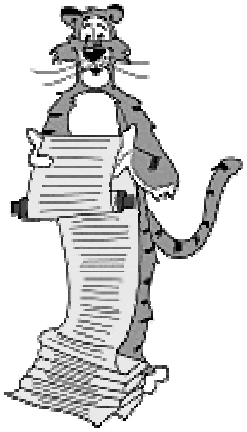
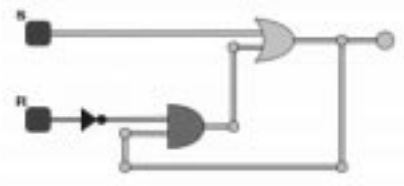
# Lecture A4: Sequential Circuits





## Architecture

**Lecture A1 – A2: TOY machine.**

**Lecture A3: Boolean logic and combinational circuits.**

- In principle, we could build TOY computer with one gigantic combinational circuit.
- Each circuit element used (at most) once.

**Today.**

- How to reuse circuit elements.
- How to store bits in "memory."

**Next time.**

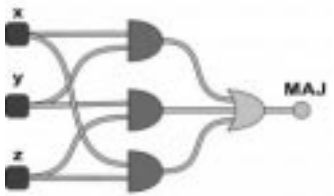
- Glue these components together to make TOY computer.



## Sequential vs. Combinational Circuits

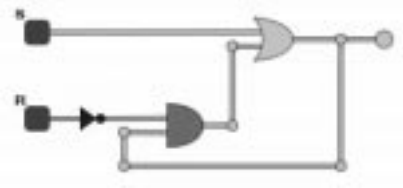
**Combinational circuits.**

- Output determined solely by inputs.




**Sequential circuits.**

- Feedback loop.
- Output determined by inputs and previous outputs.





## Flip-Flop

**Flip-flop.**

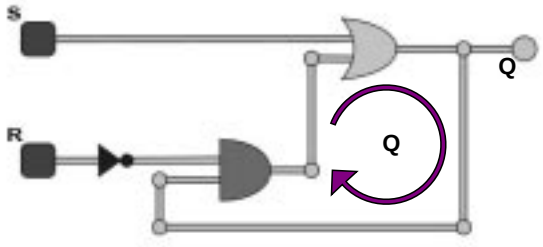
- A small and useful sequential circuit.
- "Remembers" one bit.

**We will consider many flavors.**

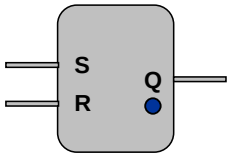
# SR Flip-Flop

**SR Flip-Flop.**

- Pulse on S (set) ⇒ Flips "bit" on.
- Pulse on R (reset) ⇒ Flips "bit" off.
- S = R = 0 ⇒ Status quo.
- S = R = 1 ⇒ Not allowed.




Implementation




Interface



# Truth Table and Timing Diagram (for SR Flip-Flop)

**Truth table.**

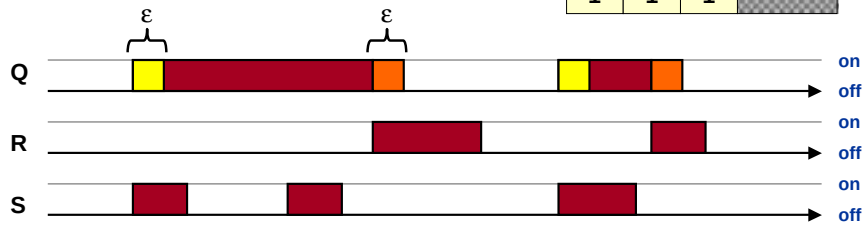
- Values vary over time.
- S(t), R(t), Q(t) denote value at time t.

| SR Flip Flop Truth Table | | | |
|---|---|---|---|
| S(t) | R(t) | Q(t) | Q(t+ε) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Characteristic equation.**

$$Q(t+\varepsilon) = S(t) + R'(t)Q(t) \qquad (SR = 0)$$

**Sample timing diagram.**

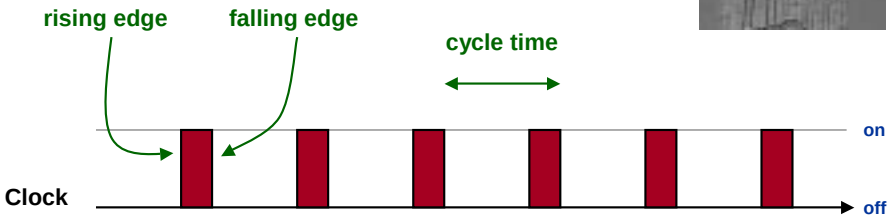





# Clock

**Clock.**

- Fundamental abstraction.
  - regular on-off pulse
- External analog device.
- Synchronize operations of different circuit elements.
- 800 MHz clock means 800 million pulses per second.

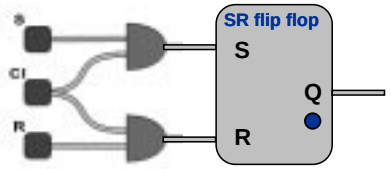




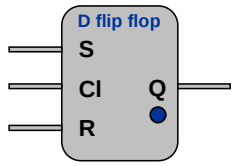

# Clocked SR Flip-Flop

**Clocked SR Flip-Flop.**

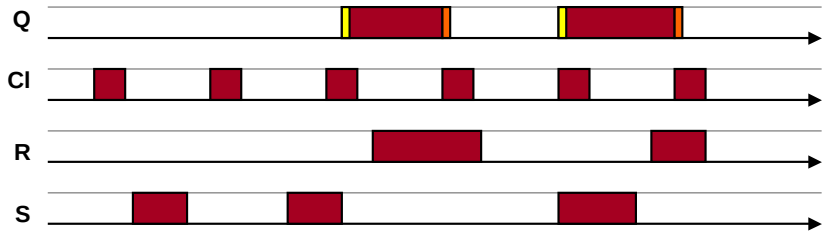
- Like SR flip-flop but S and R only work if clock is on.




Implementation



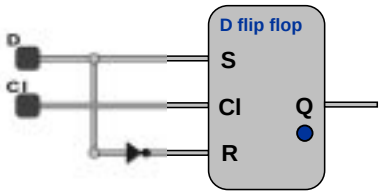

Interface



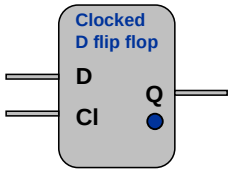
# Clocked D Flip-Flop

**Clocked D Flip-Flop.**

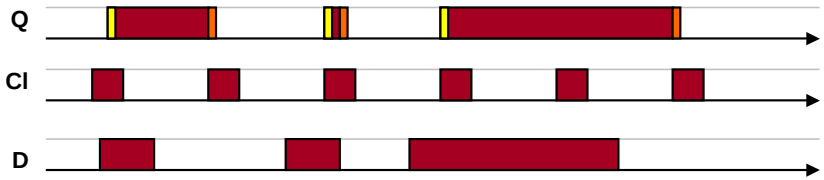
- On clock pulse: if D = 1, then set; if D = 0, then reset.




Implementation




Interface

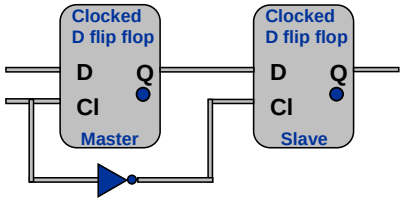


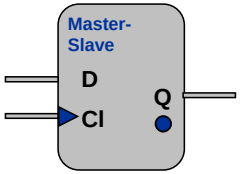



# Master Slave Flip-Flop

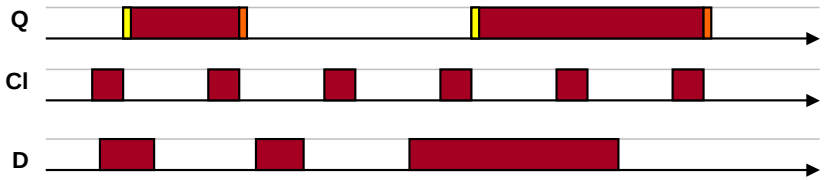
**Master-slave flip-flop (falling edge-trigger).**

- Input can only change on falling edge.




Interface






# Computer Architecture Perspective

**Circuits needed to build a computer.**

- Combinational circuit components.
  - adder, multiplexer, decoder
- Sequential circuit components (build from flip-flops).
  - counter
  - memory

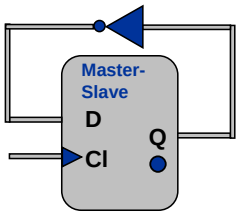
**All are built from AND, OR, NOT gates.**



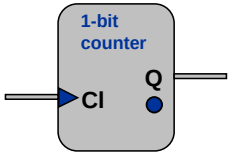
# 1-Bit Counter

**1-bit counter.**

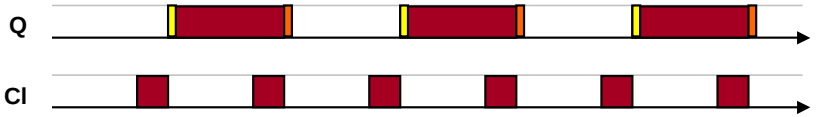
- "Clock" whose cycle is twice as long as input.




Implementation



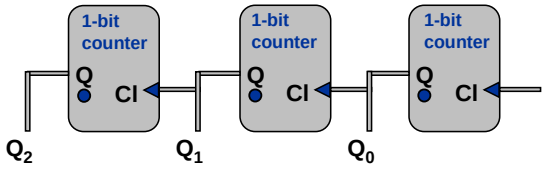

Interface

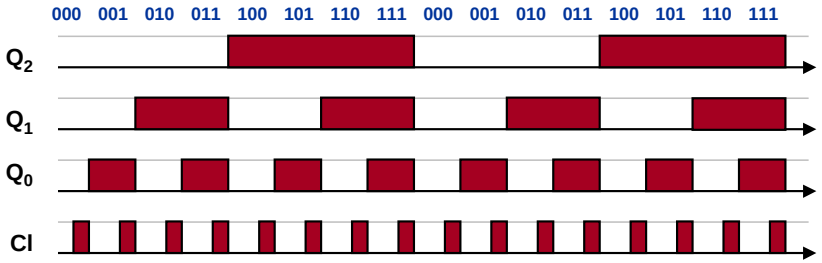


# N-Bit Counter

**N-bit counter.**

- Chain N 1-bit counters together.






# Memory Overview

**Computers have many types of memory.**

- Registers.
- Main memory.

**Master-slave flip-flop implements 1 bit of memory.**

**Need mechanism to reference, store, and extract individual bits.**

- Multiplexer, decoder.

**Bit-slice memory.**

- Word size in TOY is 16 bits.
- First: design circuit for memory with 1 bit "words."
- Then: implement 16-bit word memory with 16 copies.



# Register File: 8 1-bit "words"

**Register file: $n = 2^t$ bits.**

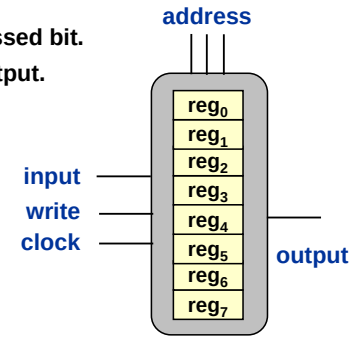
- n bits of memory.
- Address specifies which bit.
  - How many bits needed to specify address?
- If write = 1, input gets copied into addressed bit.
- If write = 0, addressed bit appears on output.

**TOY registers.**

- 8 16-bit words.
- Need 16 copies.

**TOY main memory.**

- 256 16-bit words.
- Need 16 copies of register file with 256 1-bit words.



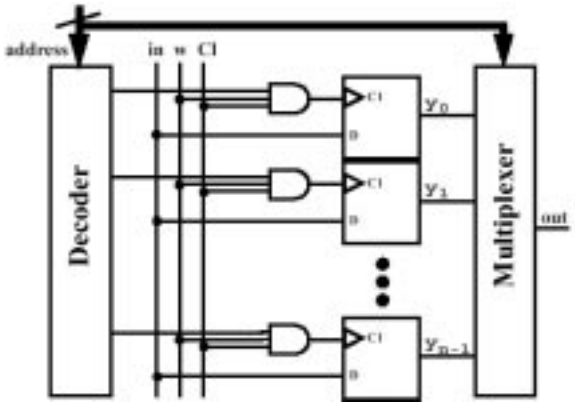

Interface
(8 1-bit "words")



# Register File: n 1-bit "words"

**Register file: n registers (words), 1 bit per register.**

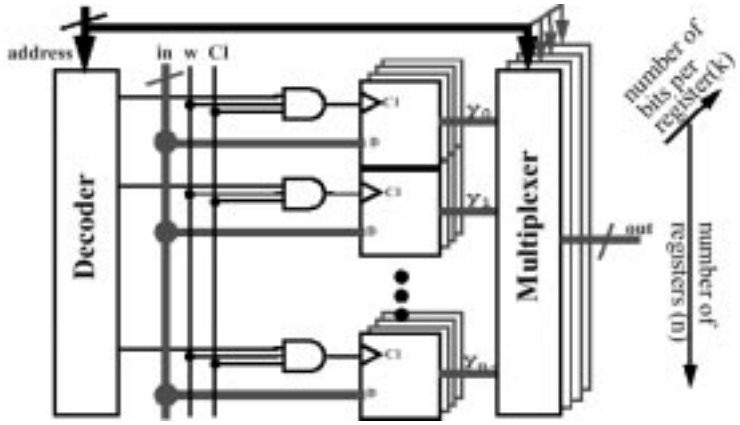
- Decoder writes input to address bit.
- Multiplexer copies address bit to output.



# Register File: n k-bit words

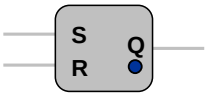
**Register file: n registers (words), k bits per register.**

- **k copies of single bit register file (k = 16 for TOY).**

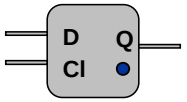




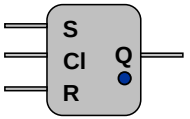

# Cheat Sheet



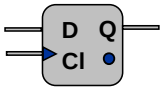

SR flip-flop



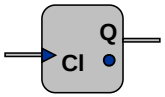

Clocked D flip-flop



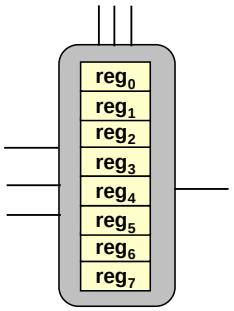

Clocked SR flip-flop



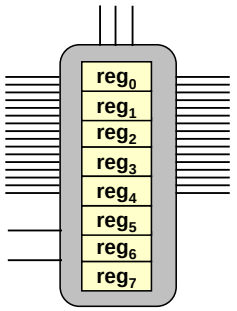

Master-slave flip-flop




1-bit counter




Register File (8 bits)




Register File (8 16-bit words)