Lecture 5 — Introduction to Sequential Circuits

Table of contents

See flip-flops-and-latches[1] and sequential-circuits[2] for the reference definitions, characteristic tables and symbols this lecture introduces.

Today's outline

- Recap of last week's combinational logic
- Circuits that remember values
- The D flip-flop: symbol, operation, characteristic table, waveforms
- The SR latch built from NOR gates
- Flip-flops vs latches; a real D flip-flop and its chips
- Combinational vs sequential circuits; synchronous sequential circuits

[1] courses/csse2010/flip-flops-and-latches.pdf
[2] courses/csse2010/sequential-circuits.pdf

Recap from last week

The course so far stacks up in three layers, each built on the one below:

1. **Binary representations** — unsigned, sign-magnitude, 1's complement, 2's complement, excess-2^{N-1} (see binary-number-representations[3], 2026-08-04-binary-arithmetic[4]).
2. **Logic gates** — NOT, AND, OR, NAND, NOR, XOR, XNOR — and Boolean algebra (see logic-gates[5], boolean-algebra[6]).
3. **Combinational logic circuits** — adder, adder/subtractor, multiplexer, decoder (see combinational-logic-blocks[7]).

The recap slide shows the 4-bit ripple carry adder (four full adders chained by their carries), the 4-bit adder/subtractor with its mode select M ($M = 0$ add, $M = 1$ subtract), a 4:1 MUX with selects $S_1 S_0$, and a 3:8 decoder with inputs A, B, C and outputs $D_0 \ldots D_7$.

Quiz: multiplexer implementing a given function

Consider a 4:1 multiplexer with data inputs A, B, C, D on lines 0, 1, 2, 3 respectively and select inputs $S_1 S_0$. What must A, B, C, D be so that the multiplexer output is

$$X = S_1.S_0 + \bar{S}_1.G$$

The options were: (1) $A = 0, B = 0, C = 0, D = 1$; (2) $A = G, B = G, C = 0, D = 1$; (3) $A = G, B = 0, C = 0, D = 1$; (4) $A = 0, B = G, C = 0, D = 1$; (5) none of the above; (6) I don't know.

Reasoning. A 4:1 MUX passes whichever data input its select code addresses, so in general

$$X = \bar{S}_1 \bar{S}_0.A + \bar{S}_1 S_0.B + S_1 \bar{S}_0.C + S_1 S_0.D$$

Now expand the target expression into the same four select terms. The $\bar{S}_1.G$ term covers *both* $\bar{S}_1 \bar{S}_0$ and $\bar{S}_1 S_0$, and the $S_1.S_0$ term is 1 for that one select code only:

$$X = \bar{S}_1 \bar{S}_0.G + \bar{S}_1 S_0.G + S_1 \bar{S}_0.0 + S_1 S_0.1$$

Matching term by term against the MUX expression: $A = G$, $B = G$, $C = 0$, $D = 1$.

[3] courses/csse2010/binary-number-representations.pdf
[4] courses/csse2010/2026-08-04-binary-arithmetic.pdf
[5] courses/csse2010/logic-gates.pdf
[6] courses/csse2010/boolean-algebra.pdf
[7] courses/csse2010/combinational-logic-blocks.pdf

Answer: option 2.

Note the trap: it's tempting to put G on only one of the two $\bar{S}_1$ lines, but $\bar{S}_1.G$ says nothing about S_0, so *both* $\bar{S}_1$ inputs must carry G.

Circuits that remember values

- The output of any logic gate or **combinational circuit** depends on the **current value of the inputs only**.
- If an input changes, the output can change too — and the previous value is **lost forever**.
- In a **sequential circuit**, the current output depends not only on the current inputs but also on the **past** outputs.
- Circuits with **memory** can remember values even when the input changes.

That motivates the whole rest of the course: to build anything that accumulates, counts, or holds a result, you need a storage element.

The D flip-flop

The lecture's first memory element is the D flip-flop: **D** is the input, **Q** the output, and **CLK** the control input. Q copies the value of D — and remembers it — whenever CLK goes from 0 to 1 (the *rising edge*). Full symbol, characteristic table and worked waveform in flip-flops-and-latches[8].

The characteristic table is introduced here as the tabular definition of a flip-flop's operation, with the right-hand column $Q(t+1)$ meaning "what the output will be on the next clock edge".

Summary points from the lecture: a D flip-flop remembers a single bit, so n D flip-flops remember n bits and an n-bit **register** is by definition n D flip-flops. JK and T flip-flops also exist but are not covered in this course. Flip-flops can be made out of logic gates — which is the next slide.

Quiz: D flip-flop output waveform

Using the (rising-edge) D flip-flop presented previously, what is the output waveform for Q, given the D and CLK waveforms shown?

Reading the figure. The clock has four rising edges. D is low, rises just after the first clock pulse has ended, stays high across the second and third clock pulses, then falls before the fourth clock pulse.

[8] courses/csse2010/flip-flops-and-latches.pdf

Rising clock edge	D at that edge	Q after the edge
1st	0	0
2nd	1	1
3rd	1	1 (no change)
4th	0	0

So Q goes high at the **2nd** rising clock edge and stays high until the **4th** rising clock edge, where it returns to 0. Q is *not* a copy of D: it lags D's rise (waiting for the next clock edge) and lags D's fall (holding the 1 until the next clock edge).

Answer: option 3 — the trace that rises at the second rising clock edge and falls at the fourth rising clock edge.

Why the others are wrong:

- **Option 1** rises and falls exactly with D — that's D itself, i.e. no clocking at all.
- **Option 2** rises correctly at the second rising edge but falls at the *falling* edge of the third clock pulse — that would be a negative-edge-triggered device reacting to the wrong edge.
- **Option 4** follows D for the duration of each clock pulse (high while CLK is high and D is high, low again when CLK goes low) — that's **level**-triggered behaviour, i.e. a latch, not a flip-flop.

SR latch from NOR gates

The lecture builds a storage element from two cross-coupled NOR gates: S into the top gate (output $\bar{Q}$), R into the bottom gate (output Q), each gate's output fed back into the other gate's spare input.

The slide's truth table is **blank** — marked "to be completed in class", with a note to attempt it at home beforehand by assuming an initial value for Q and working out the rest for each combination of S and R. The full derivation (hold / reset / set, and why $S = R = 1$ is invalid) is in flip-flops-and-latches[9].

A follow-on slide sets a **homework**: re-analyse that same S-R latch with the NOR gates replaced by NAND gates, and complete the truth table. Worked through in flip-flops-and-latches[10] as well.

[9] courses/csse2010/flip-flops-and-latches.pdf
[10] courses/csse2010/flip-flops-and-latches.pdf

Flip-flops vs latches

Latches are level triggered; flip-flops are edge triggered. A clock has a positive edge (0→1) and a negative edge (1→0), so a D flip-flop can be positive- or negative-edge triggered. Details, plus the four schematic symbols (triangle = edge-triggered, bubble = falling edge) in flip-flops-and-latches[11].

The lecture also shows a real D flip-flop's internal NAND schematic, with its asynchronous active-low $\overline{\text{PRE}}$ and $\overline{\text{CLR}}$ inputs, and the chips that package them — the 74HCT74 (dual) and the 74HCT273 (eight flip-flops, i.e. one byte). See device-pinouts[12] for pinouts, and the device symbols PDF on Blackboard.

Combinational vs sequential circuits

The formal contrast is drawn here for the first time — and re-presented in the next lecture. Combinational circuits are logic gates only, with the output uniquely determined by the inputs; sequential circuits include flip-flops, with the output determined by the current inputs *and* the current state, and the output only able to change when the clock 'ticks'. The general block diagram (combinational logic + flip-flops + feedback path) and the definition of a **synchronous** sequential circuit are in sequential-circuits[13].

Reminders

Coming up:

- **Lab 4** (Mon–Tue this week) — combinational logic. Make sure you attempt the preparation task. Use logic ICs or Logisim software to test your circuits.
- **Lab 5** (Thu–Fri this week) — flip-flops. Use Logisim software to test the circuits. **Kit loaning happens in Lab 5.**

[11] courses/csse2010/flip-flops-and-latches.pdf
[12] courses/csse2010/device-pinouts.pdf
[13] courses/csse2010/sequential-circuits.pdf