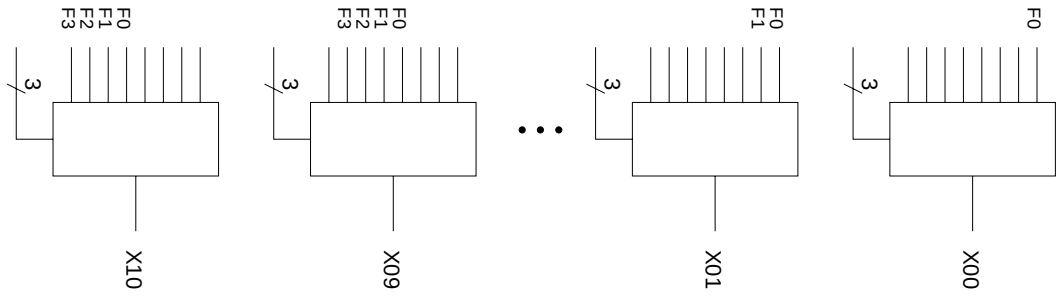
FPCVT in software

```
            FRAC <<= EXP;
            if (SIGN == 1) {
              FRAC = -FRAC;  /* 12-bit two's complement */
            }
or
            for (i = 0; i < EXP; i++) {
              FRAC <<= 1;  /* multiply by 2 */
            }

or
            switch (EXP) {
            case 1:  FRAC <<= 1; break;
            case 2:  FRAC <<= 2; break;
            ...
            case 7:  FRAC <<= 7; break;
            }
```


FPCVT in hardware

Possible values of the fixed point output are shifts of the input fraction.

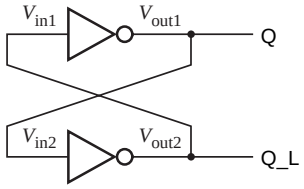


The multiplexer select signals control the amount of the shift (0 to 7).

Bistable elements

Sequential circuits require memory. Possible storage devices:

- mechanical position (requires solenoids or motors to change)
- magnetic charge
- charge on capacitors (may require sense amplifiers)
- feedback loops in logic circuit:



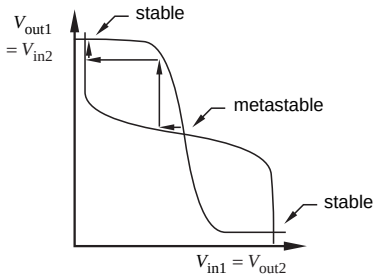
There are two solutions to system of equations, $\mathsf{Q\_L} = \mathsf{Q}', \mathsf{Q} = \mathsf{Q\_L}'$:

$$\mathsf{Q\_L} = 0,\ \mathsf{Q}' = 1 \quad \text{or} \quad \mathsf{Q\_L} = 1,\ \mathsf{Q}' = 0$$

State can be changed by setting input to a value for $\geq$ two gate delays.

Metastability

There are *metastable* states between stable states (continuity argument).

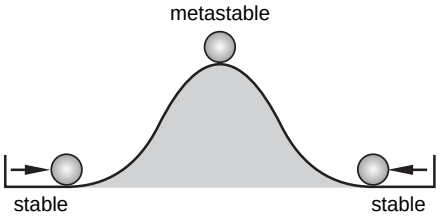


Transfer function:

$V_{out1} = T(V_{in1})$

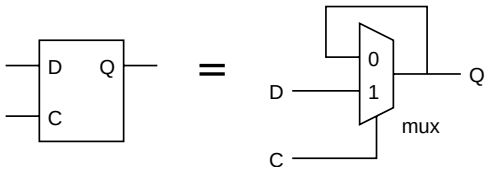
$V_{out2} = T(V_{in2})$

Changing state of a bistable requires energy and time. Insufficient driving force results in metastable state (or oscillation).

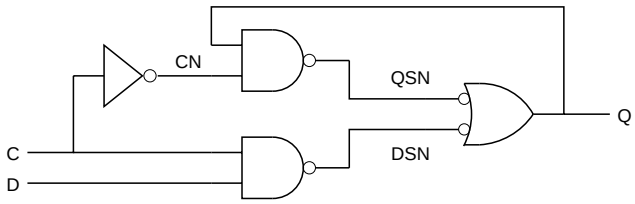


D latch

Simplest memory device is D latch: if C then D else Q



Implementation using multiplexer built from NAND gates:

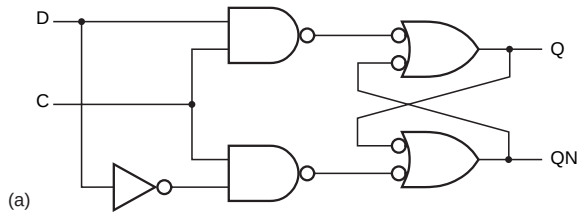


The multiplexer circuit has a hazard, which can cause failure.

Moral: let experts design the basic memory cells (latches and flip-flops).

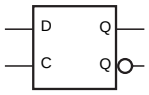
D latch: behavior and timing

Representative circuit and function table:



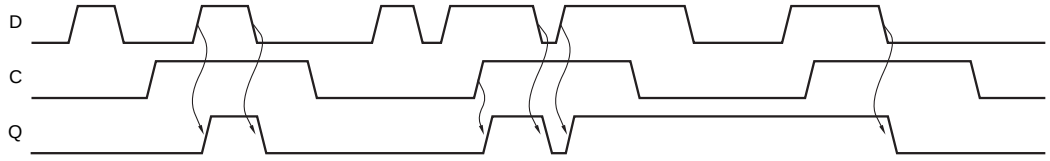
C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN

(b)



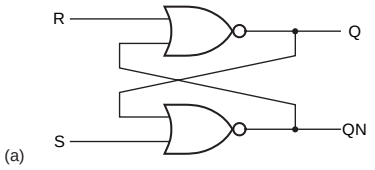
(c)

Functional behavior for various inputs:



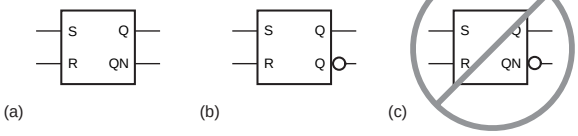
D latch precursors: S-R latch

S-R latch has two control inputs, S = set, R = reset.

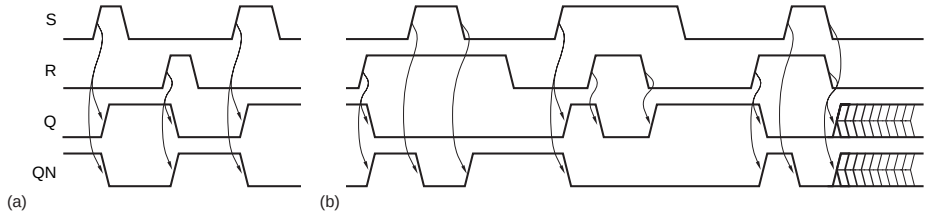


S	R	Q	QN
0	0	last Q	last QN
0	1	0	1
1	0	1	0
1	1	0	0

(b)

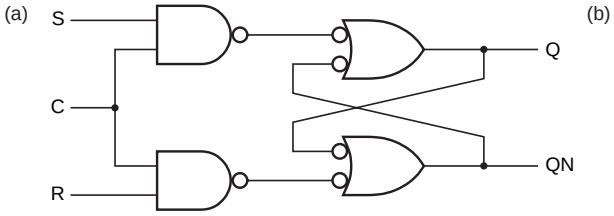


Set and reset signals normally should not be asserted simultaneously.



D latch precursors: S-R latch with enable

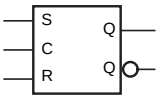
Control signals R and S are ignored when C is false.



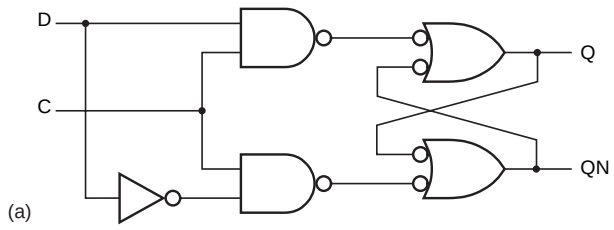
(b)

S	R	C	Q	QN
0	0	1	last Q	last QN
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1
x	x	0	last Q	last QN

(c)

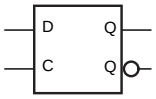


Replace R and S by D′ and D to obtain D latch.



C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN

(b)



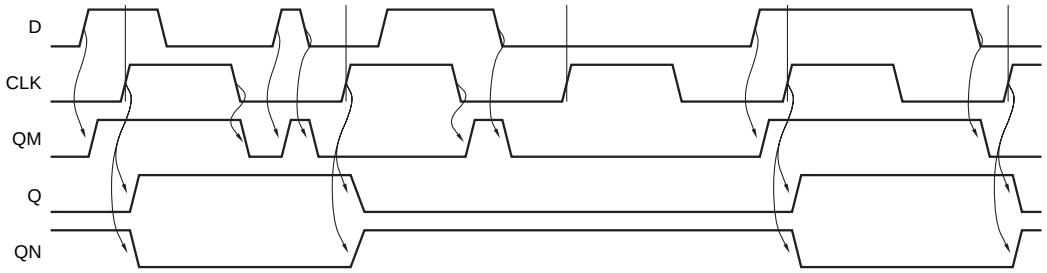
(c)

Flip-flops

D latches are *transparent*: D = Q while C = 1.

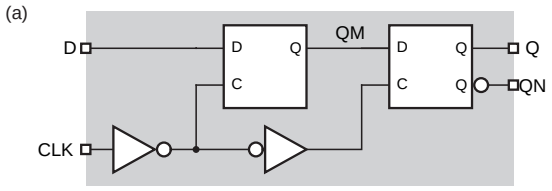
- Advantage: input changes can be seen immediately at output
- Disadvantage: output changes when input changes (while C = 1).

Desired behavior for a memory device: stored value changes only at discrete time instants, determined by rising or falling edges of *clock* signal.



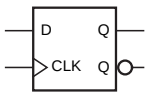
D flip-flop: implementation

One circuit design for a D flip-flop uses two D latches.

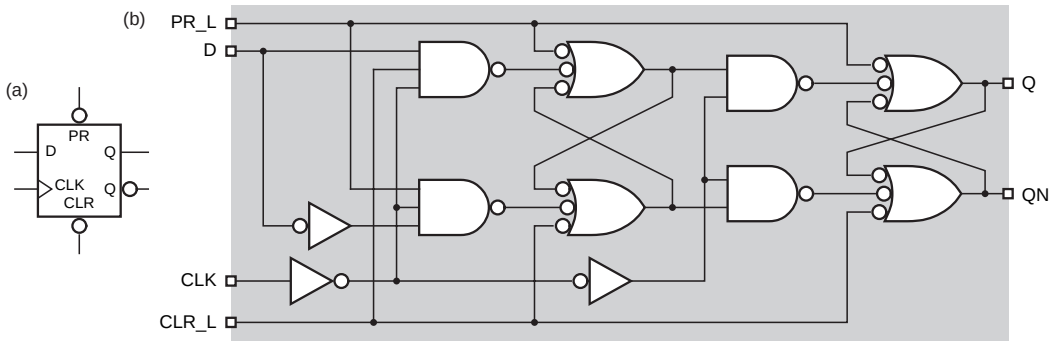


(b)

D	CLK	Q	QN
0	↑	0	1
1	↑	1	0
x	0	last Q	last QN
x	1	last Q	last QN

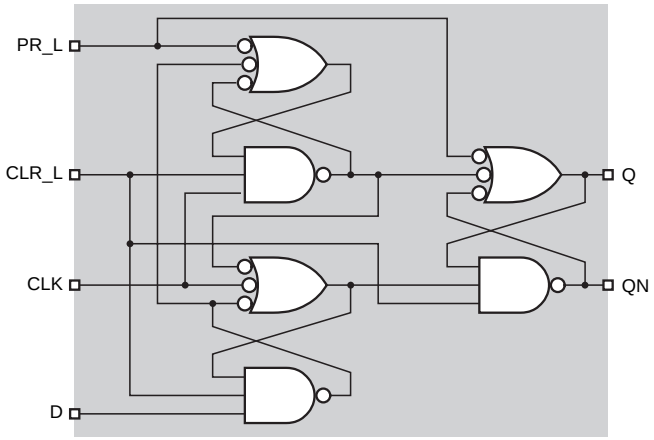


Straightforward implementation of master-slave flip-flop:



D flip-flop: implementation (2)

Representative circuit for TTL flip-flop (74x74):

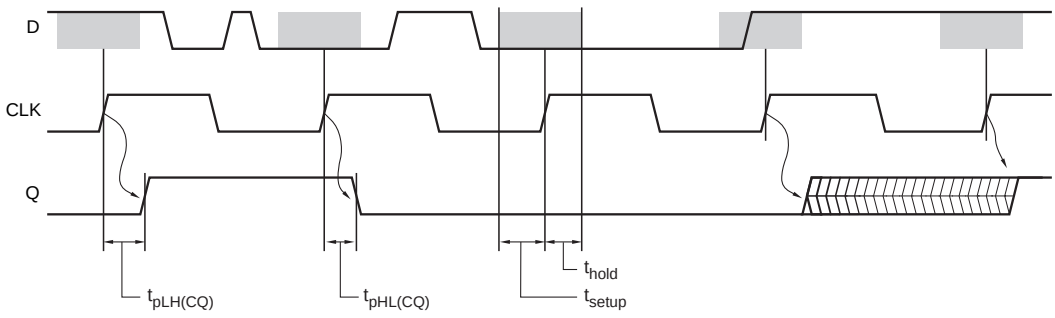


Flip-flops built directly from CMOS transistors will use clever methods to reduce size and delay and improve reliability.

D flip-flop: logic model and timing

Correct behavior is guaranteed if data is stable during a window surrounding the active clock edge:

- t_{setup} = time before clock edge
- t_{hold} = time after clock edge

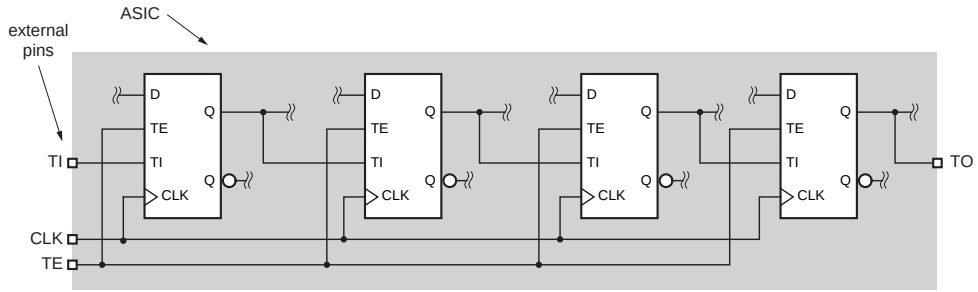


Window duration is $t_{setup} + t_{hold}$ depends on the technology and gate delays.

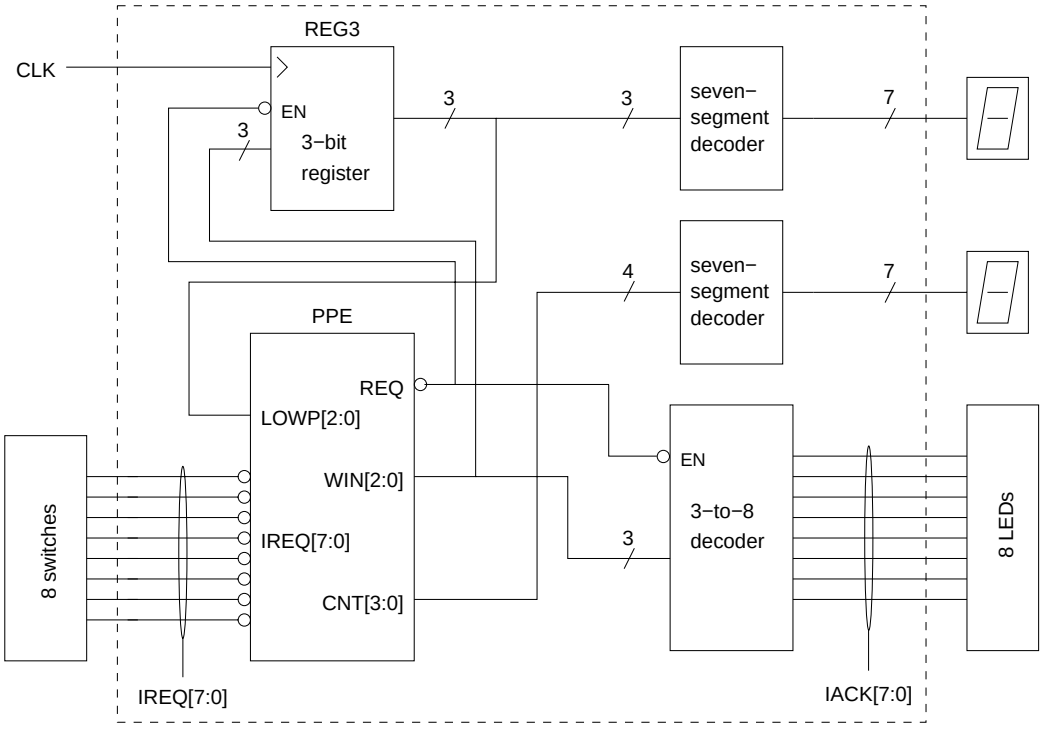
Other flip-flops

Flip-flops sometimes include additional logic.

- edge-triggered D flip-flop with enable
- master-slave S-R flip-flop
- master-slave J-K flip-flop
- edge-triggered J-K flip-flop
- T flip-flop (toggles with each clock edge)
- T flip-flop with enable
- scan flip-flop (used in boundary scan chains)



Lab #4: Block diagram

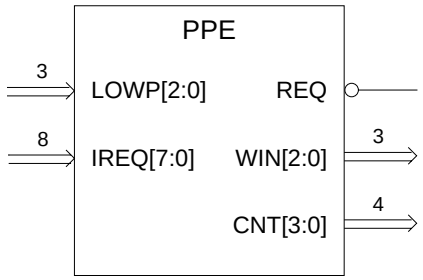


Lab #4: Auxiliary components

- 3-bit register stores state of system (last winning request).

- Seven-segment hexadecimal displays:
  - Used for observing system.
  - Two identical components
  - Simple combinational logic can be represented by a table (ROM).
- 3-to-8 decoder displays winning request. In a real system, the output would acknowlege request and initiate further action.

Lab #4: Programmable Priority Encoder



IREQ[7:0]	request inputs
LOW[2:0]	last winning request
REQ	any request active
WIN[2:0]	highest priority request
CNT[3:0]	number of active requests

- REQ is active low for historical reasons
- REQ is redundant since CNT is available, butit simplifies control of state memory flip-flops
- The main priority encoder output WIN[2:0] can be formed by multiplexing 8 priority encoders.