

Microprocessor Design Bootcamp: Combinational and Sequential Logic



December 16, 2025
Nursultan Kabylkas

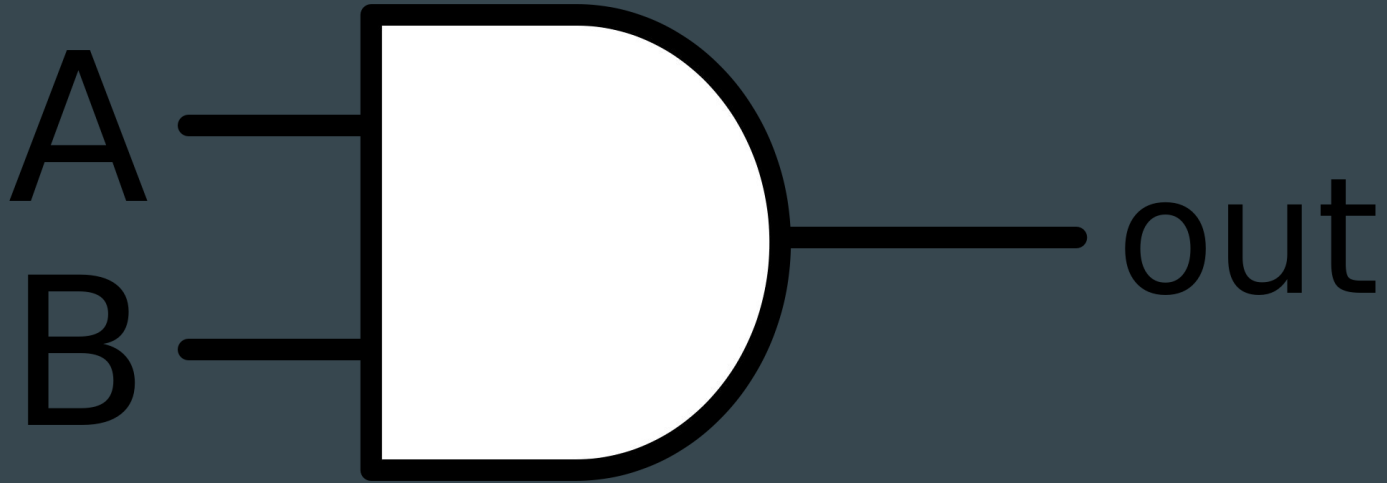
Combinational circuits

- Combinational logic is a type of digital logic where the output is a pure function of the current inputs.

Combinational circuits

- Combinational logic is a type of digital logic where the output is a pure function of the current inputs.
- Key Point: **No memory element**; the output is determined only by the present input values.

Is this a combinational logic?



All of the circuits we built so far were COMBINATIONAL

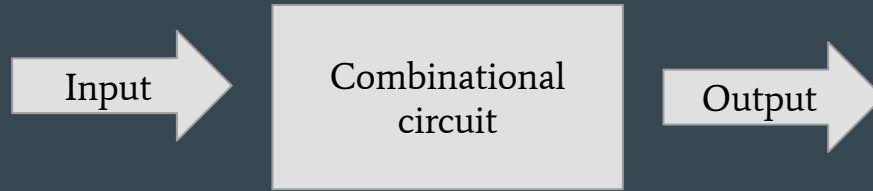
Sequential logic

- Sequential logic's output is determined by its current inputs and also prior outputs.

Combinational
circuit

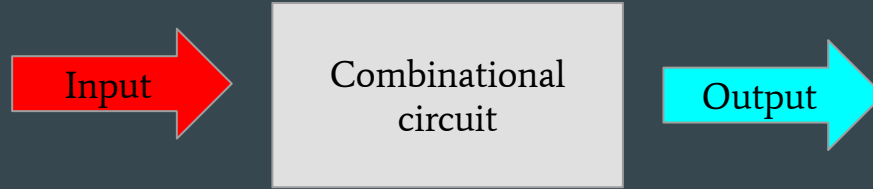
Sequential logic

- Sequential logic's output is determined by its current inputs and also prior outputs.



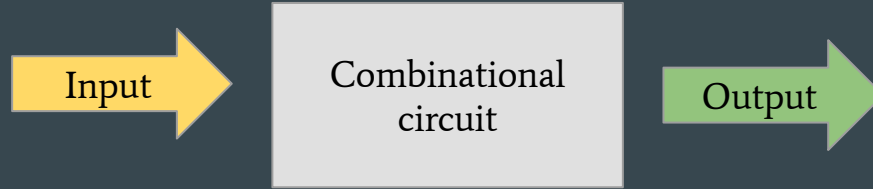
Sequential logic

- Sequential logic's output is determined by its current inputs and also prior outputs.



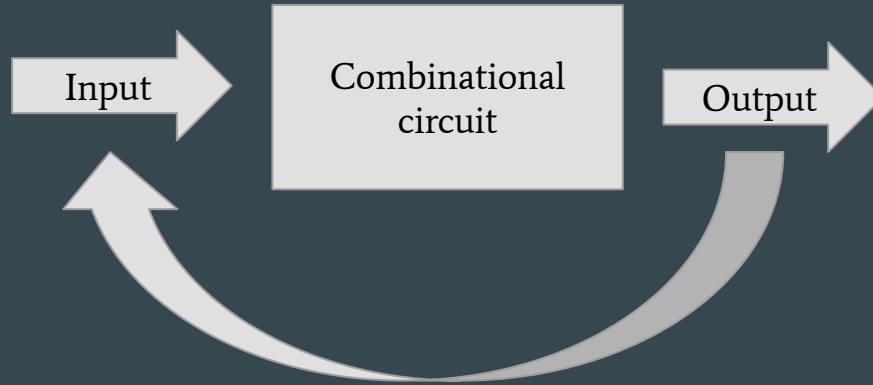
Sequential logic

- Sequential logic's output is determined by its current inputs and also prior outputs.



Sequential logic

- Sequential logic's output is determined by its current inputs and also prior outputs.



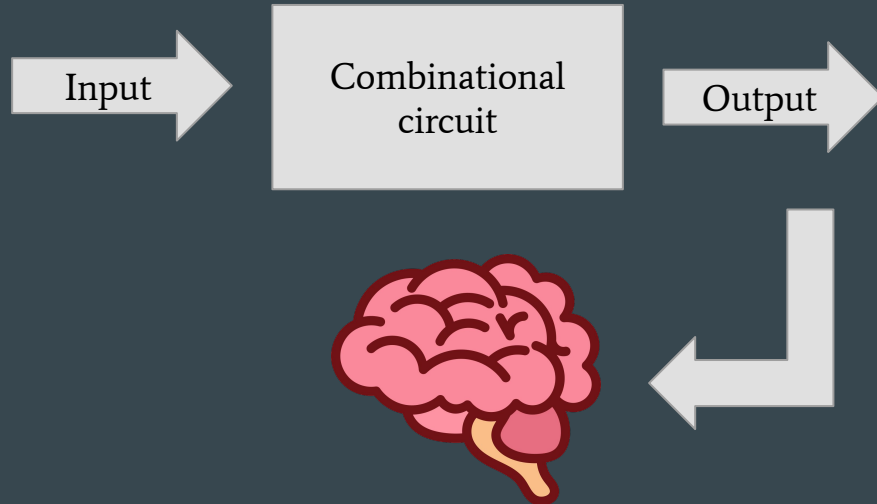
Sequential logic

- Sequential logic's output is determined by its current inputs and also prior outputs.



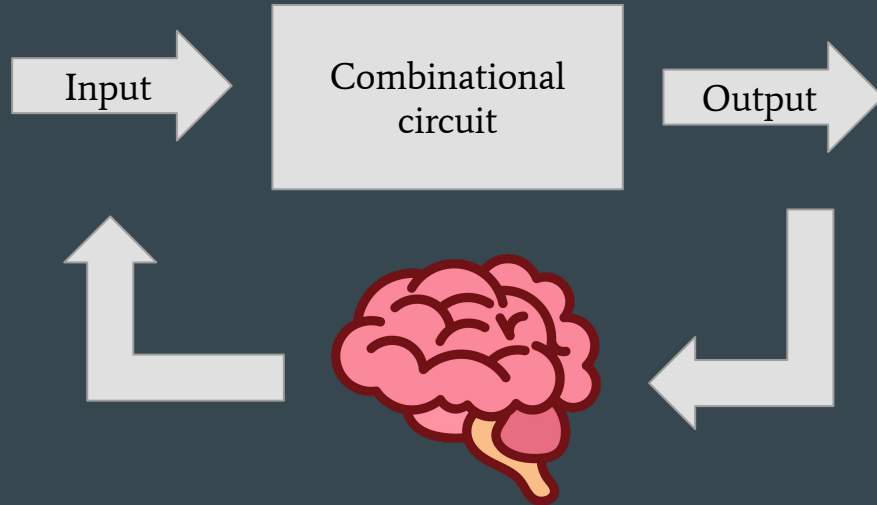
Sequential logic

- Sequential logic's output is determined by its current inputs and also prior outputs.
- For the circuit to be able to “**remember**” prior outputs, there must be a memory element!



Sequential logic

- Sequential logic's output is determined by its current inputs and also prior outputs.
- For the circuit to be able to “**remember**” prior outputs, there must be a memory element!



How can you create a memory element with circuits?

Basic memory element: flip-flop

- Inputs D, CLK and outputs Q and Q-not.



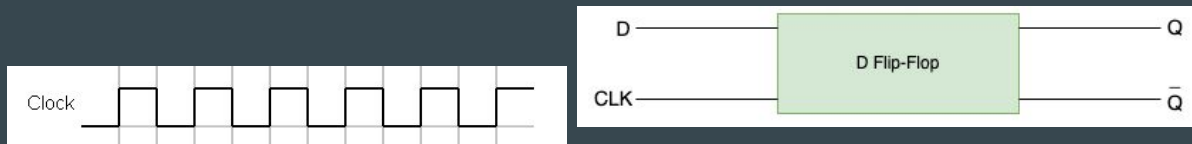
Basic memory element: flip-flop

- Inputs D, CLK and outputs Q and Q-not.
- **Behavior of this circuit: at event CLK, save the value of D to Q.**



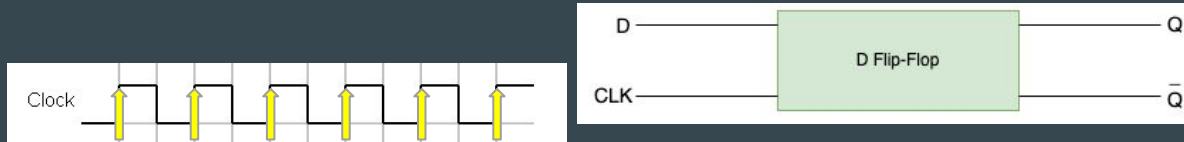
Basic memory element: flip-flop

- Inputs D, CLK and outputs Q and Q-not.
- Behavior of this circuit: at event CLK, save the value of D to Q.
- Clock is a periodic signal, the “save” event is the positive edge of the clock.



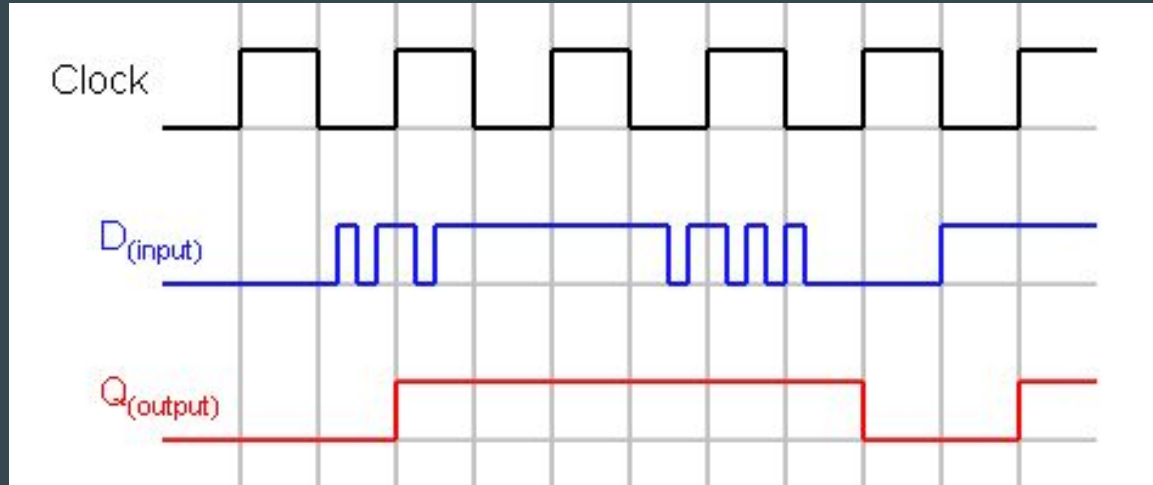
Basic memory element: flip-flop

- Inputs D, CLK and outputs Q and Q-not.
- Behavior of this circuit: at event CLK, save the value of D to Q.
- Clock is a periodic signal, the “save” event is the positive edge of the clock.
- Positive edge means when clock goes from low to high.



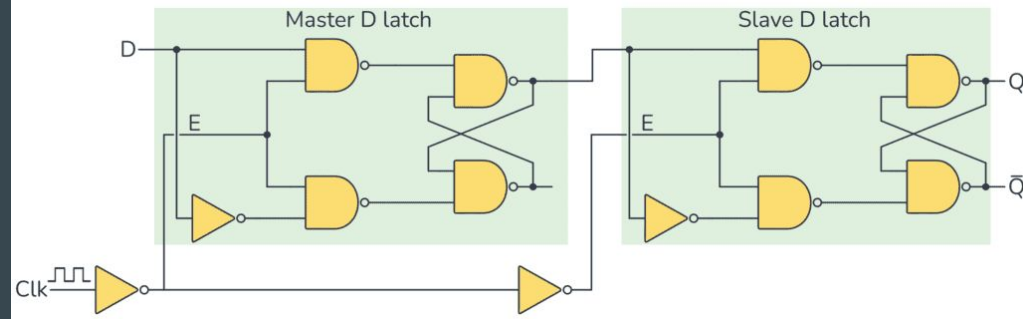
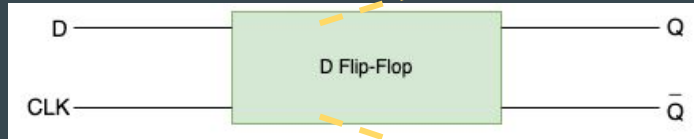
Basic memory element: flip-flop

- Flip-flop behavior: on the positive edge of clock (CLK) pass the value of D to Q.



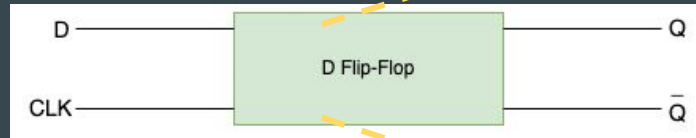
Basic memory element: flip-flop

- What's inside the flip flop?
- Bunch of gates connected in a smart way.



Basic memory element: flip-flop

- Do we care???
- No, hail to the abstraction!



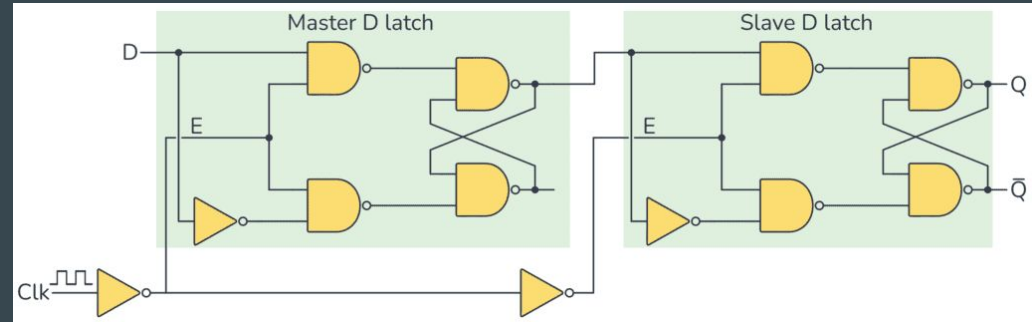
Many types of memory

- Flip-flops
- SRAM
- DRAM
- Flash
- Disk hard drive
- Tape

Many types of memory

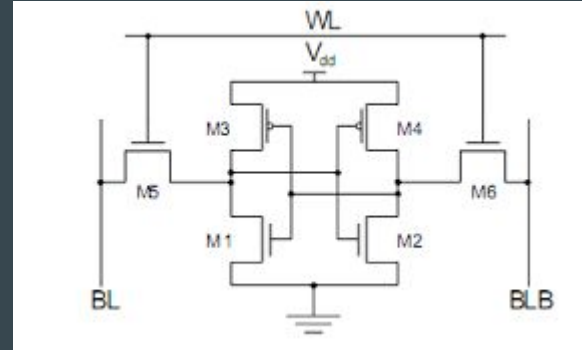
● Flip-flops

- SRAM
- DRAM
- Flash
- Disk hard drive
- Tape



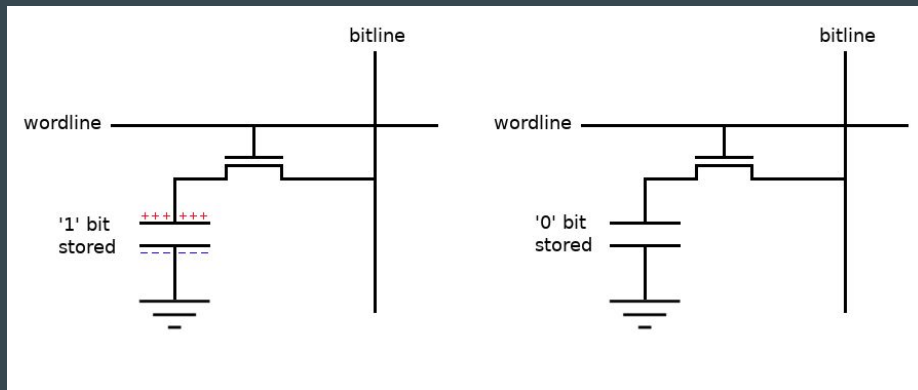
Many types of memory

- Flip-flops
- **SRAM**
- DRAM
- Flash
- Disk hard drive
- Tape



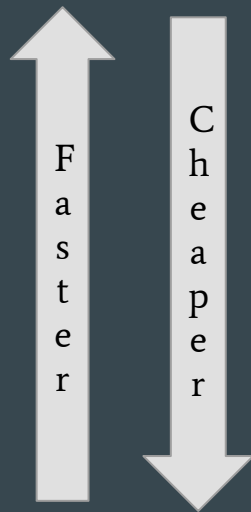
Many types of memory

- Flip-flops
- SRAM
- **DRAM**
- Flash
- Disk hard drive
- Tape



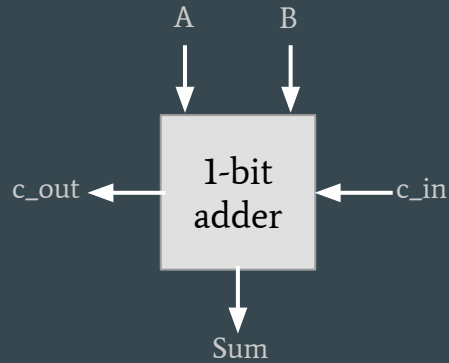
Many types of memory

- Flip-flops
- SRAM
- DRAM
- Flash
- Disk hard drive
- Tape

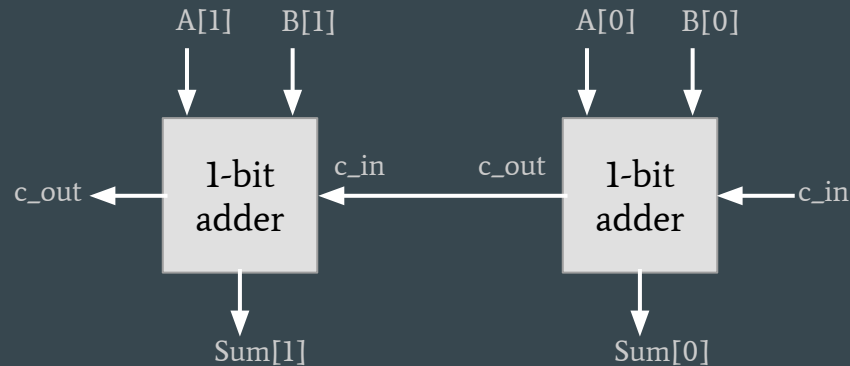


We can care about
some high level
conceptual matter.

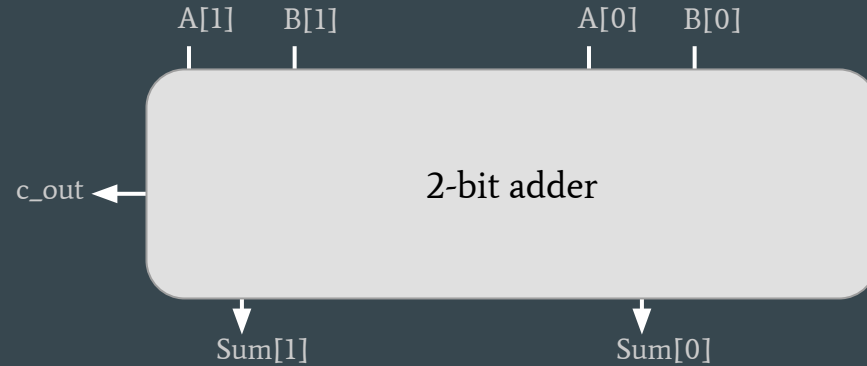
Let's build sequential logic: combinational+memory



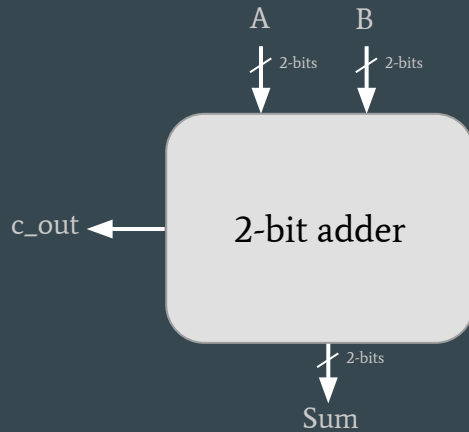
Let's build sequential logic: combinational+memory



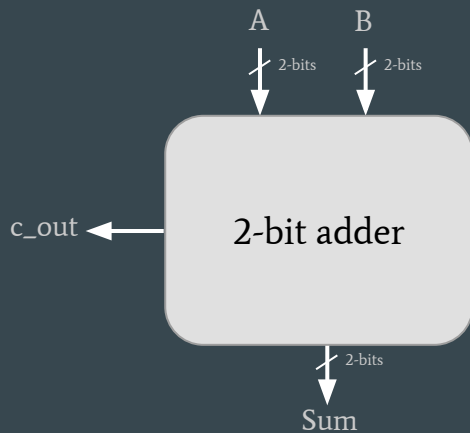
Let's build sequential logic: combinational+memory



Let's build sequential logic: combinational+memory

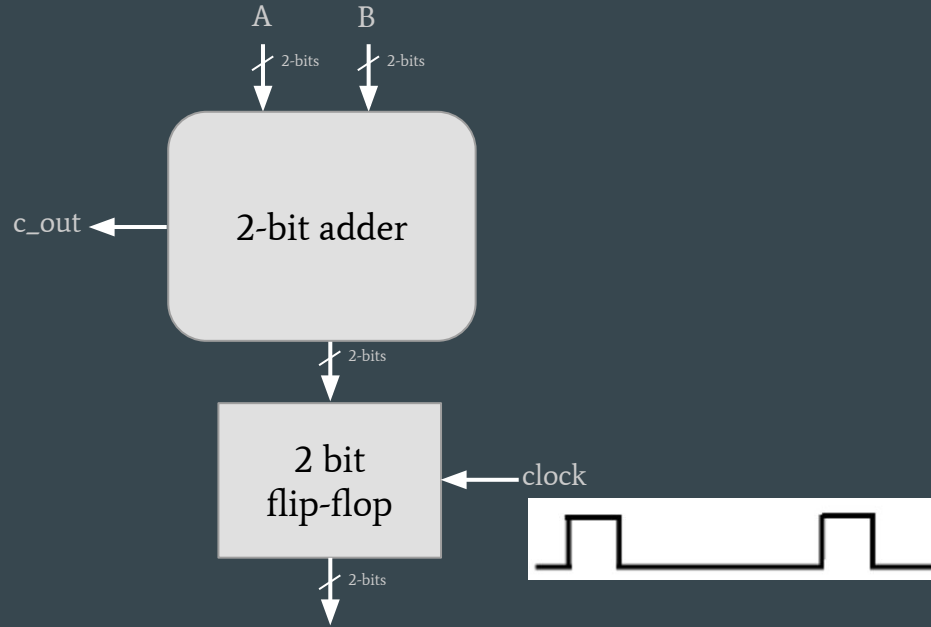


Let's build sequential logic: combinational+memory

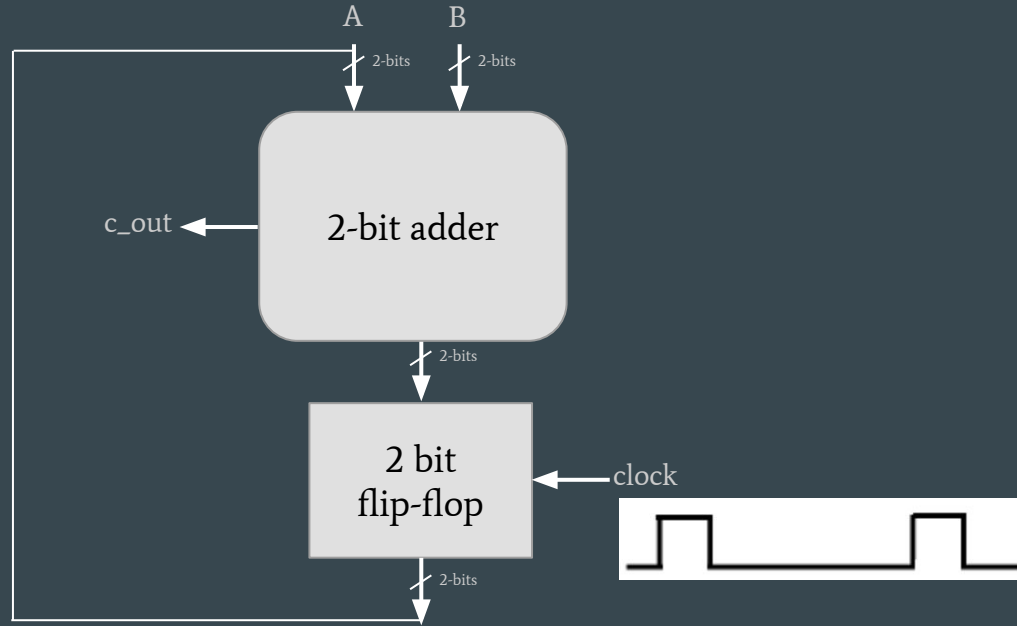


Still
combinational
logic!

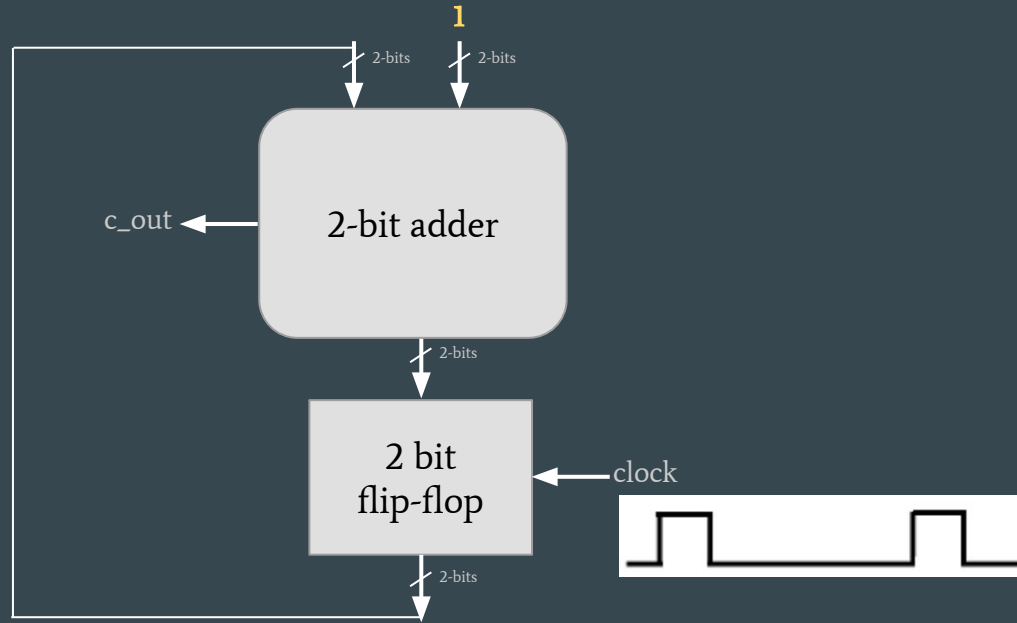
Let's build sequential logic: combinational+memory



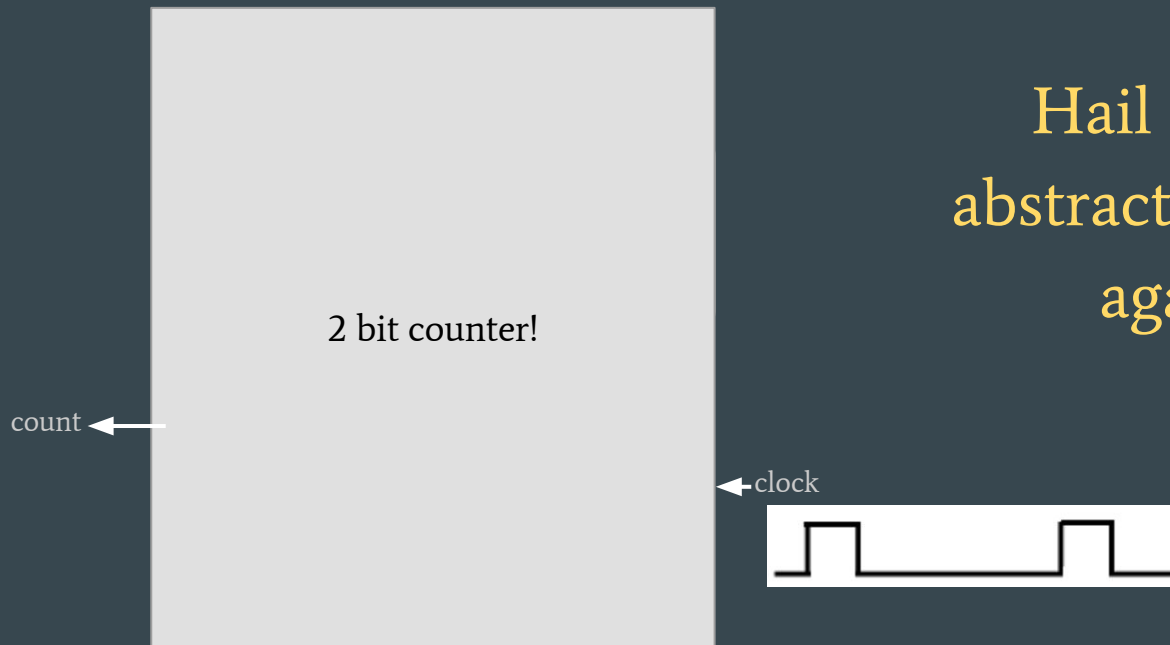
Let's build sequential logic: combinational+memory



Let's build sequential logic: combinational+memory



Let's build sequential logic: combinational+memory



Hail to the
abstraction once
again!