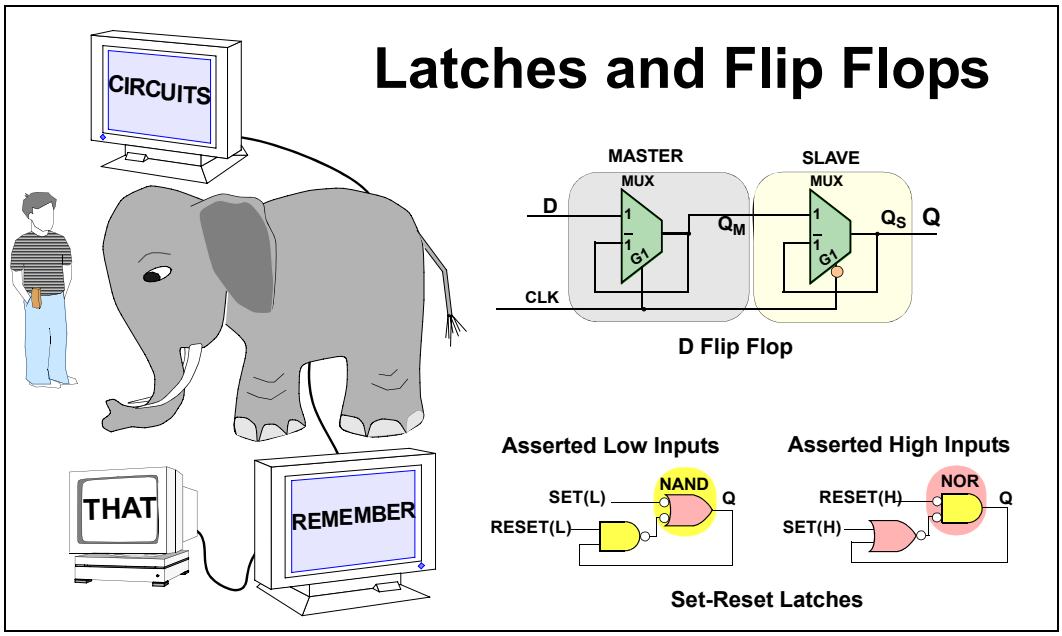


Carleton University 2009

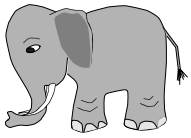
■

Latches and Flip Flops

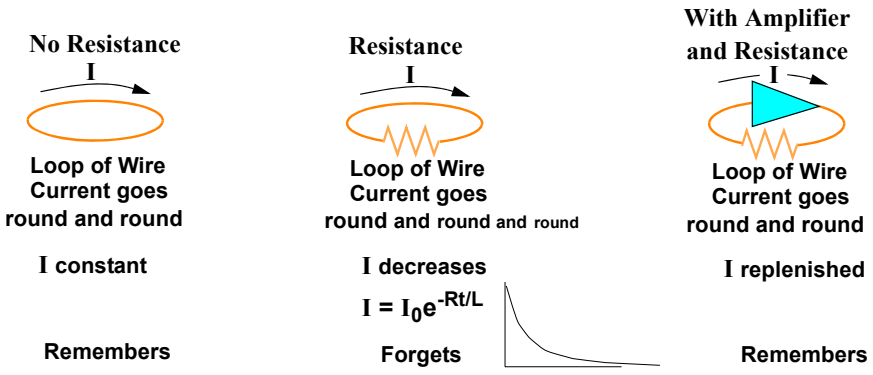
- Basic Latches
- RS Latches
- Transparent (D) Latches
- D Flip Flops
  - Master-Slave D Flip Flops
  - Asynchronous Reset
  - Synchronous Reset
  - Applications
    - Divide by Two
    - Shift Register
- Enabled D Flip Flops

Storage Circuits

Circuits That Can Remember



Storage

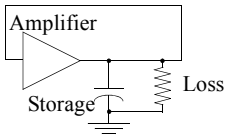


Storage Circuits

To Remember Electronically

- Need an inductor or a capacitor.[1]
- Need an amplifier to overcome the energy loss.
  Some superconducting circuits can store with no power loss, except you have to supply power to keep them cold enough.
- Amplifier must have a limit on its output so it does not increase the current or voltage forever.
  If any current is started in the loop above, the current will grow quickly and stop at the limit.

Capitative Memory



- The circuits in these notes are basically a capacitive storage circuit with an amplifier.

Other Ways To Remember.

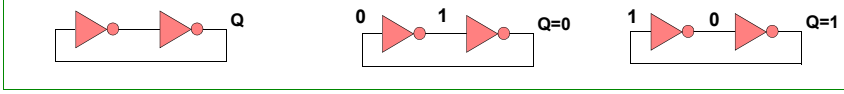
Although it is convenient in digital circuits, one does not have to remember by storage in an inductor or capacitor. For example one could throw a switch to a "1" or a "0".

[1] One does not absolutely have to remember by storage in an inductor or capacitor. For example one could throw a switch to a "1" or a "0".

The Most Basic Digital Latch

Basic Digital Latch

Two inverters connected in a loop.



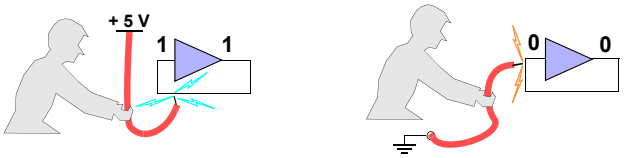
One noninverting gate in a loop.

In this gate: output signal = input signal.
But it has power gain;
The output can supply input power losses.

Will store whatever bit comes up on power up.

They will remember that bit until the power is shut off.

One must zap the input to set to "1" or clear to "0".



Basic Latch Circuit

Latches

Feedback

Feedback between gates will do one of two things:[1]

1. Oscillate, if the number of inverters in the feedback loop is odd.[2]
2. Remember, if the number of inverters in the feedback loop is even.
   Zero is an even number.

Latches

- The elements that remember are *latches*.
- Latches can be put together to form more complex elements called *flip flops*.
  These normally work from a clock which controls when the flip flop stores.
- RAM (Random Access Memory) is nothing but a very compact collection of latches.

Changing What is stored

A circuit that remembers continuously cannot change what it remembers.
To change what is remembered one must break the feedback loop and interrupt the storage.

Alternately one can change it by brute force as shown.

1. A single CMOS inverter fed back will do a 3rd thing, normally sit at a "1/2" value.

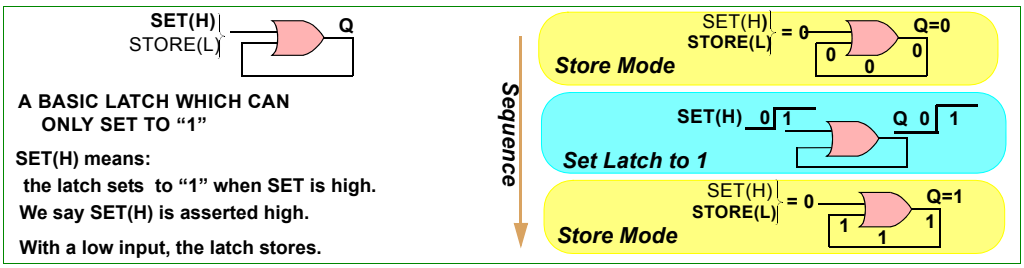
2. Prof. Rogers pointed out that a single CMOS inverter fed back to itself will not oscillate. In the analog model, one says it has only a single pole, and cannot give 180 degree feedback. In the digital model, we say the delay around the feedback path must be longer than the the *inertial delay* of inverter. This delay is the length of the minimum input pulse which can cause a pulse at the gate output.

Basic Latch With Set and Reset (Clear)

Adding SET and RESET to the Basic Latch

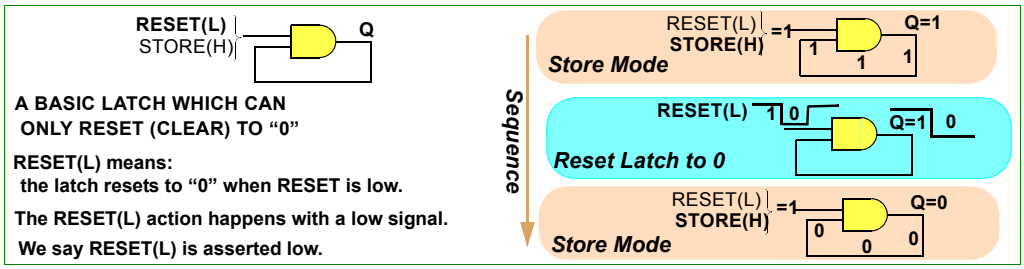
Adding Set

Replace the noninverting gate with an OR gate



Adding Reset

Replacing the noninverting gate with an AND.



Latches

Changing what is stored

These circuits can change what is stored, but only in one direction.

- The OR gate latches can be set to "1" if it is originally "0".
- The AND gate latch can be cleared to "0" if it is originally "1".

This needs to be improved.

Asserted High, Asserted Low Signals

We say a signal is asserted high if it performs the logic function stated by its name when high.

We say a signal is asserted low if it performs the logic function stated by its name when low.

Example: a line named OVERFLOW.

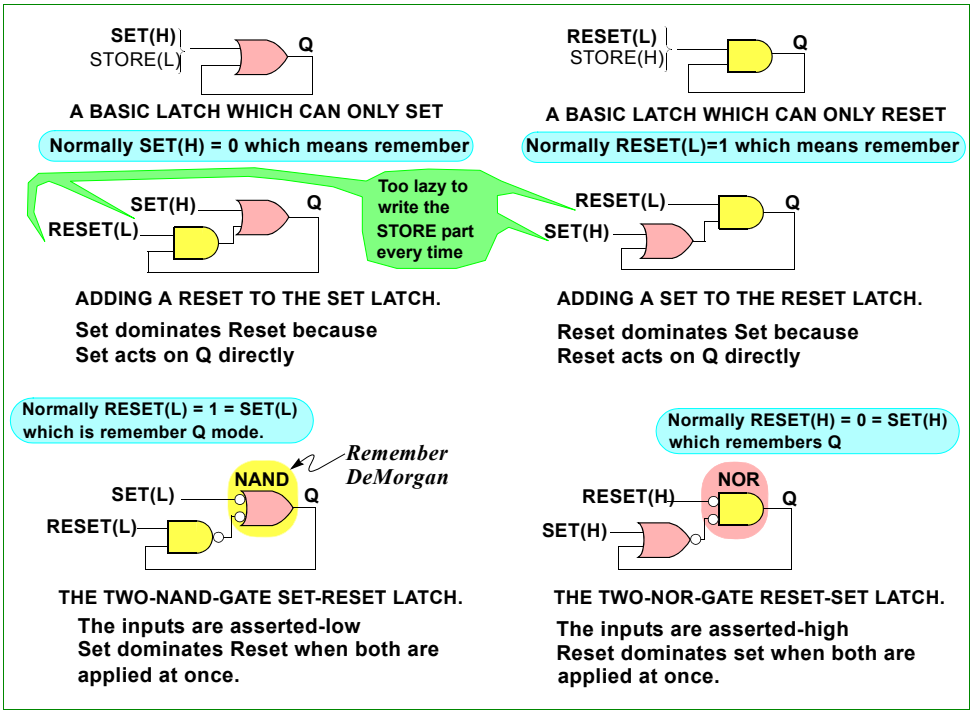
OVERFLOW(H) coming from a circuit would indicate an overflow if it was high.

OVERFLOW(L) coming from an circuit would indicate an overflow if it was low.

OVERFLOW without any "(H)" or "(L)" is taken as asserted high by default.

The Reset-Set (RS) Latch

The Basic Latch Converted to a Reset-Set latch

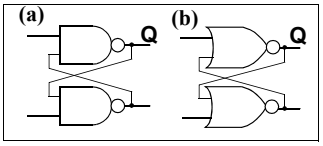


The Reset-Set Latch

The And-Or Circuit

- R-S latches are made from cross-coupled NANDs or cross-coupled NORs. However drawing them that way is the hardest way to understand them.
- The AND-OR circuit shows easily:
  a. which has asserted low inputs.
  b. which is Set dominant and which is Reset dominant.
- However it does not show the so called "not Q" output.

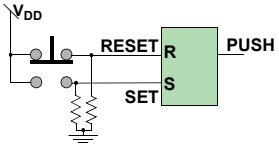
6-1.•PROBLEM_For (a) Cross-coupled NANDs, (b) Cross-coupled NORs
Label the inputs R(H), R(L), S(H), S(L) as appropriate.
Determine if Q is Set or Reset dominant



Applications of RS Latches

Debouncing Switches

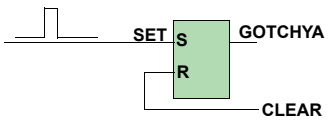
Use a switch that has an upper and lower set of contacts, which are never both contacting at the same time. If the switch is pushed, then the switch may bounce on the set contacts. But unless it bounces back so far it touches the reset contact the latch output will be stable.



Fast Pulse Catcher

To capture and hold a "glitch" on a line.

This property of capturing glitches is one reason why RS latches are not widely used. One often wants to ignore glitches.



The Transparent (or D Type) Latch

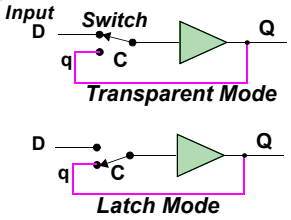
The simple Transparent latch

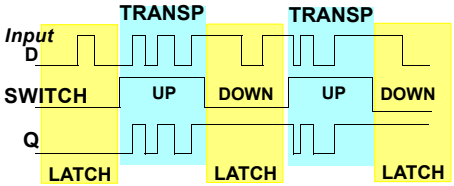
The simplest D latch stores a 0 or 1 on a capacitor.

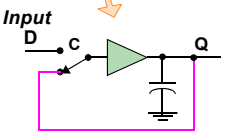
1. **Transparent Mode:**
   Switch up; the input signal goes straight through to the output.
2. **Latch (Hold) Mode:**
   Switch down, the latch remembers the old signal level

Thinking persons may ask, "Does it not forget during the time the switch is switching?"

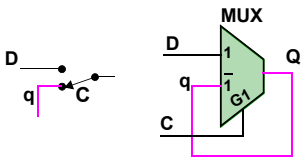
There must be a small capacitance to hold the signal for that instant.

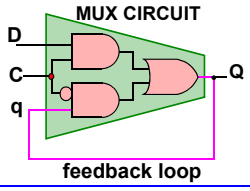






The switch is replaced by a multiplexer (MUX)





$Q = DC + \overline{C}q$

C=1, same as switch up

$\overline{C}$=1, same as switch down

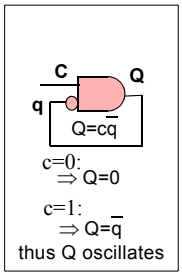
D Latches

The Simple D Latch

- This D latch has a switch to change it between the transparent mode and the storage mode.
- Some people will ask what happens during the time the switch is moving from Up to Down.
- All circuits have a small capacitance which holds the old value of the signal during the short time the switch is in motion.

The MUX

Recall the MUX was one of the common 3-input circuits. It acts like a digital switch.



Renaming Q

We rename Q to small q when it comes back around the feedback loop and acts as an input.

The signals are the same except when Q changes. Suppose C changes 1->0. Then the change in Q will depend on $\overline{C}q$, where q is the old value of Q. It is convenient to use a different symbol for new Q which will soon be the new output, and the old q, which will temporarily supply the input to calculate the new output. This will be important later on.[1]

The Latch Equation

$$Q = CD + \overline{C}q$$

This equation distinguishes between the new Q which can come when C goes to 1 (and D has changed), and the old q which was there just before when C was 0.

In transparent mode (C=1) Q = D.

In storage mode (C=0) Q = q. The new Q replenishes itself from the old

[1] There is always be a delay between when q changes and when it propagatesthrough the gate to make Q change.

The D Latch: Transparent High, Transparent Low

The Transparent-High D Latch

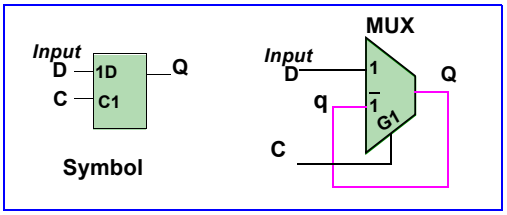
Control input C = 1

The latch is transparent and Q = D_IN.

Control input C = 0

The latch remembers the last thing going through when C was 1.

The latch uses feed-back to remember.



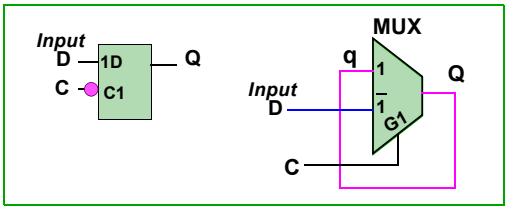
The Transparent-Low D Latch

Control input C = 0

The D latch is transparent.

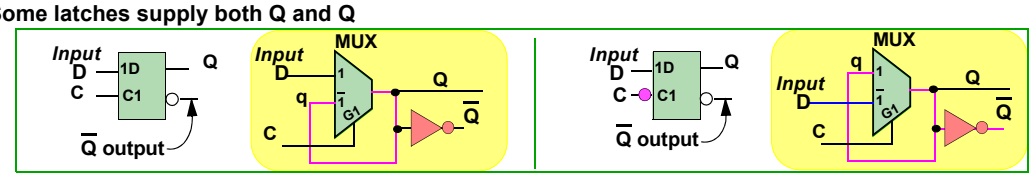
Control input C 1

The D latch remembers



$\overline{Q}$ Output

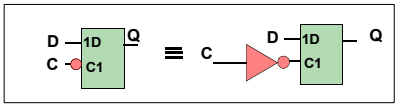
Some latches supply both Q and $\overline{Q}$



The MUX D-Latch

Inverting circles on inputs

An inverting circle on the input of a symbol is logically the same as having an inverter outside the symbol.



Latches are used to make flip flops

Latches can be put together to form more complex elements called *flip flops*.

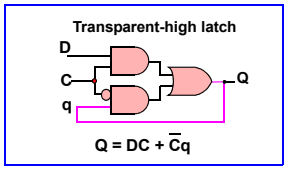
Flip flop do not have a transparent mode.

Logic circuits can generate unwanted fast output pulses (glitches) when there are unequal delays through the logic.

Because they are never transparent, flip flop will not allow these glitches to pass through.

6-2.•PROBLEM.

The gate-level circuit and equation for the transparent-high D latch are shown on the right. Find them for the transparent-low D latch.



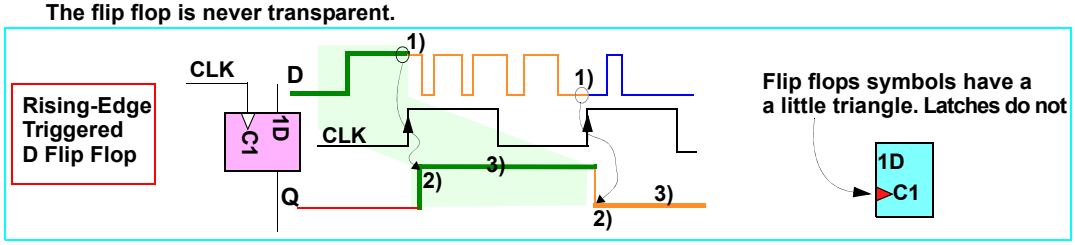
The Edge-Triggered D Flip Flop

Rising-Edge Triggered D Flip Flop

1) The flip flop samples the D input just before the rising clock edge. CLK

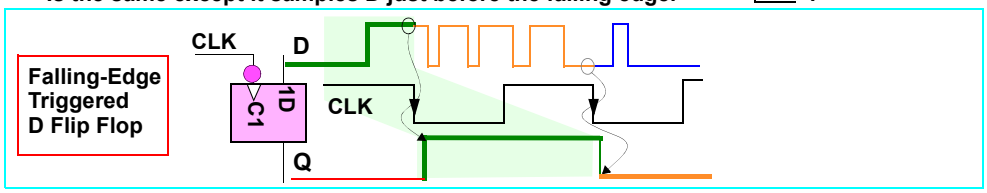
2) It sends the sample to output Q just a TINY BIT after this clock edge
3) and holds it.

It ignores D except near the clock edge.
The flip flop is never transparent.



The Falling-Edge-Triggered D Flip Flop

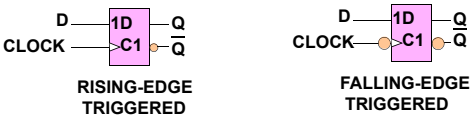
Is the same except it samples D just before the falling edge. CLK .



The Edge-Triggered D Flip Flop

D Flip Flop Symbols

The symbols for the master-slave D flip flops.
Note the little triangles —▷C1.
These mean edge triggered.
The circle in front of the triangle means falling-edge triggered —○▷C1.



Some flip flops have only a Q output. Others have both a Q and a $\overline{Q}$ output.

Timing

Pay particular attention to:

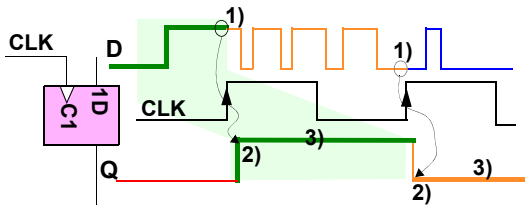
- The flip flop samples the D input just before the rising clock edge.
- It sends the sample to output Q just after this clock edge

Data transfer through the flip flop does not happen in zero time.

REMEMBER

The D FlipFlop

1) Samples the D input just before the active clock edge
2) It places that value on the Q output just after the edge
3) The Q does not change until the next active clock edge



- This is the most important fact about flip

This is the most important fact about flip flops

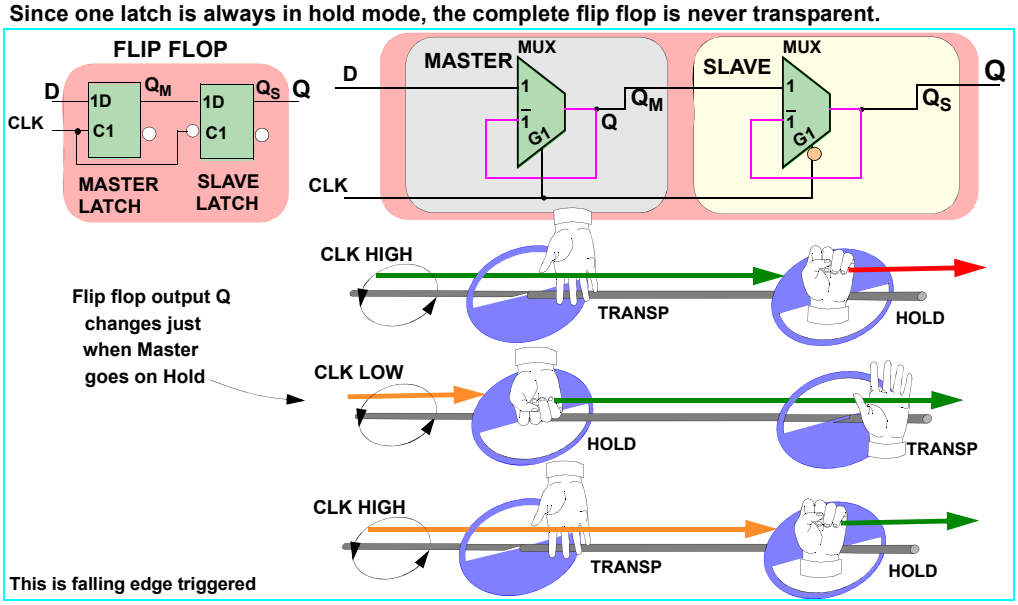
The Master-Slave Edge-Triggered D Flip Flop

Flip Flops Are Made From Latches

Two D latches in series, one clocked from CLK and one from $\overline{CLK}$.

When the MASTER is transparent the SLAVE is holding.
When the MASTER is holding the SLAVE is transparent.

Since one latch is always in hold mode, the complete flip flop is never transparent.



D Flip Flop

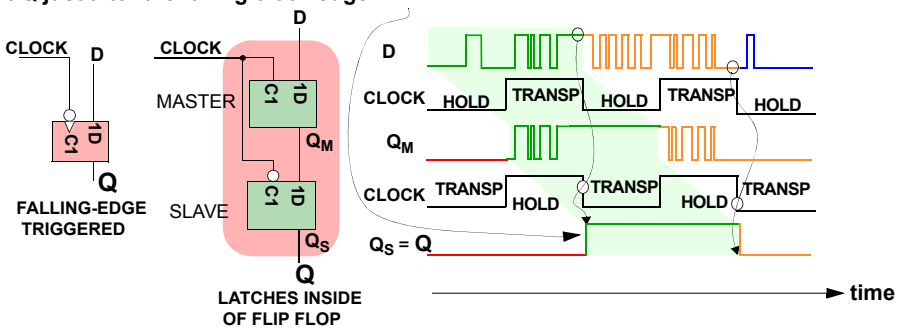
The flip flop holds the data for the full cycle even though the two latches only hold data for half a cycle.
The master latch grabs fresh data and sends it out thought the transparent slave latch for half a cycle.
Then the slave latch captures the same data and sends it out for the second half cycle.

The Master Slave Flip-Flop Operation

How the master-slave D flip flop works

It is *Edge-Triggered*

- **The data that enters D just before the clock falls comes out Q just after the falling clock edge.**



- **All master-slave D flip flops are edge-triggered but not all edge-triggered D flip flops are master-slave.**

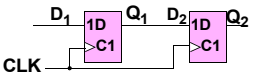
D Means *Delay*

D flip flops delay data from one clock cycle to the next.

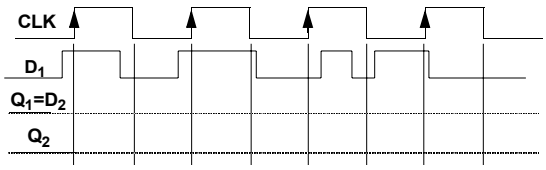
The Master-Slave D flip flop

The master-slave circuit made from D latches, is a very common circuit for a D flip flop but not the only one. Another common one is the 6-state D flip flop. A brief description is given in the footnote[1].

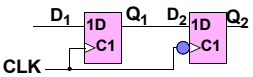
6-3.•PROBLEM



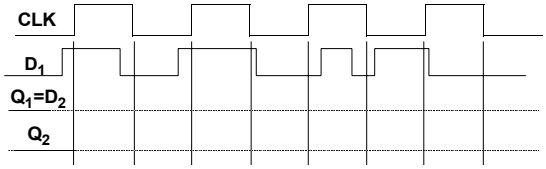
Complete the waveforms for D_2 and Q_2. The flip-flop outputs are initially zero.



6-4.•PROBLEM

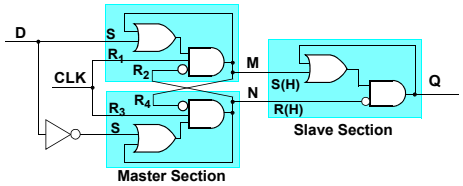


Complete the waveforms for D_2 and Q_2. The flip-flop outputs are initially zero.



1. 6-State D Flip Flop

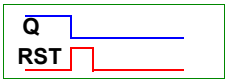
The master section has two R-S latches each with an extra R input
The slave section uses a single reset-dominant R-S latch.
When CLK=0:
M=0=N, and Q remembers the previous value.
When CLK=1:
M will be set if D=1, N will be set if D=0.
If M sets first it will hold a reset on N using R_4
If N sets first it will hold a reset on M using R_2
Thus M,N will be 1,0 or 0,1 and Q will be either set or reset.
When CLK returns to 0, Q will remember this previous value.



Asynchronous Reset (Clear)

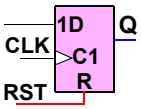
Flip Flops with Asynchronous Reset (Direct Reset)

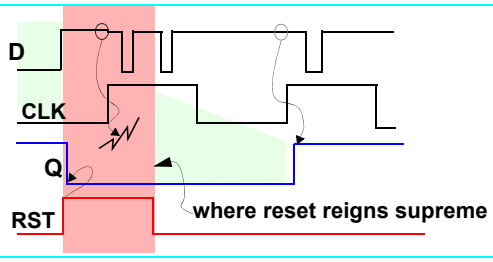
Reset



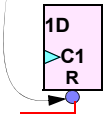
- Reset (RST = "1") clears Q to "0" immediately.
- Once it is cleared, it stays cleared until a high D input is clocked through.
- Reset overrides the D input and CLK.

Rising Edge Triggered D Flip Flop





Many flip flops have an active low reset



Reset is used to initialize all flip flops to "0" on start-up.
On power-up, a flip flop may initially be "0" or it may be "1".
Without reset, one would not know how digital machines would start.

Other Asynchronous Reset Symbols

Xilinx™ uses this

Reset

When one first turns on a digital circuit, one would like it to start in the correct state. It should not start in a routine which might cause damage.

Reset is used to be set the circuit into the proper state. When you reset (reboot) your computer, you are resetting all the flip flops among other things.

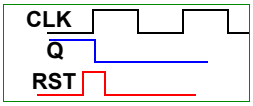
Documentation for Xilinx™ complex programmable logic devices (CPLDs) and field-programmable gate-arrays (FPGAs), which are used in Carleton's ELEC 2607 and ELEC 3500 courses, use:

- CLR and PRE for asynchronous reset and set (preset) respectively.

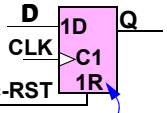
Synchronous Reset for Flip Flops

Reset

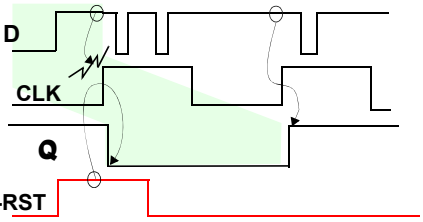
- Reset (Synch_RST = "1") clears Q to "0" after an active clock edge.
- Once Q is cleared, it stays "0" until:
  Synch_RST = "0", and D=1 just before an active clock edge.
- Synch-RST overrides the D input but it must wait for the clock.



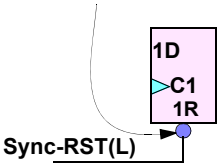
Rising Edge Triggered D Flip Flop with Synch. Reset



1R means synchronous reset
R alone is asynchronous



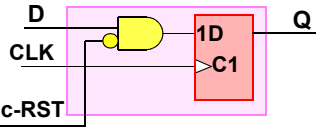
Many flip flops have an active low reset



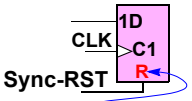
Reset is also used to initialize all flip flops to "0" on start-up.

In some designs, synchronous may be better (take next year's course).t.

Equivalent circuit of a synchronous reset



Another Synchronous Reset Symbol



Some manuals use R for synchronous reset
Some use R for asynchronous reset
This causes confustion!

Synchronous Reset

Set

Besides reset, flip flops may have asynchronous or synchronous *set*. This makes the output "1" instead of "0".

Synchronous and Asynchronous Notation

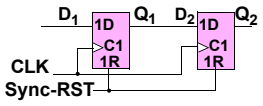
The American National Standards Institute (ANSI) and the Institute of Electrical and Electronic Engineers (IEEE) standard *ANSI/IEEE Std 91a-1991* uses:

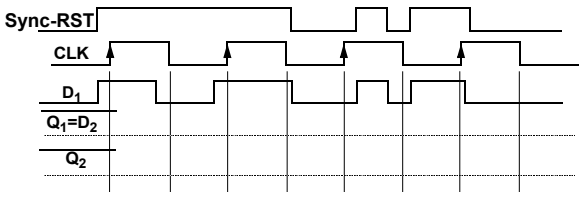
- **1R** and **1S** for synchronous reset and set, and[1]
- **R** and **S** for asynchronous reset and set.

Documentation for Xilinx™ complex programmable logic devices (CPLDs) and field-programmable gate-arrays (FPGAs), which are used in Carleton's ELEC 2607 and ELEC 3500 courses, use:

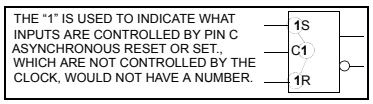
- **R** and **S** for synchronous reset and set, and
- **CLR** and **PRE** for asynchronous reset and set (preset).

6-5.•PROBLEM: Complete the waveforms for Q_1 and Q_2. They are initially "1" as shown



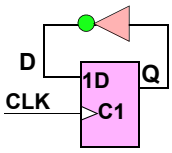


[1] The ANSI/IEEE notation uses numbers to indicate what signals have control over other signals. A number after a pin name indicates it controls the pin(s) which have the same number before them. Thus synchronous set and reset have a preceding number, and asynchronous set and reset, which are not controlled by another signal, would not have a nunber.



Circuits Using Flip Flops

Divide the Clock Frequency By Two



1st Cycle

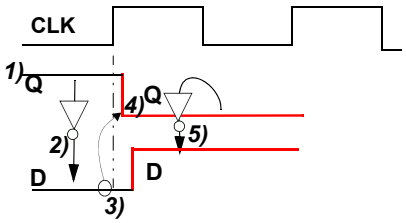
1) Assume Q starts at 1

2) The inverter makes D = 0

3) The flip flop captures D just before the clock edge

4) The flip flop transfers D to Q just after the clock edge.

5) After Q changes to 0, the inverter changes D to 1



2nd Cycle

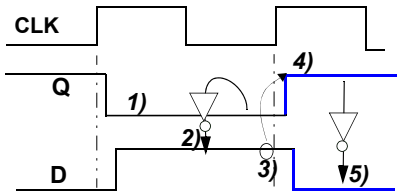
1) Now Q is at 0

2) The inverter makes D = 1

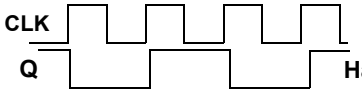
3) The flip flop captures D just before the clock edge

4) The flip flop transfers D to Q just after the clock edge.

5) After Q changes to 1, the inverter changes D to 0.



Final Waveform



Half the clock frequency

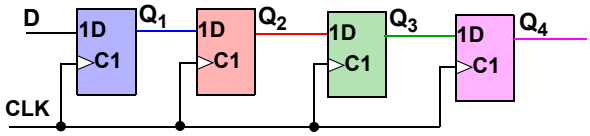
Frequency Dividers

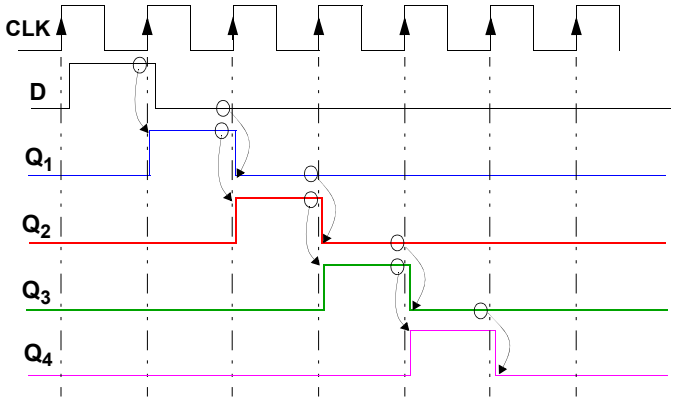
These circuits contain storage

- This circuit divides the clock frequency by two.
- Similar circuits can divide by 4, 6, 8, 10, ...
- Dividing by 3, 5, 7 ... is harder.

Circuits Using Flip Flops

A Shift Register





Someone generated the input pulse D

After that the pulse travels through the register, delayed by one clock cycle at each flip flop.

The Q output of a flip flop is the D input for the next flip flop

If the D pulse is two cyles long. what will the Q outputs look like?

The Shift Register

The state is the collective contents of all the storage in some prearranged order

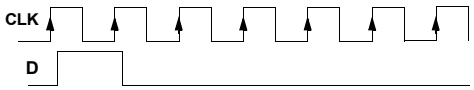
Whatever waveform is fed into the first D input, is shifted through all the flip flops.

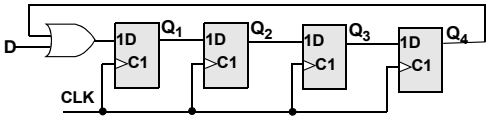
Remember that the flip flop only transfers what comes in just before the clock edge. Hence if the D input goes up and down between the clock edges, those bounces do not get shifted through.

A shift reqister is very useful to delay a waveform for a few clock cycles.

6-6.•PROBLEM

Plot the waveforms that will come out of this circuit Q_1, Q_2, Q_3 and Q_4, for the single-pulse D input. Assume all Qs are initially 0.



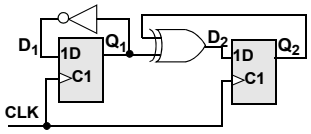


6-7.•PROBLEM

Plot the waveforms that will come out of this circuit, Q_1, Q_2. Assume Q_1 and Q_2 are initially 0. .

Hint

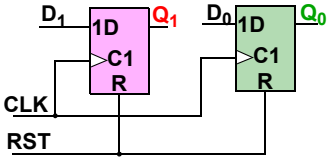
Start with Q_1 and Q_2.
From them calculate D_1 and D_2.
After the clock edge, Q_1 becomes D_1 and Q_2, D_2.
From the new Q_1 and Q_2, recalculate D_1 and D_2.



States

The State

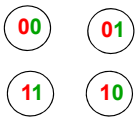
- The state is the collective output of all flip flops and latches in some agreed order.
- Most design only uses flip flops.
  Then the state is the collective contents of all flip flops.



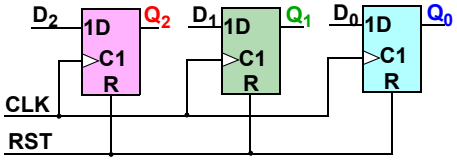
two flip flops

Four Possible States

Q_1	Q_0
0	0
0	1
1	1
1	0



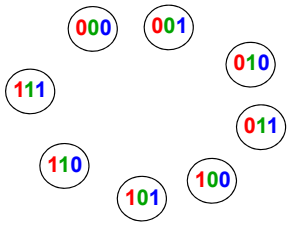
We often show the states inside circles



three flip flops

Eight Possible States

Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	1
0	1	0
1	0	0
1	0	1
1	1	1
1	1	0



The State

The state is the collective contents of all the storage in some prearranged order

The order is arbitrary. Only once you define it for one state, you must use the same order for all states.

If a finite state machine has 4 flip flops, Q_A,Q_B,Q_C,Q_D and they contain 1,1,0,1, then the state is 1,1,0,1.

6-8.•PROBLEM

How many states are possible with 8 flip flops?

6-9.•PROBLEM

How many flip flops would be needed to make a digital counter that would count to 20 in binary?

Flip-Flops in Hibernation

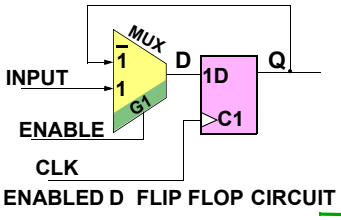
The Enabled D Flip Flop

A simple way to hold the previous Q over several clock cycles.

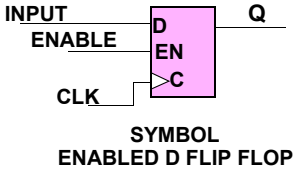
It switches the flip flop D input between the old Q, and the new input.

Operation

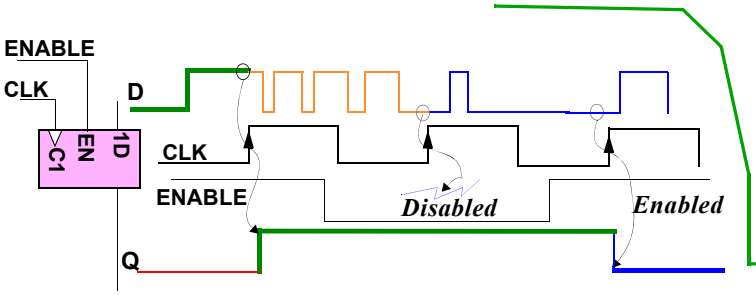
- **When ENABLE is high the INPUT is fed into the D flip flop.**
- **When ENABLE is low Q reloads the flip flop from its own output.**

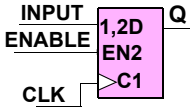


ENABLED D FLIP FLOP CIRCUIT



SYMBOL
ENABLED D FLIP FLOP





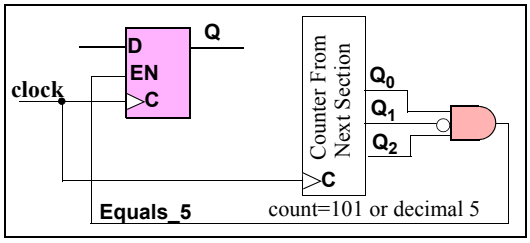
IEEE SYMBOL
ENABLED D FLIP FLOP

Enabled flip Flops

These help thinking about logic

Suppose you only want a flip flop to change every 5th clock cycle out of 8. You can use a binary counter. Use a gate to select when the count is 5, and use that signal to enable the flip flop.

Binary counters are the first topic in the next section on circuits using flip flops.



Don'tShut Down The Clock

An alternate circuit would be to AND the Equals_5 signal with the clock. **DON'T**.
ONE NEVER PUTS GATES IN THE CLOCK LINES UNLESS ONE IS A VERY KNOWLEDGABLE DESIGNER.[1]

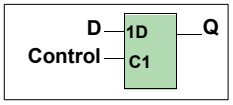
[1] Putting a gate in the clock line to one flip-flop will delay the clock to that flip-flop slightly. This means some other flip-flops may flip sooner. If you don't see a problem with this, go back and try doing the shift-register output problem,. with the clock to flip-flop 2 delayed 1/8 cycle from that of the other flip-flops. Alternately, be patient. This will be discussed in later courses.

Summary

D Latches

When Control = 1, D input feeds through to Q output.

When Control = 0, Q stays the same as (remembers) the last value before Control went low.



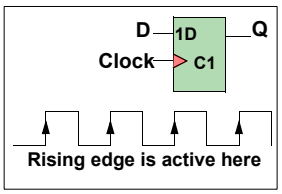
D Flip Flops *(Have a little > in the symbol)*

They change just after an active Clock edge.

The flip flop samples D just before the edge

The flip flop puts that value on Q just after the Clock edge.

Q never changes except just after the active Clock edge.



Add Ons

Asynchronous Reset: clears Q to 0 immediately.

Synchronous Reset: clears Q to 0 after the next active clock edge.

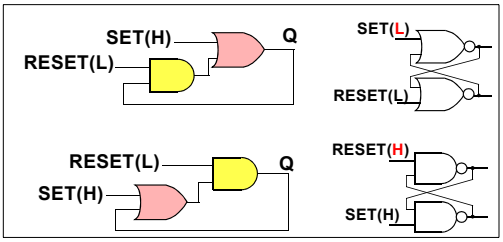
Enable: if Enable=1, flip flop acts normally;
if Enable=0, Q stays at its old value.

R-S (or S-R) Latches

Set Dominant

Reset Dominant

With clocked circuits use D-flip-flops



Common Problems

Most Common

Not memorizing and UNDERSTANDING the D flip-flop operation given above.

When analyzing signals going through circuits:

1. Check for resets. If no resets, initial Q values must be given.
2. Trace these Q signals through the gates to find the D input to all flip-flops.
3. Wait for the active clock edge,
4. Transfer the signals D ->Q in each flip-flop.
5. Repeat steps 2 to 4.