SWINBURNE UNIVERSITY OF TECHNOLOGY

COS10004 Computer Systems

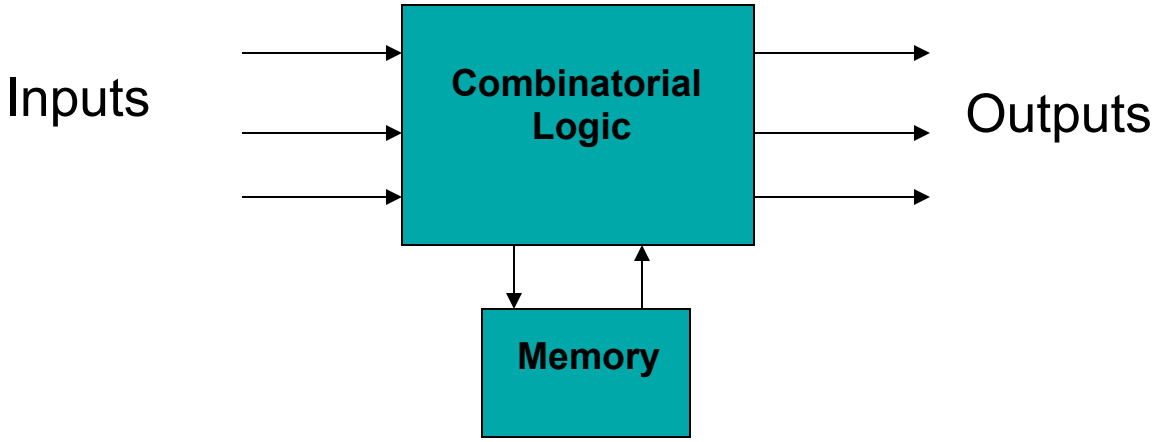
Lecture 2.4 – Clocked Flip Flops (D, JK and T FFs)

CRICOS provider 00111D

Dr Chris McCarthy

BASICS OF ELECTRONIC STORAGE - CPU REGISTERS

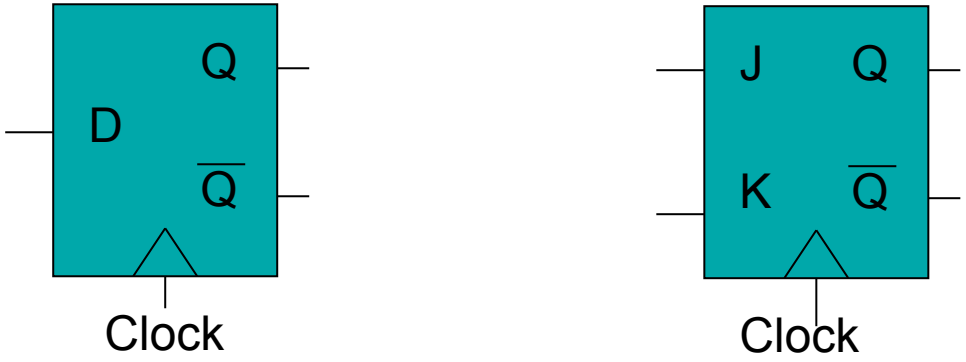
- In a computer the output is determined by current inputs and memory (previous inputs) computed together.



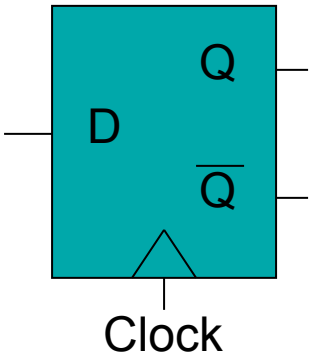
CPU needs some way to remember information

FLIP-FLOPS WITH CLOCKED INPUTS

- The two most common *clocked* Flip-Flops are the D and the JK

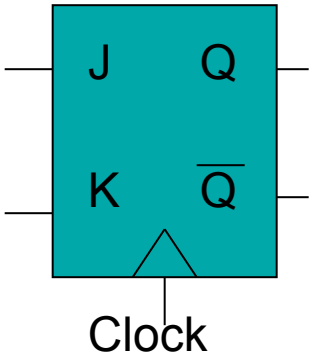


D Flip Flop



- The D flip-flop has only one input, Q is updated to be the same as D when the clock goes active

J-K Flip Flop



The JK flip-flop has two inputs.

It is updated when the clock goes *active* but how it is updated depends on both the J and K inputs.

D Flip-Flop

The external D input (Data) internally generate both an R and an S input.

These are complements so never get both being active at once.

Hence the true table for D and Q after each clock change to active is

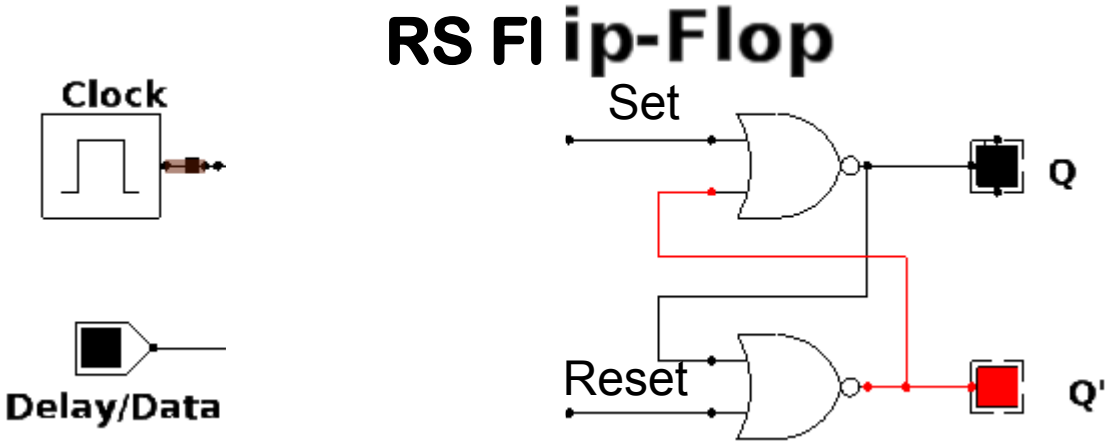
D_N	Q_{N+1}
0	0
1	1

(N means at the clock, N+1 means after the clock)

D-FFs are used in computer registers and memories and in counters and shift registers

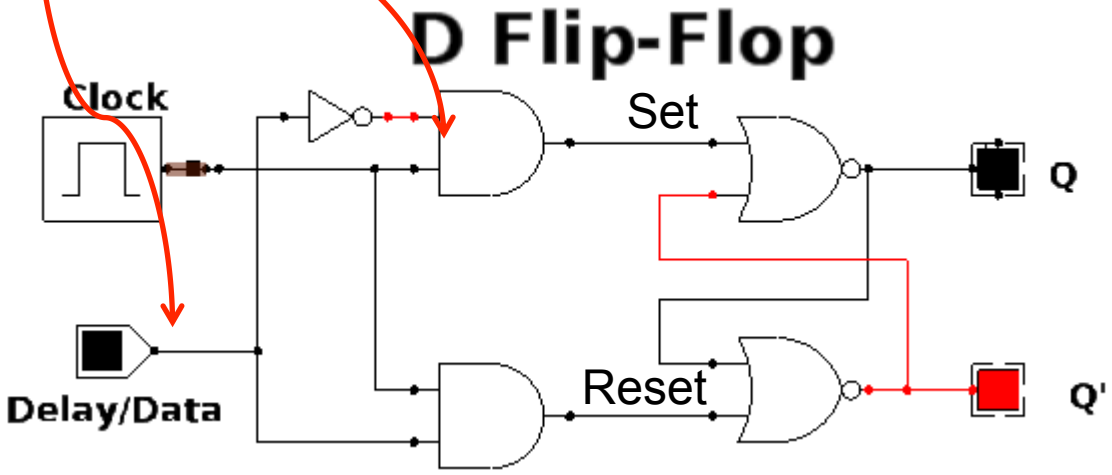
Exercise

- **From the description given in the last slide, convert the NOR-gate RS FF into a D-type FF. Use the rising edge of the Clock for "active".**
- ***Hint: add a couple of AND gates controlled (oppositely) by D***
- ***Clock the AND gates …***

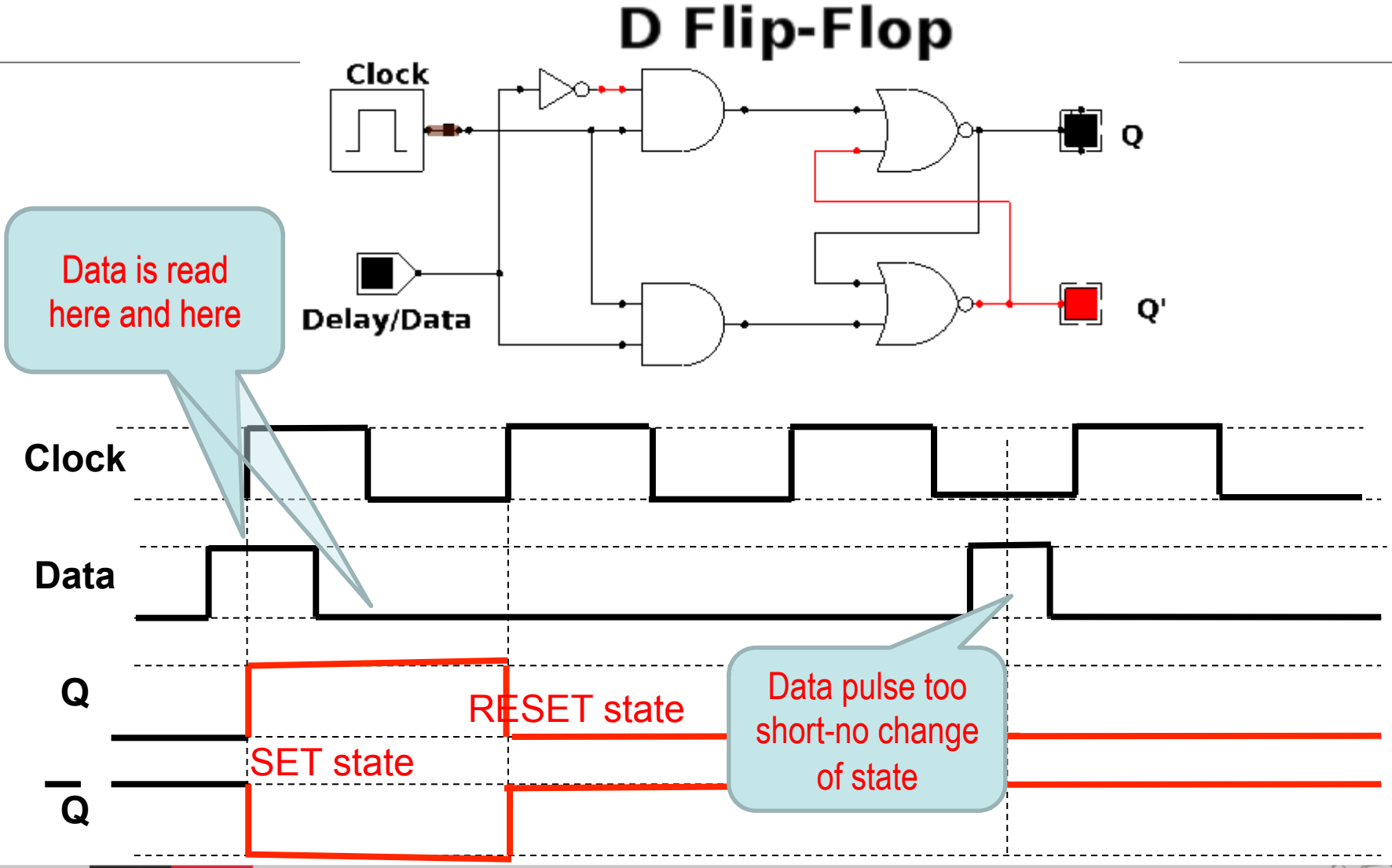


Exercise - Solution

- Step 1, Add a couple of AND gates to permit updates of state when the clock ticks over.
- Step 2, Replace R and S with one input (D) and use an inverter to ensure that R is always the opposite of S.



Exercise – complete the timing diagram below



J K Flip-Flop

J K FF's are more flexible and can be used for a number of operations.

The JK truth table (remember that N+1 means after the clock)

J	K	Q_{N+1}
0	0	Q_N (No Change)
0	1	0 (Reset)
1	0	1 (Set)
1	1	$\overline{Q}_N$ (Toggle)

Note the change

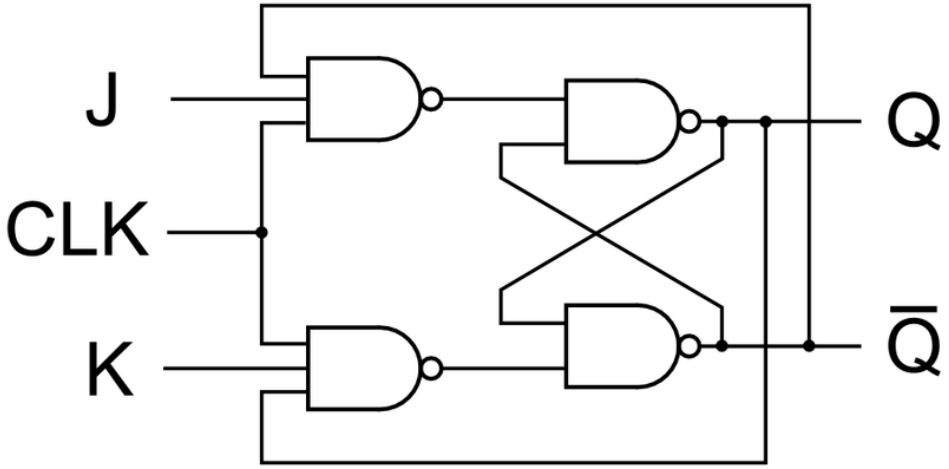
A sort of programmable gate

The state when both inputs are active has been turned from a problem state (as it was for an RS flip flop) into something very useful (more on this later).

J K Flip-Flop

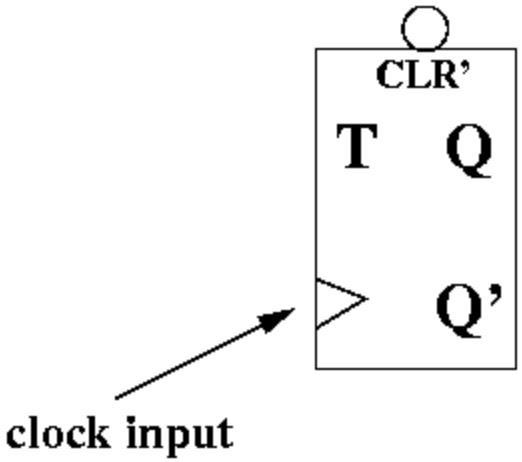
- Uses tri-input NAND gates.
- Has a defined state (toggle) when both J and K (equivalent to S and R) are high.
- Waits for the clock before it changes state – can't be made unstable.

Q' is another notation for $\overline{Q}$



T FLIP-FLOPS

- The T input just toggles the output.



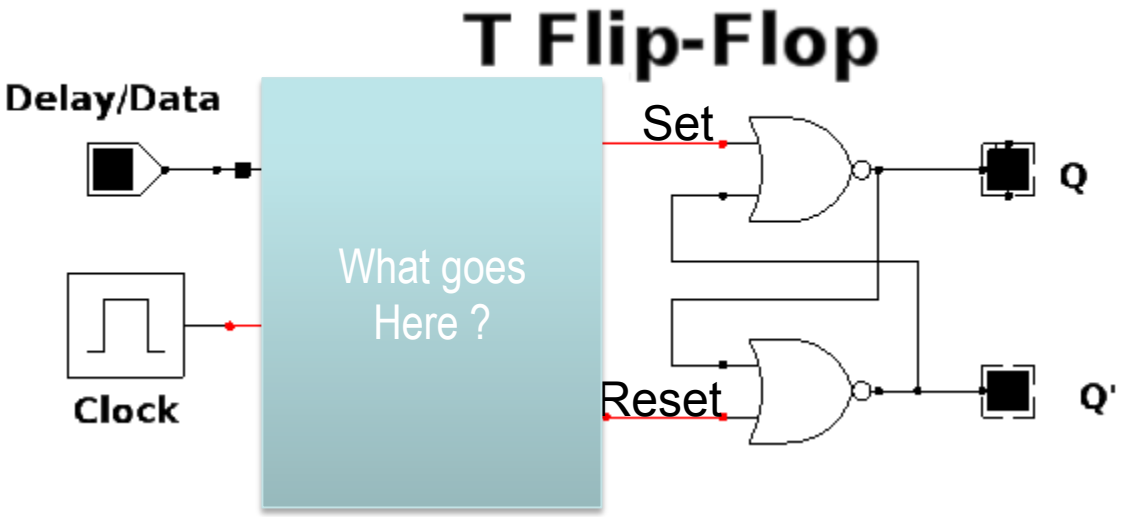
Note the ‘ notation. Q’ means the same as $\overline{Q}$

For inputs, CLR’ == $\overline{CLR}$ which means:

“pull CLR down to activate”

Making a T from an R S Flip-Flop

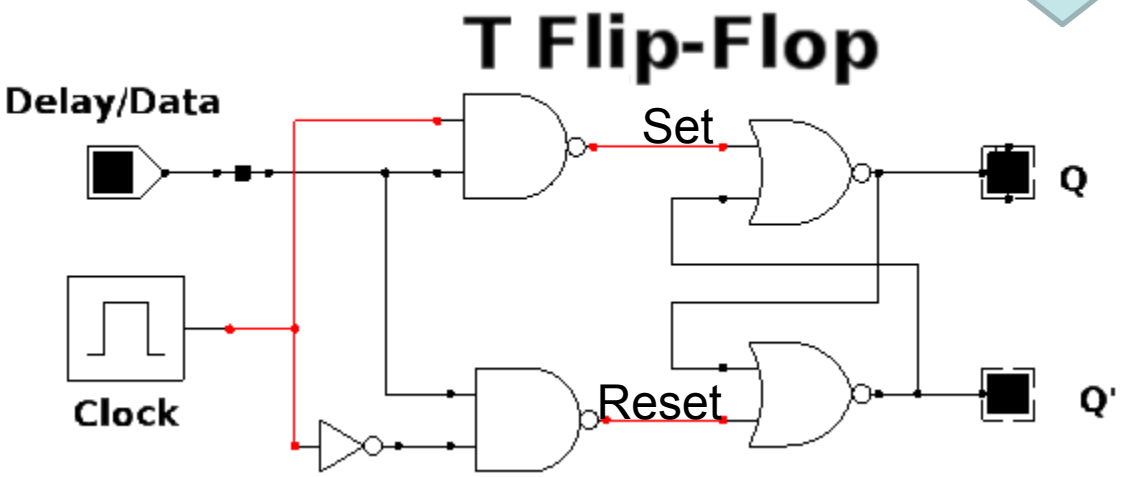
- ***Add a couple of NAND gates controlled by D***
- ***Clock the NAND gates* inversely…**



Making a T from an R S Flip-Flop

- ***Add a couple of NAND gates controlled by D***
- ***Clock the NAND gates* inversely…**

We invert the Clock to one NAND gate instead of inverting the Data



More on clocked flip-flops

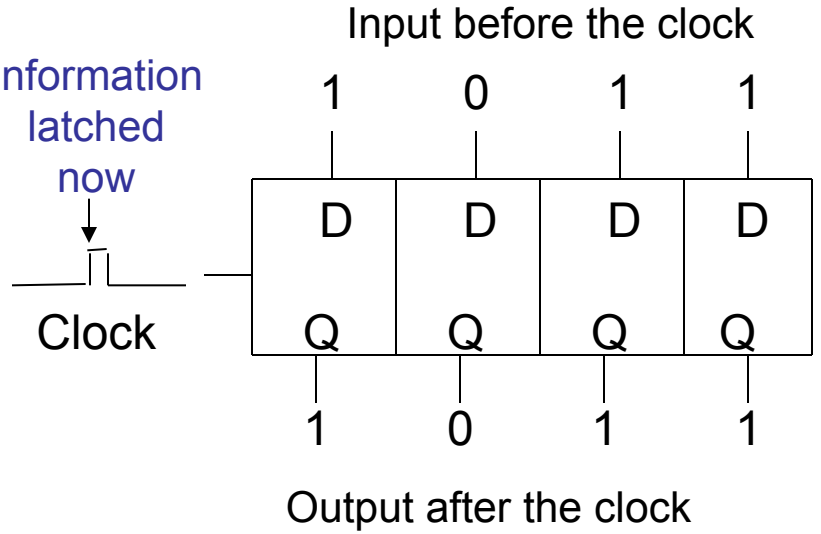
- Sometimes it is useful to provide both clocked (synchronous) and non-clocked (asynchronous) inputs.
- When power is applied to a flip flop its state cannot be predicted.
- The asynchronous inputs are used as master reset (MR) or set (MS) inputs and override the clocked inputs if we should try to use one of these at the same time as the clock edge.
- http://wearcam.org/ece385/lectureflipflops/flipflops

D-Flip-Flops as a Register or Latch

A *register (many bits)* or *latch (usually one bit)* can be made up from a series of D-Flip-Flops driven by a common clock.

The transfer from the D side to the Q side for all D flip flops occurs simultaneously as this clock changes.

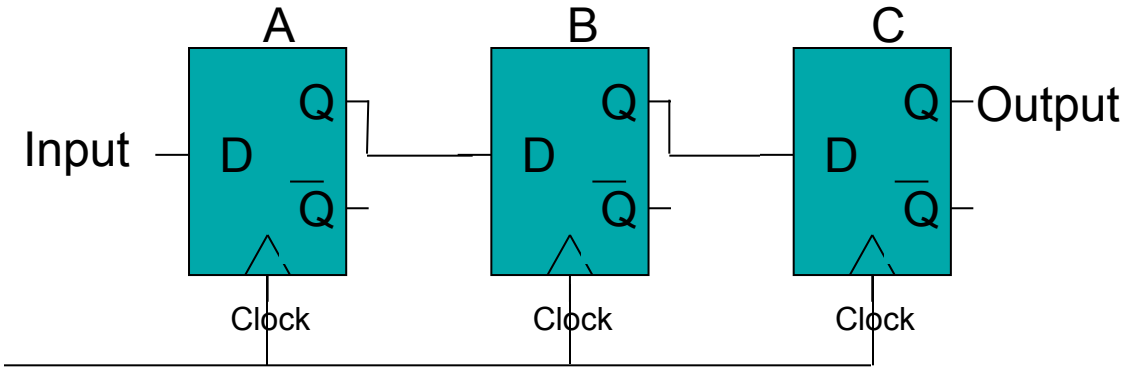
This arrangement is found in CPU registers

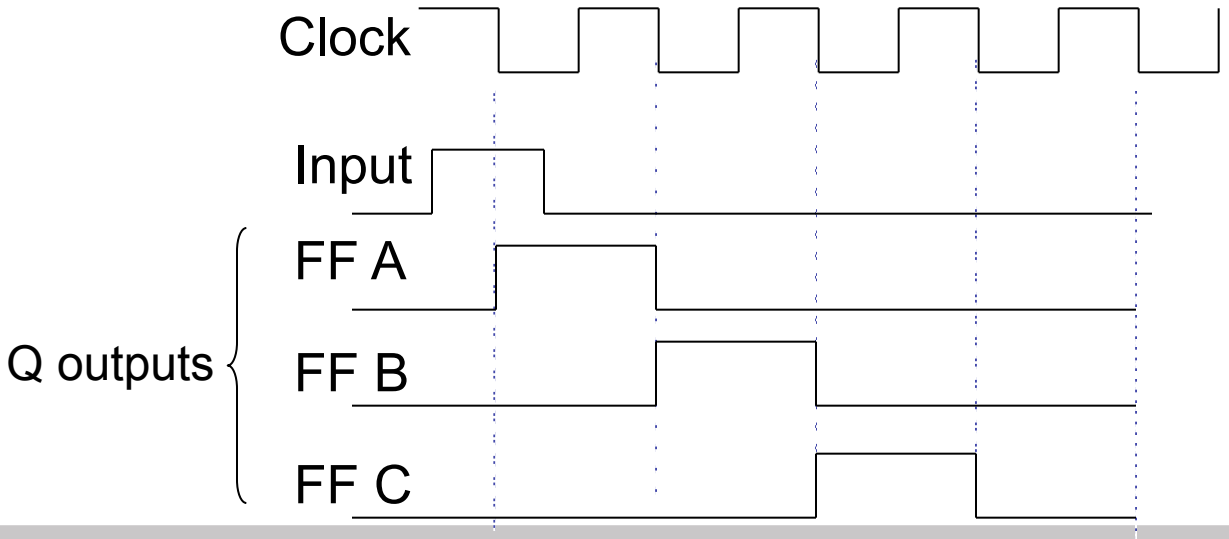


http://www.electronics-tutorials.ws/sequential/seq_4.html

SHIFT REGISTERS

Here is how a *high* travels through a 3 bit shift register. For this example we assume that each of the shift register bits is cleared at the start.

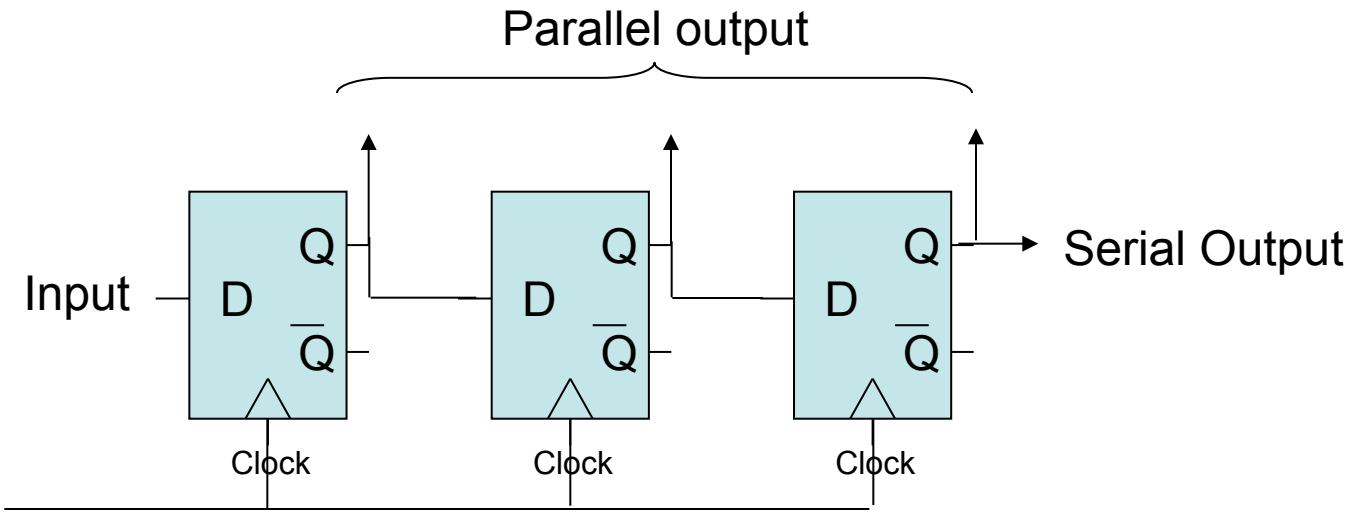




SHIFT REGISTERS

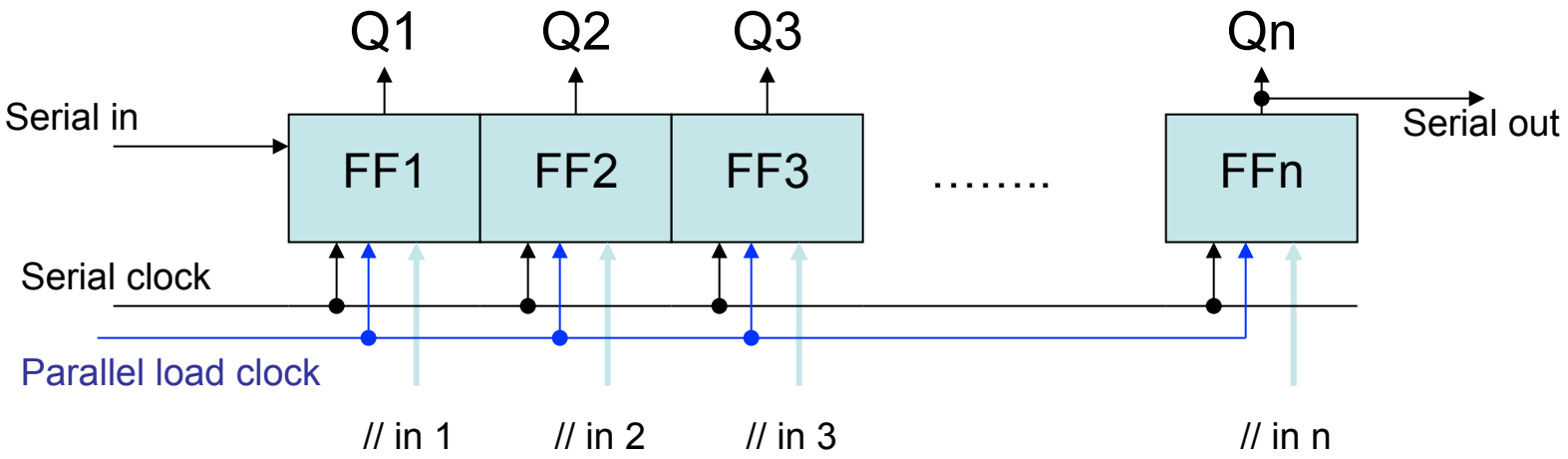
A shift register takes input from one end, and at each clock change this value is moved to the next D-Flip-Flop.

This is used in serial data transfer when a byte (say) of data sent on a cable one bit after another can be collected in a series of D Flip-Flops to rebuild the whole data byte. This is called *serial-to-parallel* conversion.



PARALLEL LOAD SHIFT REGISTERS

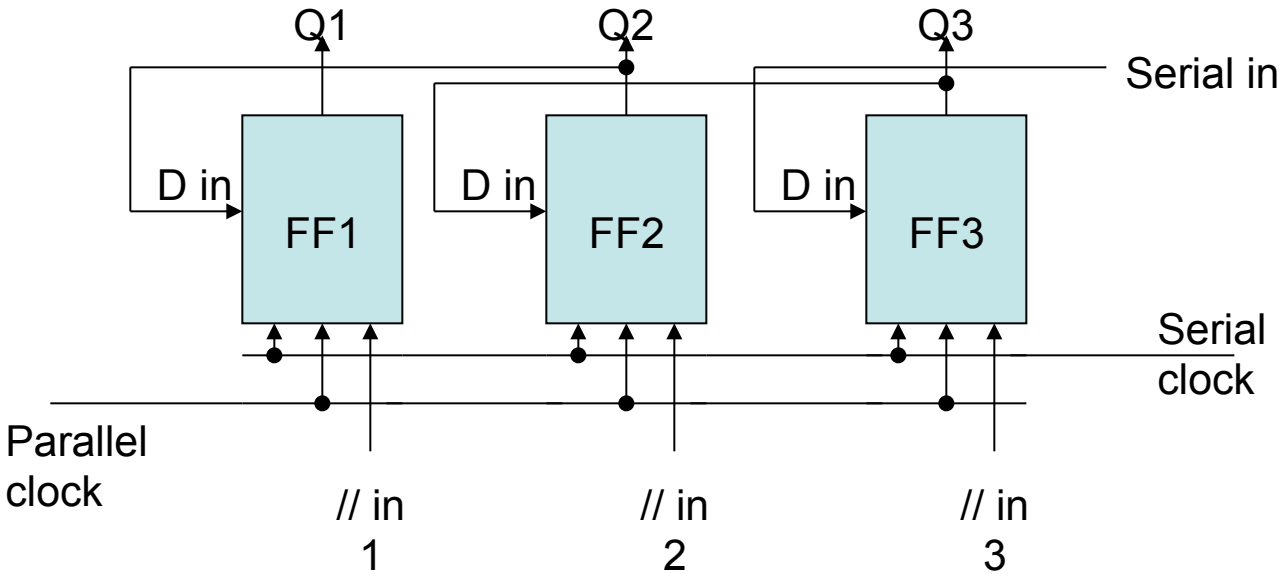
- Some shift registers have the ability for you to load all their flip-flops at once, i.e. in *parallel*.



- Doing this then allows you to "clock out" each bit in turn (starting with bit *n*) using the serial clock. This gives *parallel-to-serial* conversion. (// is a common abbreviation for "parallel")

SHIFT LEFT SHIFT REGISTERS

- The shift registers shown so far shift data to the right. A simple rewiring gives a shift register that can move data left. Of course these may also have the ability to parallel load.

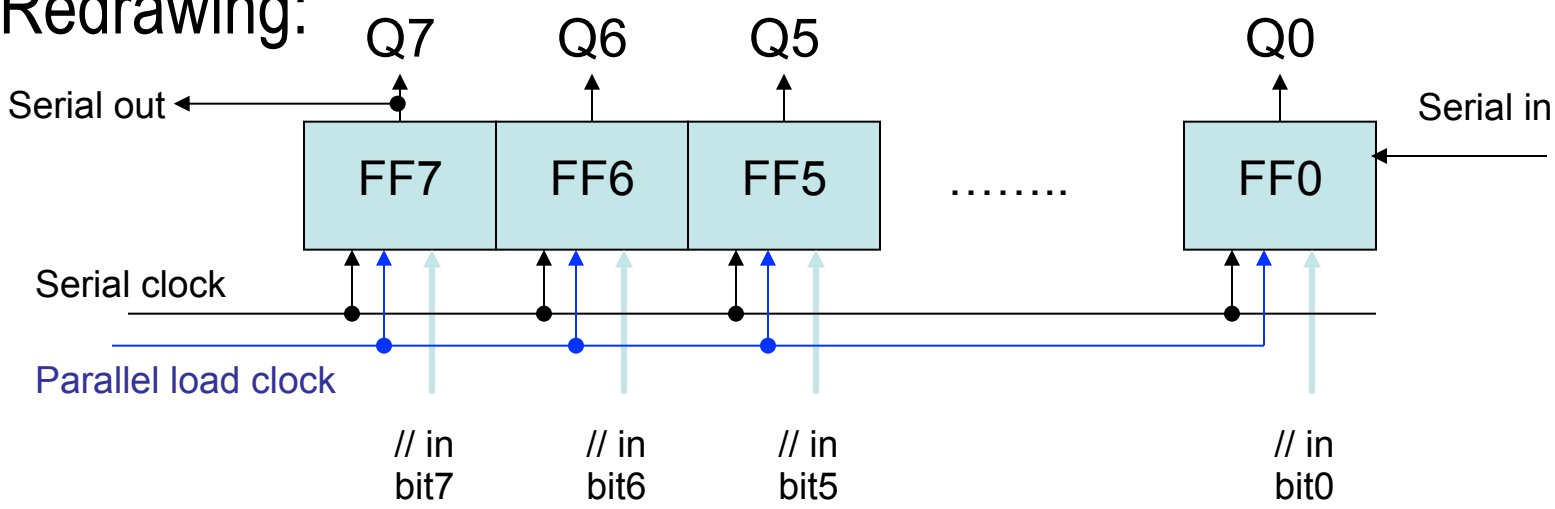


SHIFT REGISTERS AND MATHS

- As well as serialising and de-serialising data (parallel to serial, serial to parallel), shift registers have other purposes.
- Shifting a binary number left one place multiplies it by 2.
- Shifting a +ve binary number right one place divides it by 2.
- Shifting left, shifting right, in addition to adding and complementing, allow us the perform the basic arithmetic operations – ***addition, subtraction, multiplication*** and ***division.***
- Multiplication is performed as a series of shifts-left and adds.
- Division is performed as a series of shifts-right and subtractions.

"LEFT" VS "RIGHT"

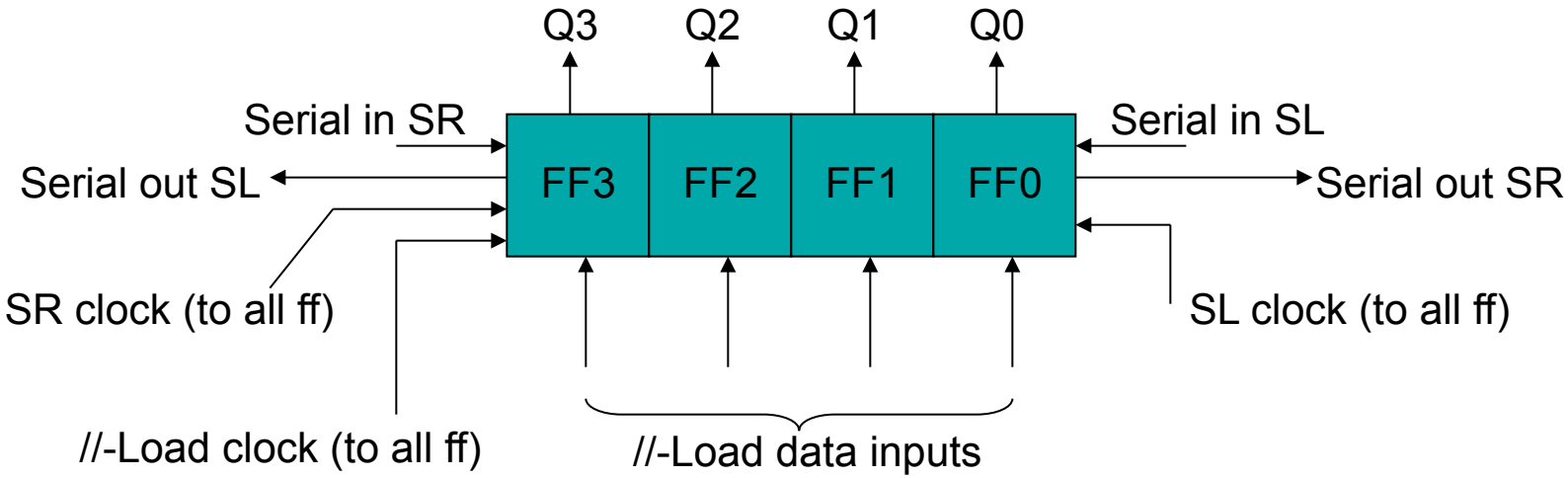
- What do we mean by a “shift-left register”?
- No matter which way we draw them, "left" should mean "from least significant bit position to most significant".
- Redrawing:



Here the serial transfer convention is “big end first”.

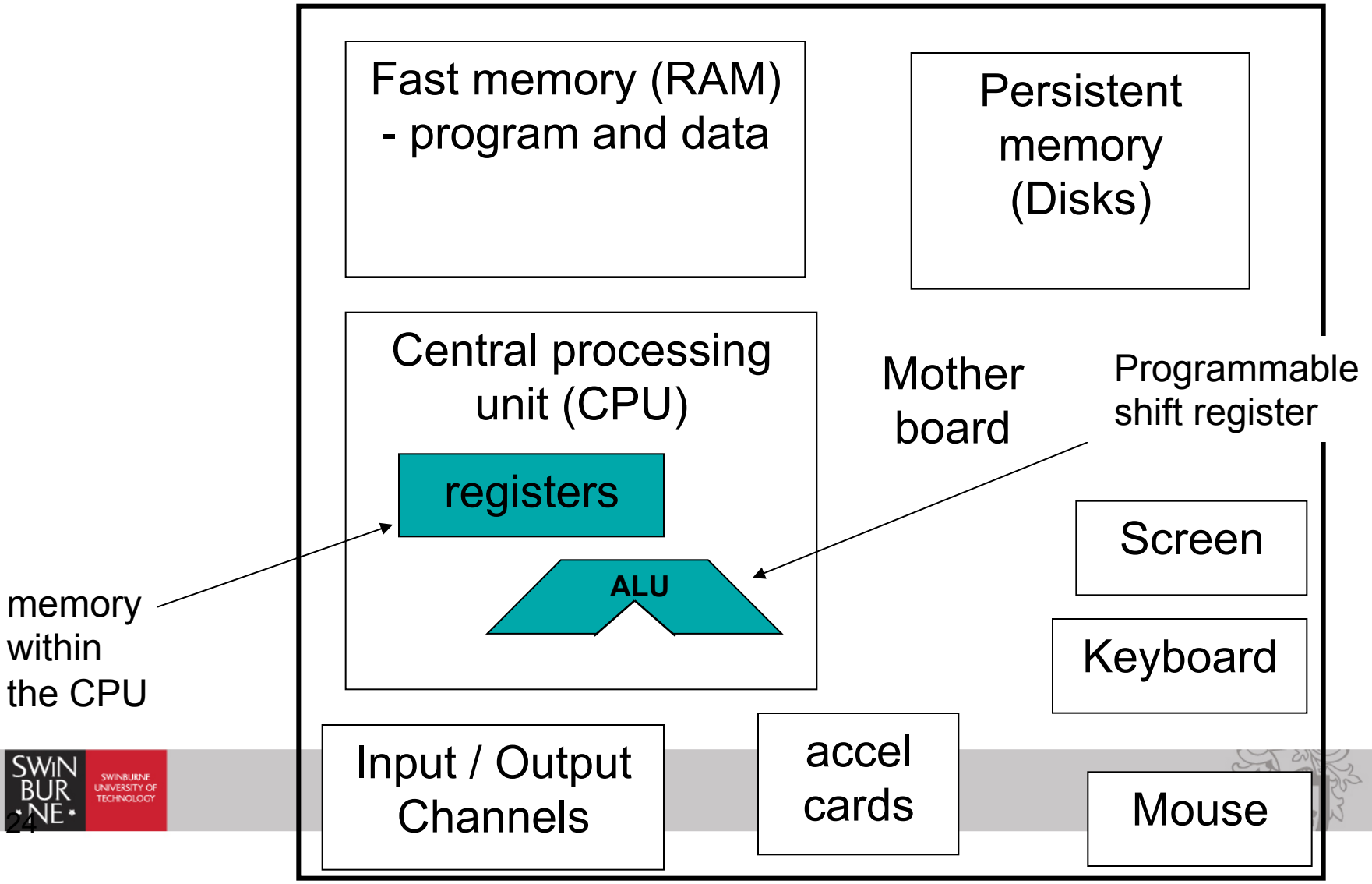
THE COMPLETE SHIFT REGISTER.

- To perform all these actions a shift register may need to be able to shift left, shift right and parallel load. It may have three different clock inputs to perform these actions as shown.



The main register in an UART (universal asynchronous receiver/transmitter) is a shift register like this (except more than 4 bits long)

WHERE ARE WE?


Fast memory (RAM)
- program and data
Persistent
memory
(Disks)
Central processing
unit (CPU)
registers
ALU
Mother
board
Programmable
shift register
memory
within
the CPU
Screen
Keyboard
Input / Output
Channels
accel
cards
Mouse

THINGS TO REMEMBER (FILL THIS IN)

Flip-Flop	Uses	Inputs	Clocked?	Circuit Elements (Gates)
RS				
D				
JK				
T				

In the Lab...

- Build an 8-bit adder.
- Build a bunch of Flip-Flops from gates
- Test them.