FF applications

1. Parallel data storage: register

![Parallel Data Storage Diagram]

* Inputs (I₀ - I₃) at S edge of CLK are stored in the register.
* CLR input clears the data asynchronously.

2. Shift register

![Shift Register Diagram]

* Data shifts right after each CLK pulse.
Apps:

* **serial to parallel** data conversion  
  (transmit digit. info over serial connection)

* divide/multiply by 2  
  (same as shift right/left)

* programmable delay

**Timing terminology**

transition time = time for FF output to go  
0→1 or 1→0 after the input change (c.f. prop. delay)

setup time = min. time inputs must be stable  
before clock pulse enables FF (~20ns)

hold time = min. time inputs must be stable  
after the clock pulse enables FF  
(~100ps, essentially 0)

③ Counters, ripple counters (asynchronous)

T flip-flops ganged together

asynchronous vs. synchronous counters

"domino effect":  
0111 → 1000  
0111 → 0110 → 0100 → 0000 → 1000
Timing circuits & clocks

* many cases when one needs to output a pulse at/after some event

"one-shots" set width/delay

* clocks are important to synchronize sequential circuits
  (need stable period, sharp rise/fall time)

Q: why need a precise clock?
A: E.g. watch "Longitude" (about John Harrison)

555 Timer

* 8 DIP package (like 741 op-amp)
* in part analog, in part digital

* can be wired as one-shot
  or as a clock (not very precise)
2 comparators $\frac{2}{3}V_{cc} \& \frac{1}{3}V_{cc}$

RS latch

saturated switch

$T_1 = \text{ON} \text{ if } \overline{Q} = 1$

$T_1 = \text{OFF} \text{ if } \overline{Q} = 0$

simplified symbol

supply voltage

$4.5V \leq V_{cc} \leq 16V$

output current

$(I_{out})_{max} \approx 200mA$

drives small loads

all by itself

\( \times \) as a one-shot: $T_w \approx 1\mu s \rightarrow 100s$

\( \times \) as a clock: $f \approx 0.01Hz \rightarrow 1MHz$

pin #

GND

Vcc

\( \overline{Q} \)

\( Q \)

\( \overline{Q} \)

\( Q \)

R

S

OUT

DISCHARGE

T1

\( 0 0 \) no change

\( 0 1 \) not allowed

\( 1 0 \) 0 1

\( 1 \) 1

RS

Q

\( \overline{Q} \)

set

reset

THRESH

CTRL

TRIGGER

1

2

3

4

5

6

7

8

GND

VCC

CTRL

RST

DISC

TRIG

GND

OUT

\( \frac{1}{2} \)