-only** memory.
- "Stores" a bit value, and allows us to set the value being stored
- Relies on setting inverter time delays to function

Definition: Transition Tables

A **transition table** is a table showing what **state** a system will move to, based on the **current state and other inputs**.

$$Q \rightarrow Q^+$$

Think of it as a **truth table that can work with *time and memory***.

Setting and resetting with NOR gates

Set-Reset (SR) NOR Latch

What it does

Stores one bit of **memory**—made with NOR gates.

S	R	Q^+	\overline{Q}^+
0	0	Q	\overline{Q}
0	1	0	1
1	0	1	0
1	1	undef	undef

Table 1: SR-NOR *characteristic* table

Setting and resetting with NAND gates

Set-Reset (SR) NAND Latch

What it does

Stores one bit of **memory**, made with NAND gates.

S	R	Q^+	\overline{Q}^+
0	0	Q	\overline{Q}
0	1	0	1
1	0	1	0
1	1	undef	undef

Table 2: SR-NAND *characteristic* table

Transparency and proto-transition tables

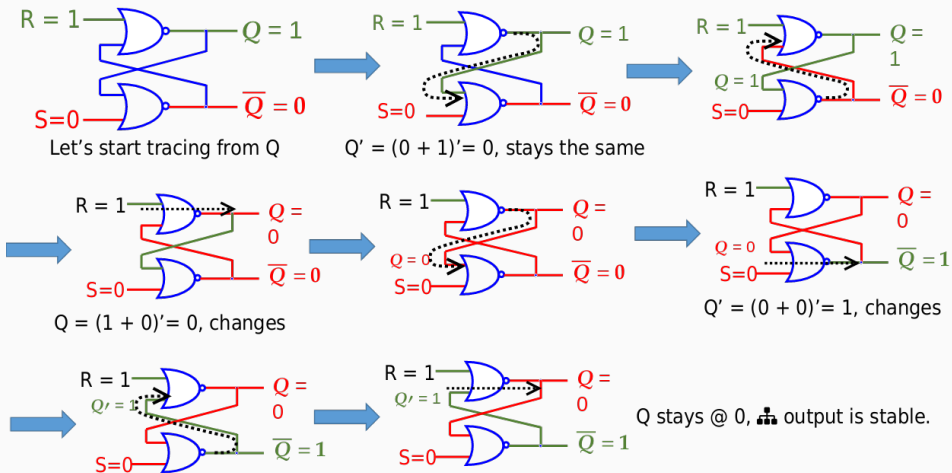
Note

For latches, our truth tables are more accurately called *characteristic tables* or *proto-transition tables*. They describe

$$Q \rightarrow Q^+, \quad \text{and} \quad \overline{Q} \rightarrow \overline{Q}^+,$$

because the Latches are *transparent*. This means they basically act like the combinational circuits we've seen so far—the input is instantly passed to the output through the latch. **(more on this later).**

Tracing latches over time

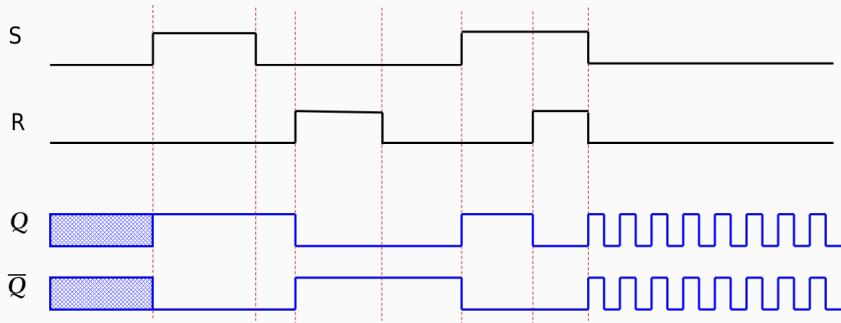


The failure of latches

Latches can be unstable

- Remember that we can use **timing diagrams** to show how digital systems change over time
- Unfortunately, latches can create unstable switching conditions.
- Let's look at an example.

Switching instability example



Prelude to flip-flops

To fix the instability of latches, we need to prevent the **forbidden (undefined) state**:

$$S = 1 \quad \text{AND} \quad R = 1.$$

With this modification, we can develop a physical representation of **data—memory**. (It's super cool.)

Data Latch—a usable basic unit of memory

Data Latch (with SR-NOR)

What it does

One bit of memory, can **read/write**,
and **forbidden state is fixed!**

En	D	Q^+	\overline{Q}^+
0	0	Q	\overline{Q}
0	1	Q	\overline{Q}
1	0	0	1
1	1	1	0

Table 3: Data latch characteristic table.

Building a data latch with SR-NOR latch

Building a data Latch with SR-NOR latch

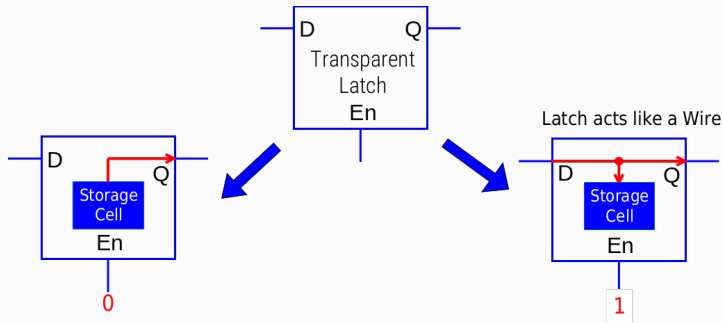
Note

Building a data latch with an SR-NAND latch is possible as well. **Try it out to help you succeed in this module.**

Fixing forbidden state is not enough

- The issue with latches goes deeper than just the forbidden state.
- The **transparency property** itself is a problem.
- The input follows the output immediately—it's “level-sensitive”.

Latches are transparent



5

Figure 2: The inputs and outputs of a Latch are transparent to one another.

Latches: can we do better?

Wouldn't it be nice if we could just...fix latches?

Wishlist:

- Activated by *transitions*—the edges of timing diagrams—*foreshadowing: state machines*.
- No forbidden state, no transparency
- Only holds the signal value taken at the edge instantaneously—“edge-sensitive”.

Could this fix unstable switching?

Flip flops

Edge detector

Example: AND gate
with multiple
inverters, fed into
D-Latch

The data flip-flop

What is it

A data flip-flop is the combination of an edge detector + data latch. It is **edge-triggered** memory.

Symbol + equation

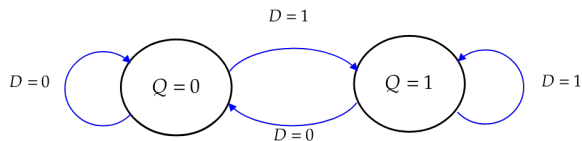
Characteristic table

in_k	state_k	state_{k+1}	function
D_k	Q_k	Q^+	\overline{Q}^+
0	0	0	reset
0	1	0	reset
1	0	1	set
1	1	1	set

State transition

For flip-flops, we can now truly make a transition table!

State diagram



Transition table

D	clk	Q
0	\uparrow	0
1	\uparrow	1

Types of activation characteristics for the D flip flops

D flip-flops are like Pokemon—there are multiple types. Each type has different activation, or “trigger” characteristics.

Positive edge

A diagram of a positive edge-triggered D flip-flop. It consists of a large rounded rectangle with a black border. At the top center, there is a small black rounded rectangle containing the text "Positive edge" in white.

Negative edge

A diagram of a negative edge-triggered D flip-flop. It consists of a large rounded rectangle with a black border. At the top center, there is a small black rounded rectangle containing the text "Negative edge" in white.

Dual edge

A diagram of a dual edge-triggered D flip-flop. It consists of a large rounded rectangle with a black border. At the top center, there is a small black rounded rectangle containing the text "Dual edge" in white.

Puzzle

Indicate your teammate for the revisions and lab 2 on the “People” tab in canvas.

Fill in the participation puzzle on canvas to reflect on the past module.

Password: state.

Puzzle 10/22

Fill in this timing diagram for a D flip-flop. Discuss with your neighbor and share. Can you repeat for rising edge-trigger? What about a dual-edge trigger?

