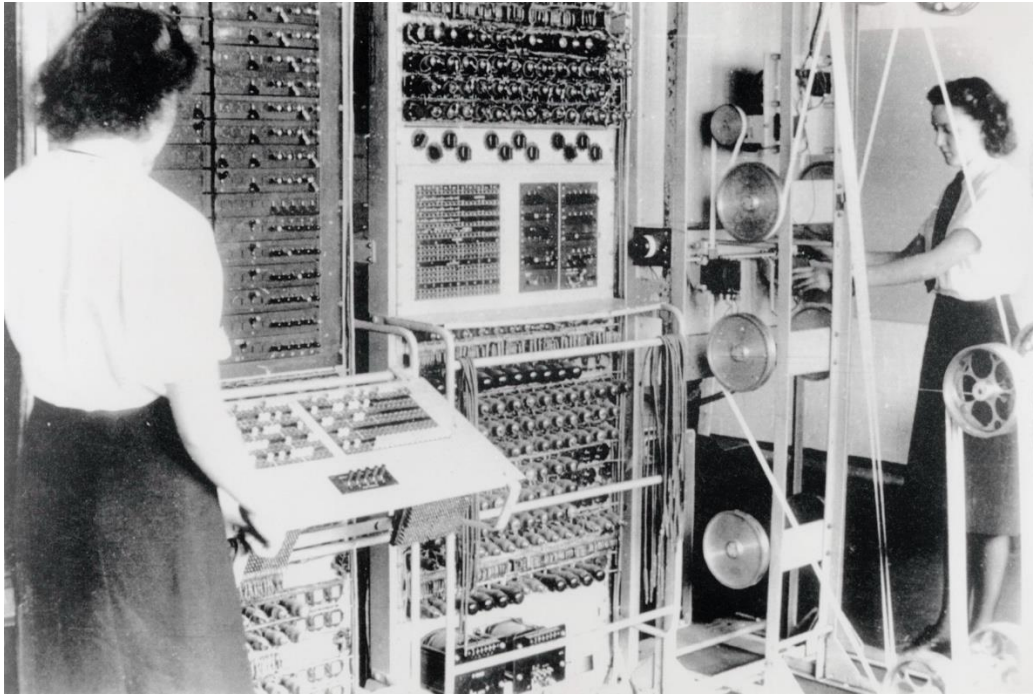


CSCI 210: Computer Architecture  
Lecture 19: Clocks, Latches, and Flip Flops

Stephen Checkoway  
Slides from Cynthia Taylor

# CS History: Latches

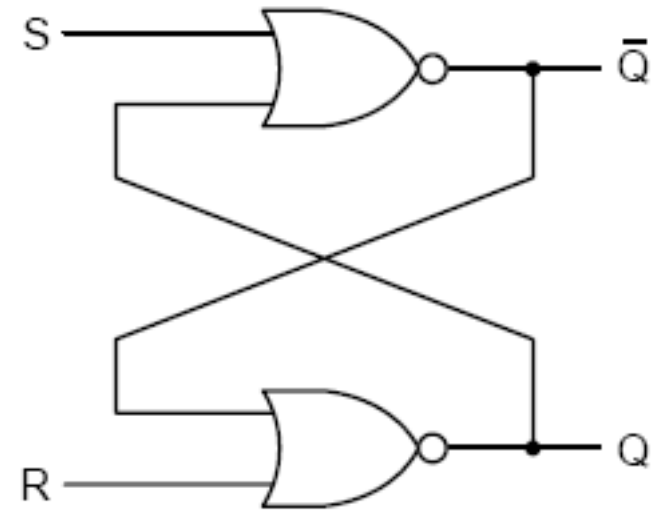


A Colossus Mark 2 codebreaking computer being operated by Dorothy Du Boisson (left) and Elsie Booker (right), 1943

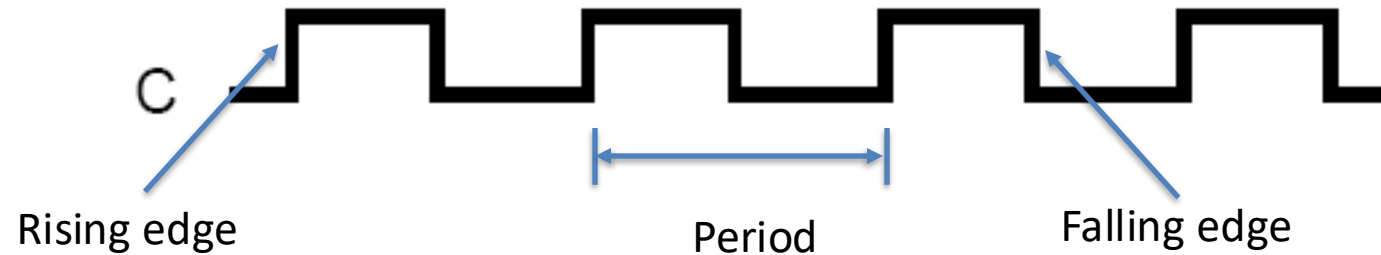
- The first electronic latch was invented in 1918 by British physicists William Eccles and F. W. Jordan.
- Used in the 1943 British Colossus codebreaking computer (made out of vacuum tubes)
- Modern flip flops made of logic gates were first discussed in a 1954 UCLA course on computer design by Montgomery Phister

# S-R Latch

- Set:  $Q_t = 1$        $\bar{Q}_t = 0$
- Reset:  $Q_t = 0$        $\bar{Q}_t = 1$
- Otherwise:  $Q_t = Q_{t-1}$        $\bar{Q}_t = \bar{Q}_{t-1}$



# Clock



- Oscillates between 1 and 0 with a fixed period
  - 0 to 1 transition is a **rising edge**
  - 1 to 0 transition is a **falling edge**
  - Time between two rising (or falling) edges is one **period** or **cycle**
- Used to control when values change

# Clocked SR Latch

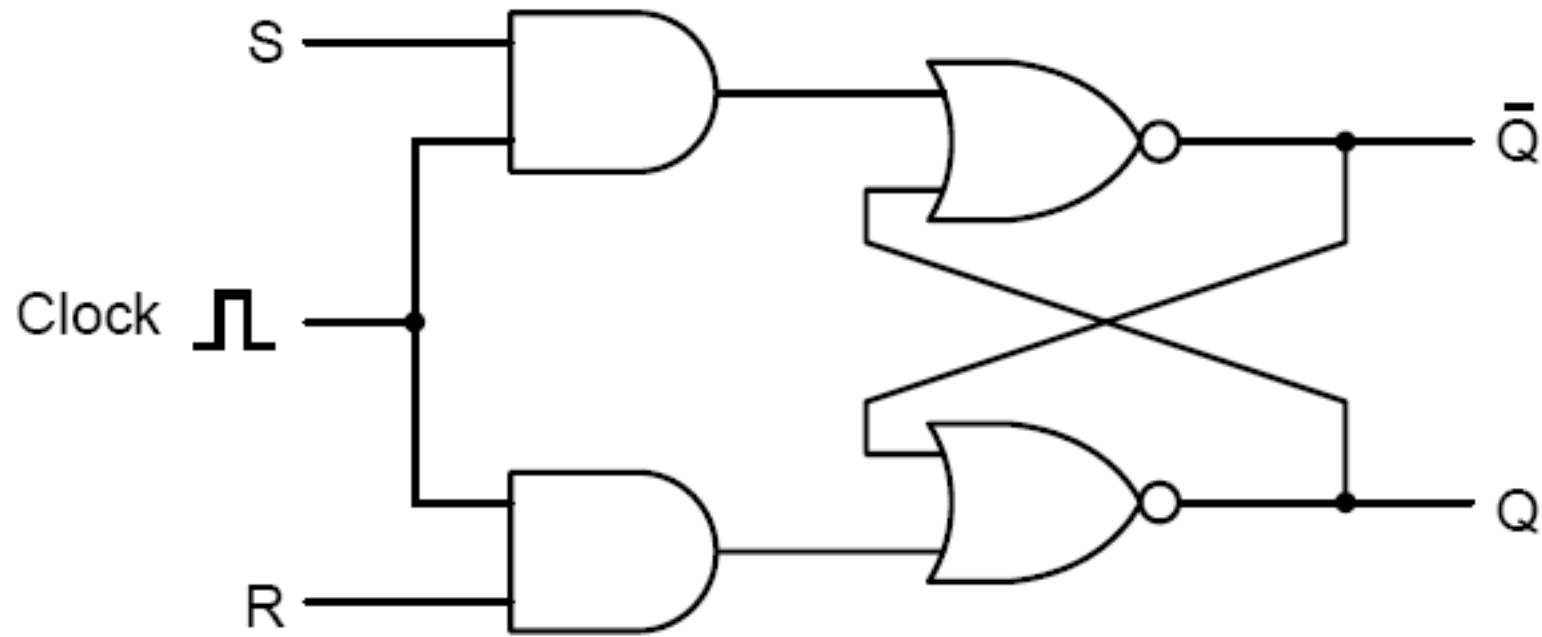
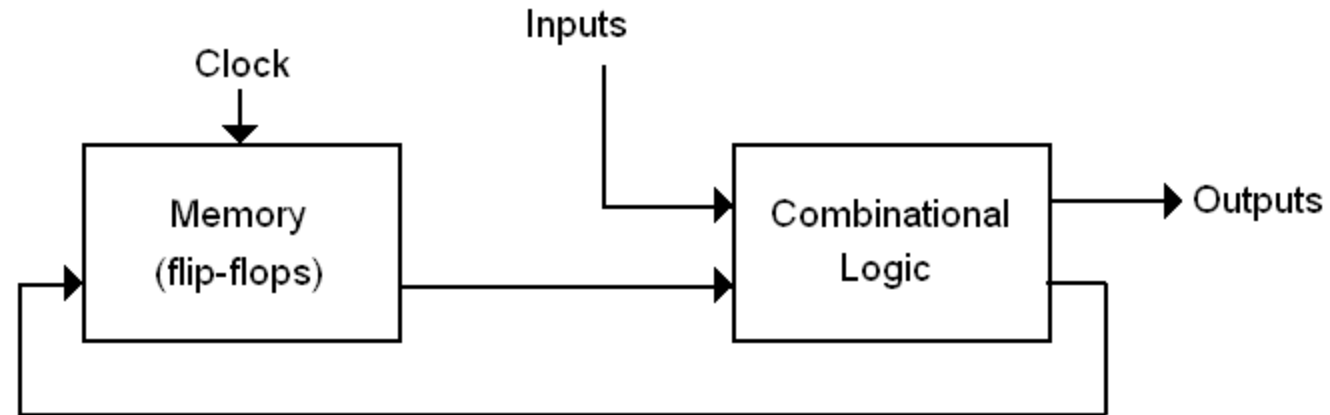


Figure 3-23. A clocked SR latch.

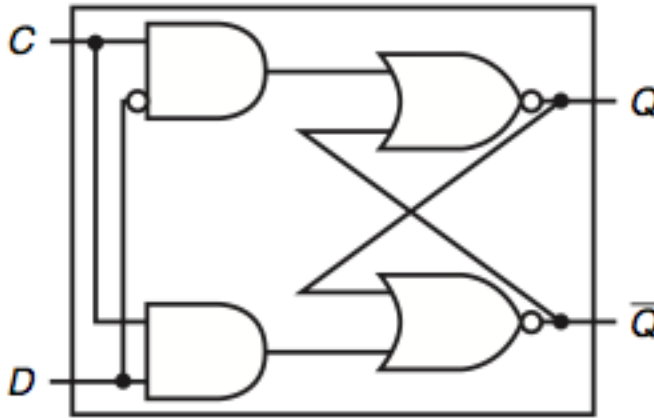
- Only changes state when the clock is asserted

# Why Clock a Latch?



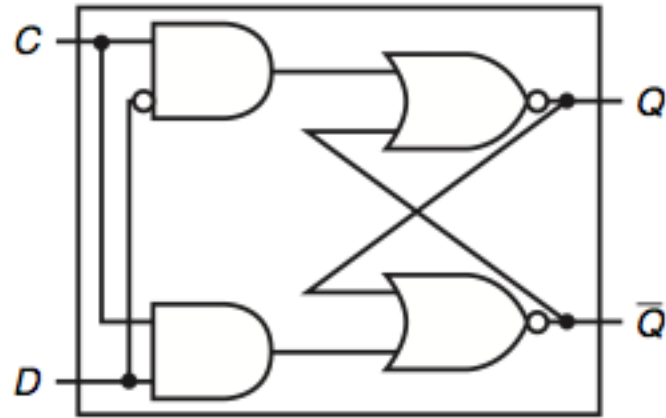
- Can save the results of combinational logic (think the ALU)
- If a latch is clocked, we know the values in it won't change as we perform combinational logic on them
  - Think of performing `addi $t0, $t0, 1`
- In today's class, we'll build memory that only changes values at one precise instant

# Clocked D-Latch



- S-R latch, but now there is a single input,  $D$ , ANDed with the clock  $C$
- Now impossible to have both inputs set to 1

# Which Column Completes the Truth Table?



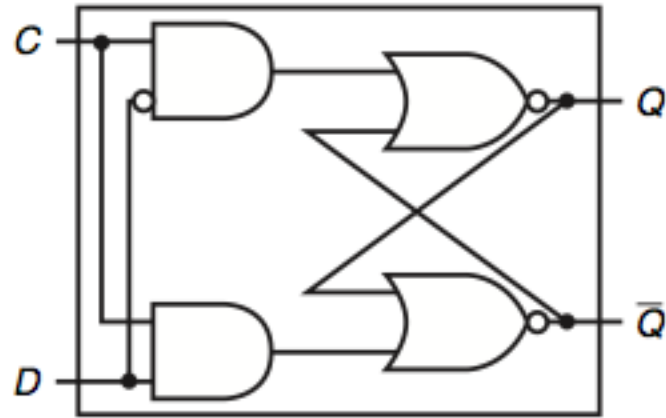
C	D	Q
1	1	
1	0	

A	B	C	D
1	1	0	1
1	0	1	$Q_{\text{prev}}$

E. None of the above



# Which Column Completes the Truth Table?

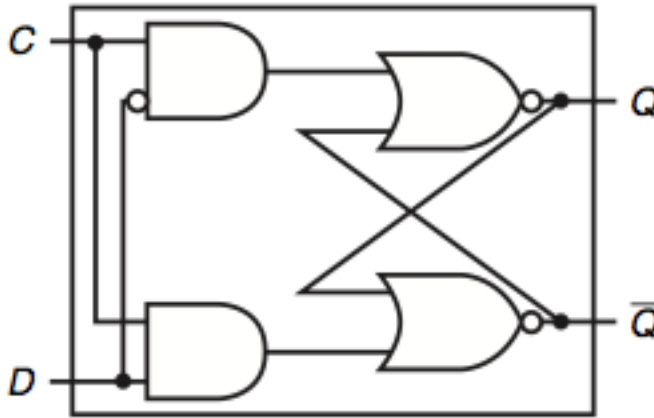


C	D	Q
0	1	
0	0	

A	B	C	D
0	1	1	$Q_{\text{prev}}$
0	0	$Q_{\text{prev}}$	$Q_{\text{prev}}$

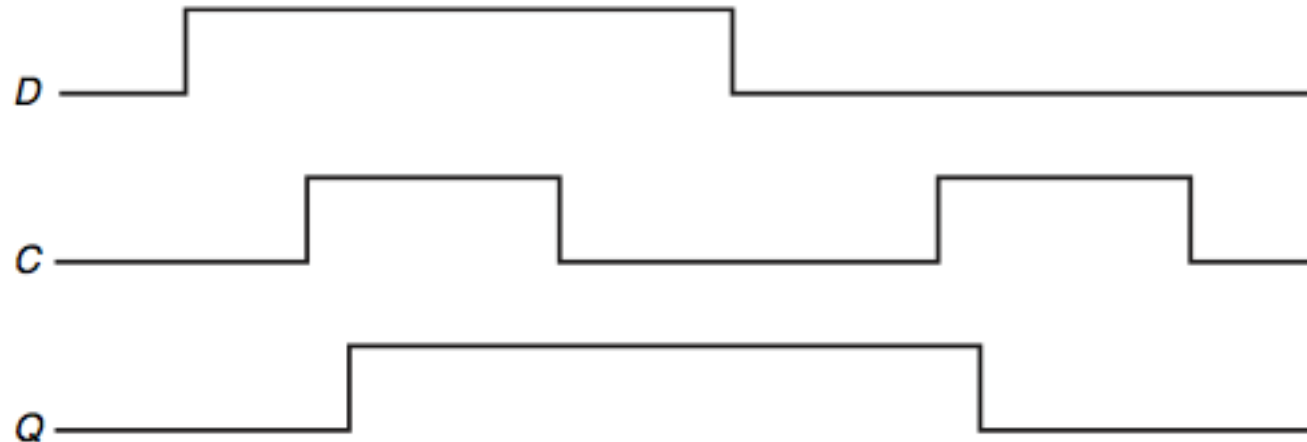
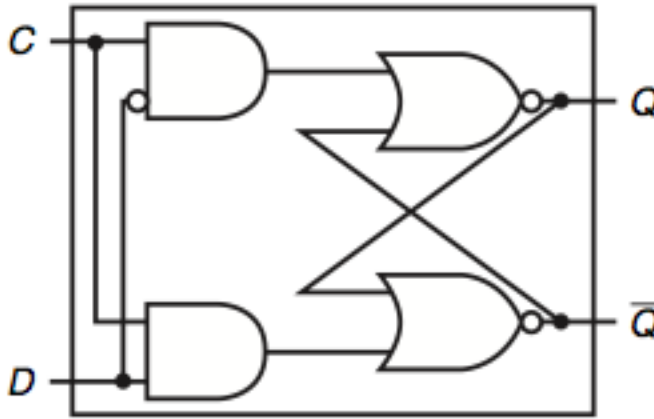
E. None of the above

# Clocked D-Latch

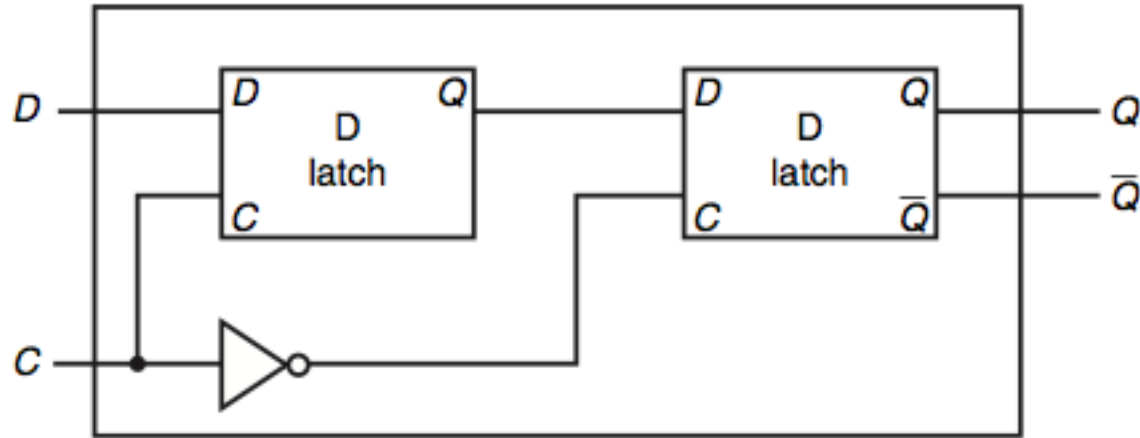


- Latch is “open” when clock is asserted (asserted = logical 1)
- $Q = \text{value of } D \text{ when the latch is open}$   
 $Q = \text{most recently set value when the latch is closed}$

Clocked D-Latch; note output takes a little time to change after the clock goes high



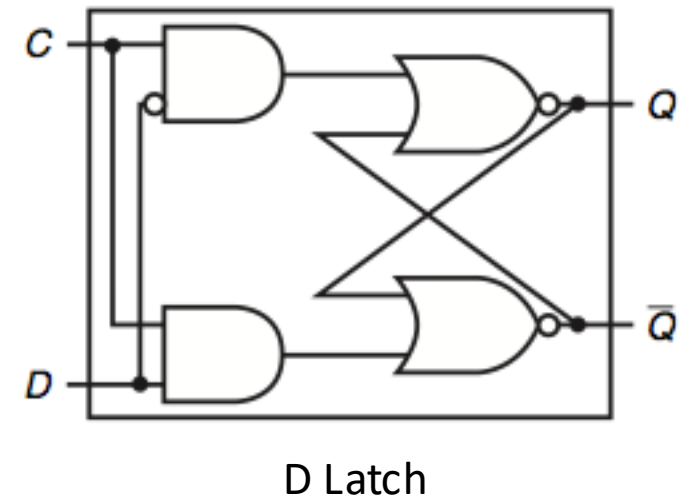
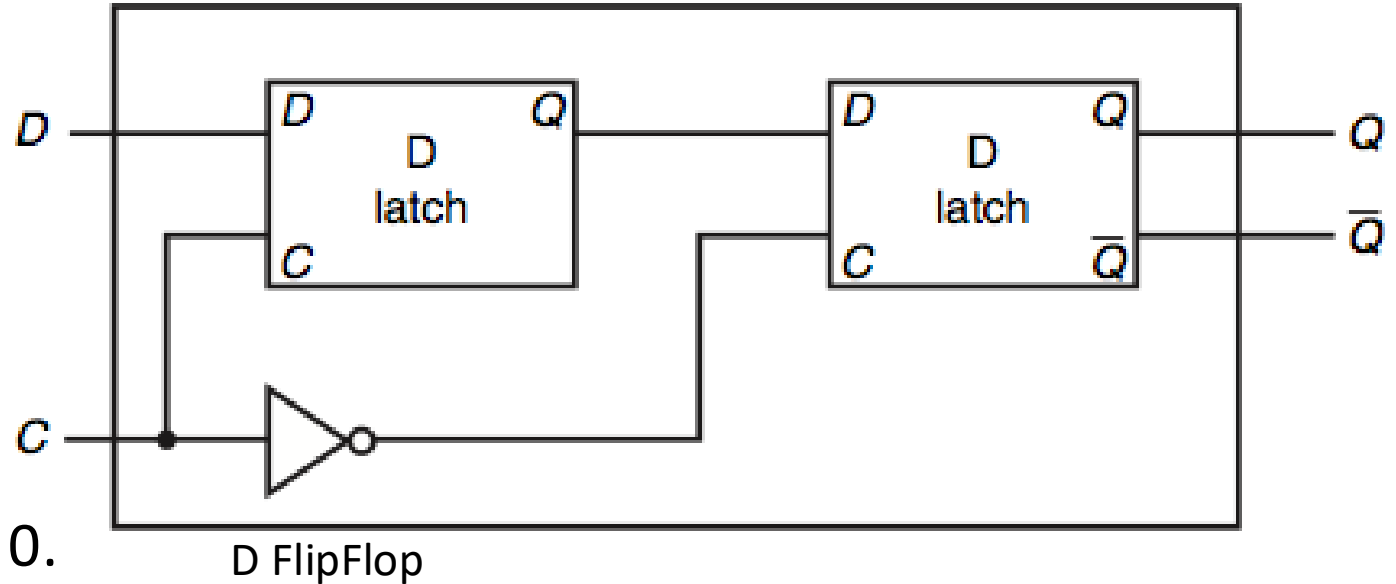
# D Flip-Flop



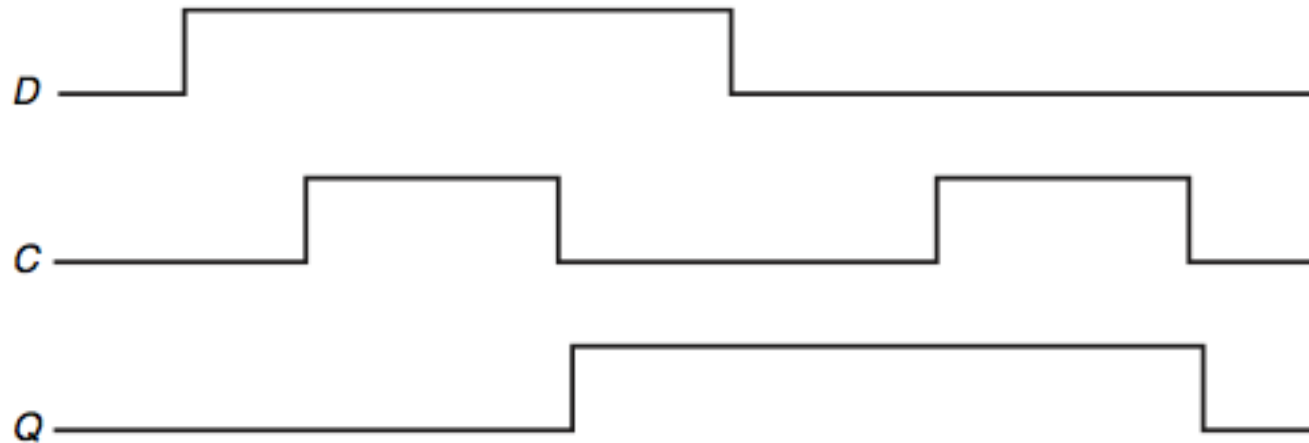
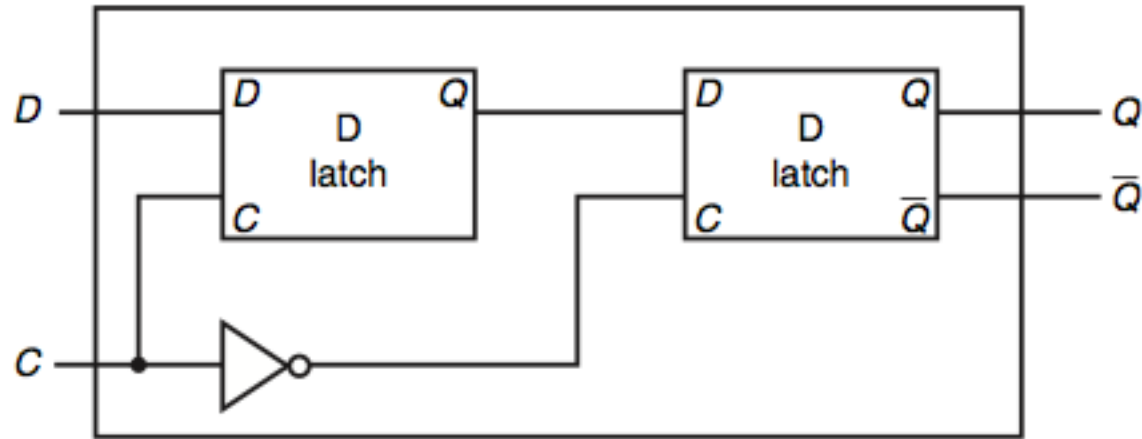
- Two D-Latches, with the clock negated to the second latch

The value of (the right-most) Q in the flip flop can change

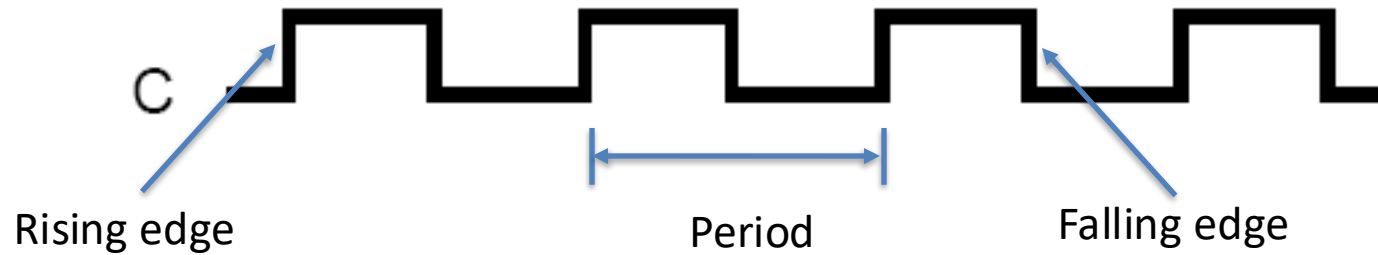
- A. Any time the clock is 1.
- B. Any time the clock is 0.
- C. When the clock changes from 1 to 0.
- D. When the clock changes from 0 to 1.
- E. None of the above



# D flip flop: Falling Edge Trigger

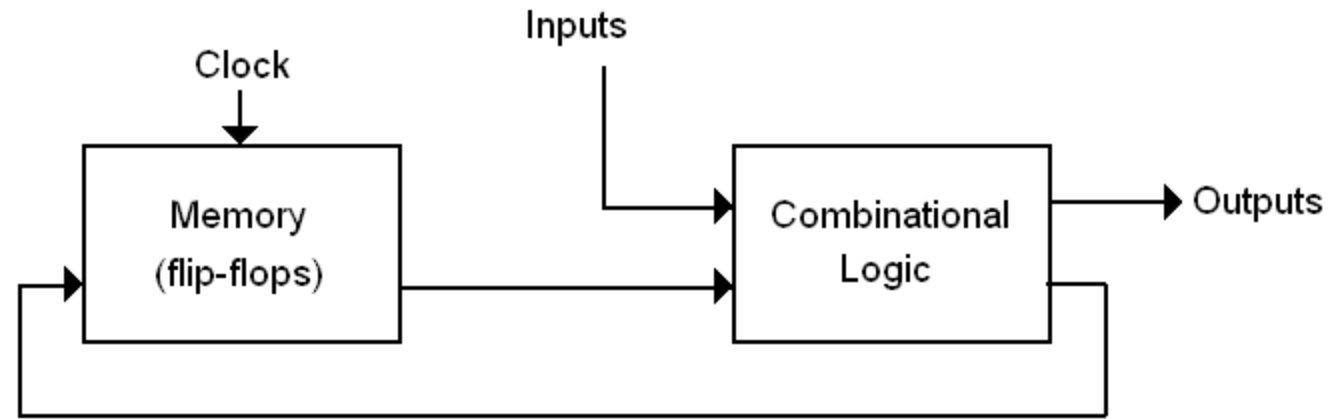


# Edge-triggering



- All changes to state happen at one point in the clock cycle (either rising edge, or falling edge).

# Memory

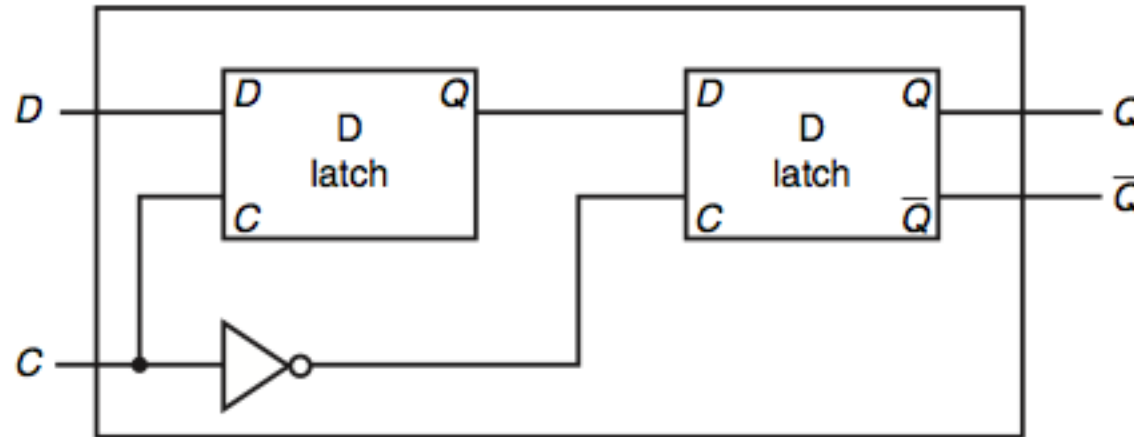


- Can save the results of combinational logic (think the ALU)
- Registers are (multi-bit) flip-flops!

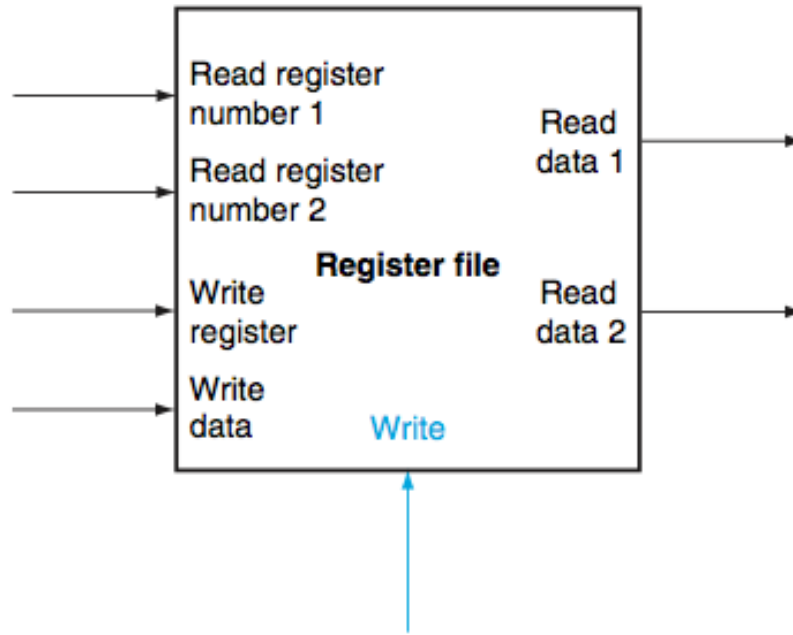


# Registers

- Each 32-bit register will consist of 32 1-bit D flip-flops

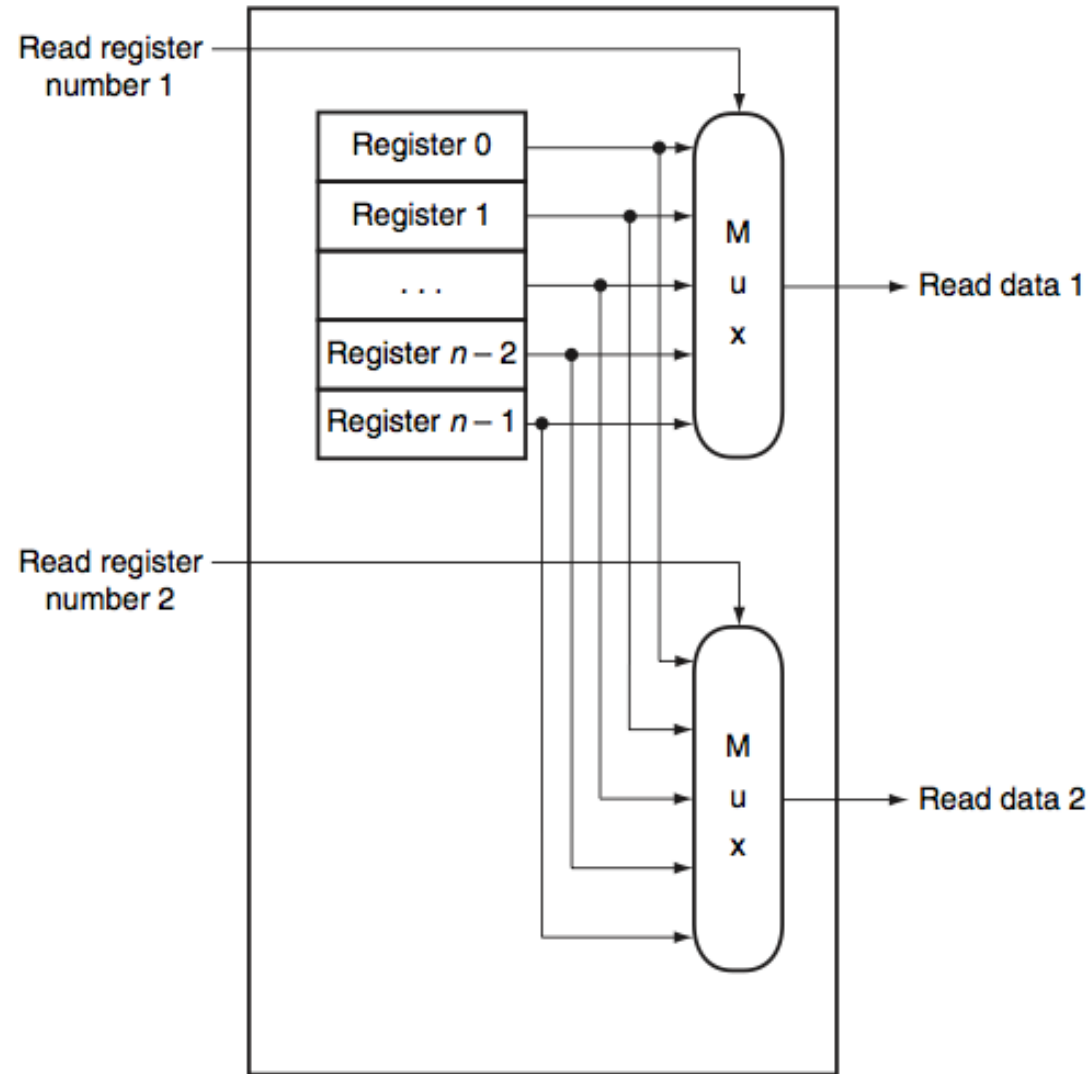


# Register File

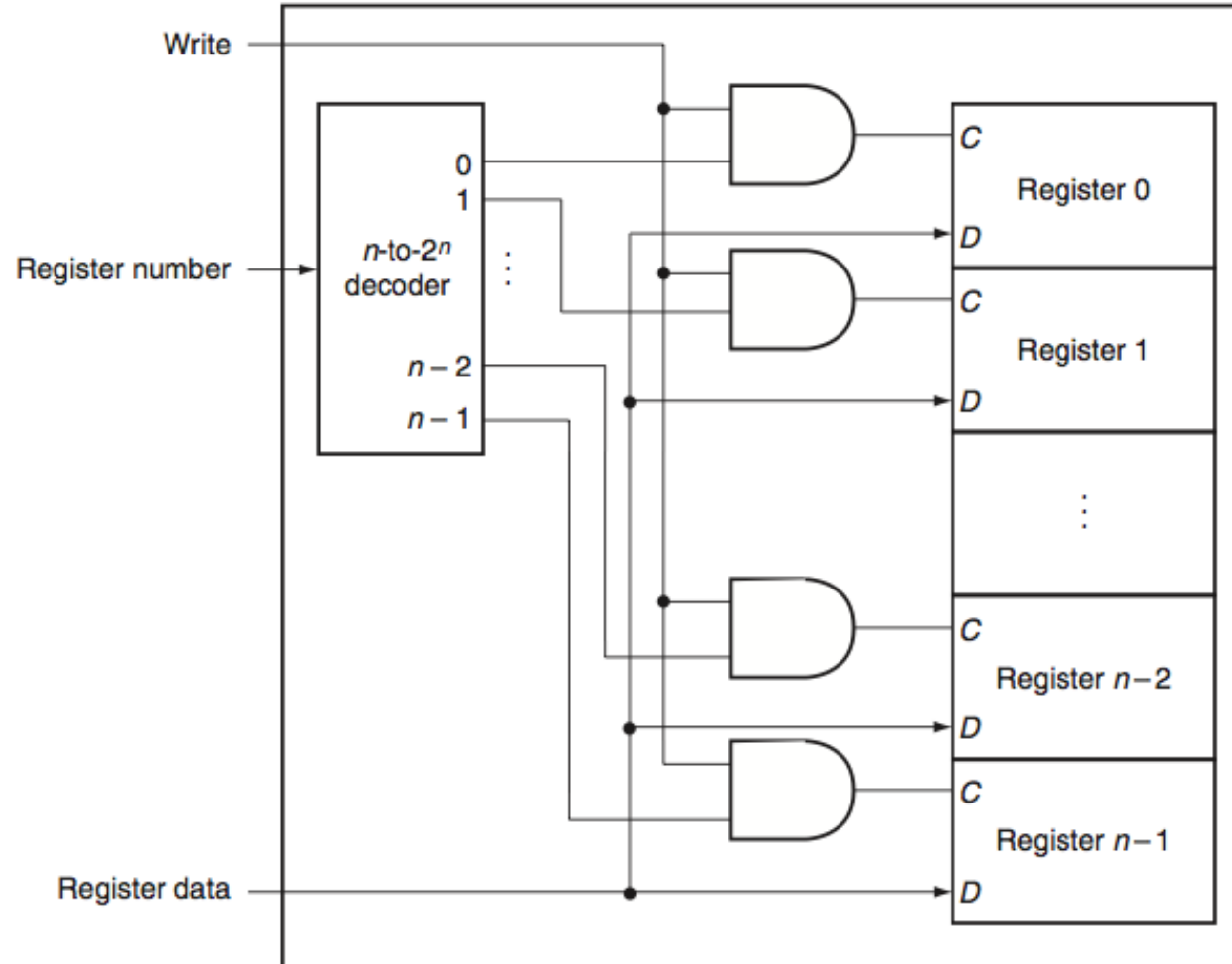


- Set of registers that can be written/read by supplying a register number
- MIPS has a register file with thirty-two 32-bit registers

# Read two register values



# Write one register value

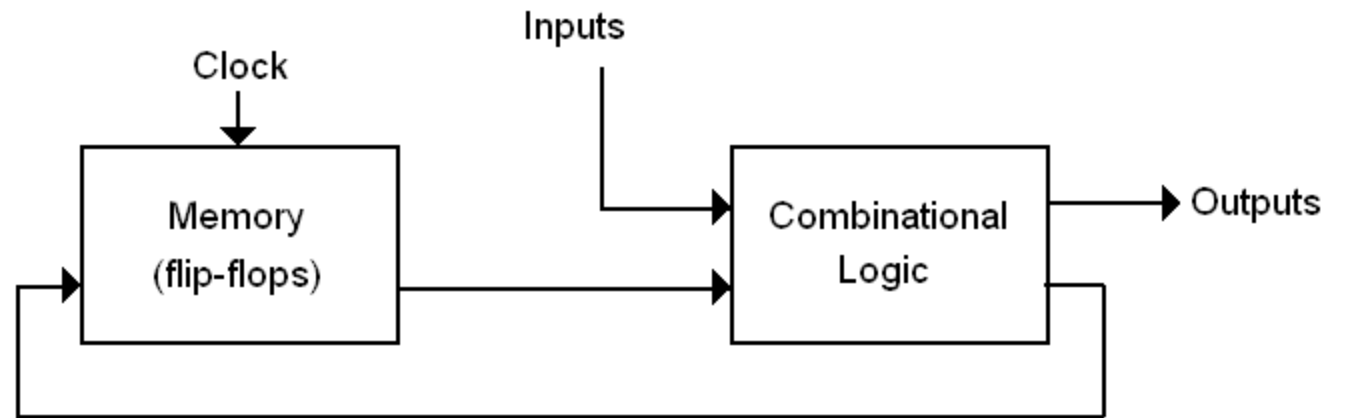


\*The image is not quite correct. It should be a  $\lg n$  to  $n$  decoder

In MIPS, we have 32 registers so we need a 5-to-32 decoder, not a 32-to-4294967296 decoder!

What will happen if we read and write to a register in the same clock cycle, as in `add $t0, $t0, $s0`

- A. The read will get the original value
- B. The read will get the just written value
- C. It is ambiguous
- D. None of the above

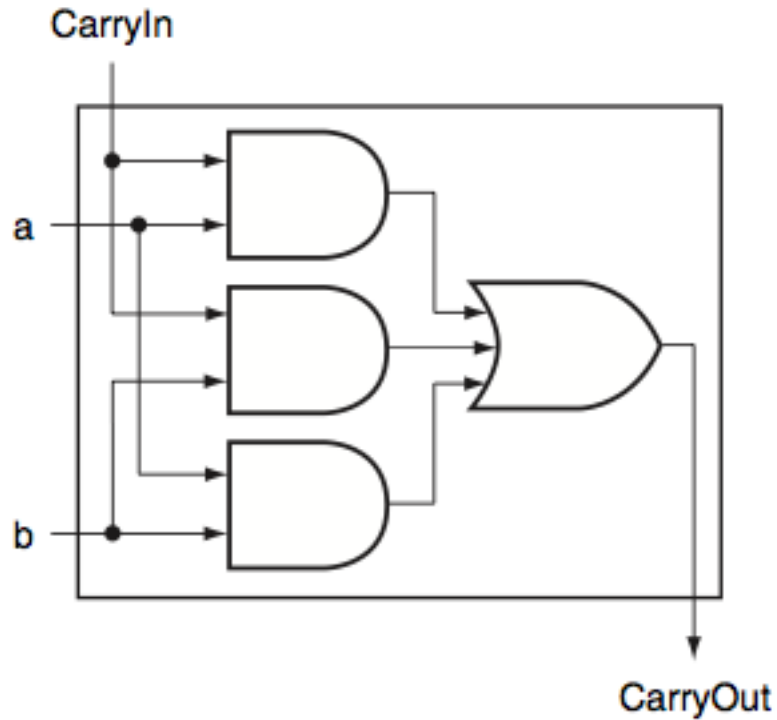


# Register Questions

# Speed of Combinational Circuits

- Assume each gate takes a certain amount of time for the signal to pass through.
- *Gate Delay* is measured by counting the number of gates along a path.
- Note that all non-sequential gates operate in parallel

What is the gate delay to calculate carry-out with this circuit?



A. 1

B. 2

C. 3

D. 4

E. None of the above



# Minimum clock cycle length

- Minimum clock cycle length is determined by
  - The gate delay of the combinational logic
  - The propagation delay of the flip flop (how long does it take for the output to appear after the flip flop's state is changed)
  - The setup time for the flip flop (how long does the value have to be stable before the falling edge of the clock (for falling edge-triggered flip flops))