(*) Sequential Logic

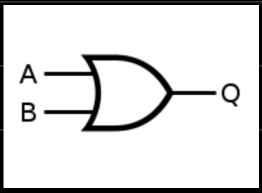
^ AFTER a circuit is CLOSED, the VALUE on a wire is said to be 0/1 when SOMETHING makes the VALUE on the wire (i.e. DRIVES the wire) to be 0/1.
For eg., (i) when a wire is DIRECTLY fixed to 0/1, then the POWER SUPPLY makes the VALUE on the wire to be 0/1, (ii) when a wire connects the OUTPUT of an OR gate to the INPUT of a NOT gate, then the OR gate makes the VALUE on the wire (and consequently, the VALUE of the NOT gate's INPUT) to be 0/1, etc.

However, AFTER a circuit is CLOSED, when NOTHING makes the VALUE on a wire to be 0/1, then the VALUE on the wire is said to be FLOATING.
A FLOATING value MAY be 0, 1 or somewhere between 0 & 1, or it MAY be oscillating among 0, 1 and somewhere between 0 & 1.

A FLOATING value which is FLOATING at 0/1 is ALSO considered to be 0/1.
So, for eg., if a CERTAIN behaviour occurs due to the VALUE on a wire being 0/1, then the SAME behaviour will occur if the VALUE on that wire is FLOATING at 0/1.

AFTER a circuit is CLOSED, while the initial PROPAGATIONS have NOT completed, the VALUES on the wires (except for those which are DIRECTLY fixed to 0/1) will be FLOATING (at 0, 1 or somewhere between 0 & 1, or oscillating among 0, 1 and somewhere between 0 & 1), including the VALUE on the wire which is connected to the OUTPUT of the MASTER–CLOCK generator (discussed later).

In GENERAL, for EVERY logic gate, the VALUE on the wire which is connected to its OUTPUT will become 0/1 only AFTER the VALUE(s) on the wire(s) which is/are connected to its INPUT(s) become(s) 0/1 (including FLOATING at 0/1) and AFTER its PROPAGATION delay.



For eg., if the PROPAGATION DELAY of this OR gate is 20 ns, if A & B are FIXED to 1's, and if the circuit is CLOSED at t = 0, then Q will be FLOATING from t = 0 to t = 20 and it will be 1 AFTER t = 20.

Also, if the VALUE(s) on the wire(s) which is/are connected to the INPUT(s) of a logic gate become(s) FLOATING somewhere between 0 & 1, then AFTER its PROPAGATION DELAY, the VALUE on the wire which is connected to its OUTPUT will become FLOATING (at 0, 1 or somewhere between 0 & 1, or oscillating among 0, 1 and somewhere between 0 & 1) if NOTHING ELSE is making the VALUE on that wire to be 0/1.

For eg., if the VALUE on the wire which is connected to ONE input of an OR gate becomes 1 (including FLOATING at 1), then after its PROPAGATION delay, the VALUE on the wire which is connected to its OUTPUT will become 1 IRRESPECTIVE of the VALUE on the wire which is connected to its OTHER input (including FLOATING (at 0, 1 or somewhere between 0 & 1, or oscillating among 0, 1 and somewhere between 0 & 1)).
However, if the VALUE on the wire which is connected to ONE input of an OR gate becomes 0 (including FLOATING at 0) and the VALUE on the wire which is connected to its OTHER input becomes FLOATING somewhere between 0 & 1, or if the VALUES on BOTH of the wires become FLOATING somewhere between 0 & 1, then after its PROPAGATION delay, the VALUE on the wire which is connected to its OUTPUT will become FLOATING (at 0, 1 or somewhere between 0 & 1, or oscillating among 0, 1 and somewhere between 0 & 1) if NOTHING ELSE is making the VALUE on that wire to be 0/1.

An INPUT of a LOGIC GATE will be FLOATING (at 0, 1 or somewhere between 0 & 1, or oscillating among 0, 1 and somewhere between 0 & 1) if

1. it is DISCONNECTED,
2. the VALUE on the wire which is connected to THIS input is FLOATING (at 0, 1 or somewhere between 0 & 1, or oscillating among 0, 1 and somewhere between 0 & 1),

Etc.

A VALUE on a wire which is FLOATING MAY float at 0/1 sometimes, but an ACTUAL 0/1 on a wire will REMAIN constant until SOMETHING changes the VALUE on that wire.

For eg., if the VALUE on the wire which is connected to the OUTPUT of a NOT gate is currently 0, and if the VALUE on the wire which is connected to its INPUT becomes 0, then AFTER its PROPAGATION delay, the VALUE on the wire which is connected to its OUTPUT will become 1. In this case, the NOT gate will change the VALUE on the wire which is connected to its OUTPUT.

As opposed to an ACTUAL 0/1 on a wire, a FLOATING value on a wire MAY get changed ARBITRARILY.

Thus, for eg., the VALUE on the wire which is connected to the OUTPUT of a MASTER-CLOCK generator (discussed later) PERIODICALLY oscillates between 0 & 1 due to the MASTER-CLOCK generator changing the VALUE on the wire which is connected to its OUTPUT.

This is DIFFERENT from a VALUE on a wire which is FLOATING, which MAY oscillate among 0, 1 and somewhere between 0 & 1 ARBITRARILY.

Also, when the VALUE on a wire gets CHANGED from 0/1 to 1/0, it goes from 0/1 to somewhere between 0 & 1, and then to 1/0.

So, even though the VALUE on the wire TEMPORARILY becomes somewhere between 0 & 1, this TRANSITIONING phase is DIFFERENT from FLOATING.

^ The outputs of a COMBINATIONAL circuit at ANY instant of time are ENTIRELY dependent upon the inputs present at THAT time, taking into account the PROPAGATION delays.

However, in a SEQUENTIAL circuit, MEMORY elements are present as well.
The information stored in the MEMORY elements at any given time defines the STATE of the sequential circuit at THAT time.
The external OUTPUTS and the next STATE of a sequential circuit are both functions of the external INPUTS and the present STATE.
Thus, a sequential circuit is specified by a time sequence of EXTERNAL INPUTS, EXTERNAL OUTPUTS and STATES.

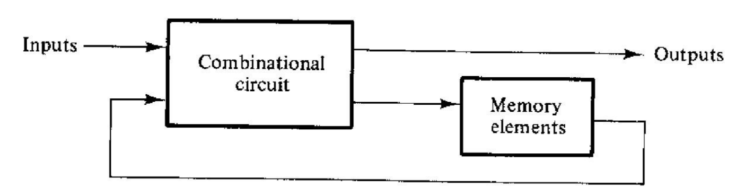


FIGURE 6-1
Block diagram of a sequential circuit

^ A SYNCHRONOUS sequential circuit is a system whose BEHAVIOUR can EASILY be defined from the knowledge of its signals at DISCRETE instants of time.
The BEHAVIOUR of an ASYNCHRONOUS sequential circuit depends upon the ORDER in which its input signals CHANGE and can be affected at ANY instant of time.
So, the BEHAVIOUR of an ASYNCHRONOUS sequential circuit becomes DIFFICULT to be defined at DISCRETE instants of time, and an ASYNCHRONOUS sequential circuit MAY, at times, become UNSTABLE.

^ The MEMORY elements used in ASYNCHRONOUS sequential circuits are TIME-DELAY devices, whose memory capabilities are due to the FINITE amounts of time signals take to PROPAGATE through devices.
Instead of using PHYSICAL time-delay devices, LOGIC GATES may also be used to produce SIMILAR effects due to their internal PROPAGATION delays.

^ In SYNCHRONOUS sequential circuits, inputs are allowed to affect the MEMORY elements ONLY at DISCRETE instants of time. ONE way of achieving this goal is by using a MASTER-CLOCK generator, and a circuit which uses a MASTER-CLOCK generator is known as a CLOCKED SYNCHRONOUS sequential circuit.

A MASTER-CLOCK generator is a TIMING device which generates a PERIODIC train of CLOCK PULSES, i.e. it makes the VALUE on the wire which is connected to its OUTPUT PERIODICALLY oscillate between 0 & 1.
This PERIODICALLY oscillating signal becomes the MAIN CLOCK of a circuit.

In CLOCKED SYNCHRONOUS sequential circuits, EVERY MEMORY element is affected ONLY with the ARRIVAL/DEPARTURE of a pulse, i.e. during the POSITIVE/NEGATIVE edge-transition of the CLOCK input, depending upon the implementation, and NOT during the ENTIRE time the CLOCK input stays at 1/0.

It should be noted that 'DURING the POSITIVE/NEGATIVE edge-transition of the CLOCK input' means 'IMMEDIATELY AFTER the CLOCK input goes from 0/1 to 1/0'.

The MEMORY elements used in CLOCKED SYNCHRONOUS sequential circuits are MODIFIED GATED LATCHES and FLIP-FLOPS, which are both EDGE-TRIGGERED.

^ The DIFFERENCE between a LATCH and a FLIP-FLOP is that a LATCH is LEVEL-SENSITIVE, i.e. the outputs of a LATCH respond to NEW inputs AT ALL TIMES, whereas a FLIP-FLOP is EDGE-TRIGGERED, i.e. the outputs of a FLIP-FLOP respond to NEW inputs ONLY during the positive/negative EDGE-TRANSITIONS of its CLOCK input.

Hence, in addition to TIME-DELAY devices, LATCHES are also examples of MEMORY elements used in ASYNCHRONOUS sequential circuits.

^ Adding a CLOCK input to a LATCH will NOT make it a FLIP-FLOP, as its outputs will respond to NEW inputs during the ENTIRE time its CLOCK input will stay at 1/0, and will rather make it a GATED LATCH.
Since GATED LATCHES are also LEVEL-SENSITIVE, therefore they are MODIFIED in order to become EDGE-TRIGGERED, for eg., by using EDGE-DETECTORS on their CLOCK inputs, so that they can be used in CLOCKED SYNCHRONOUS sequential circuits.

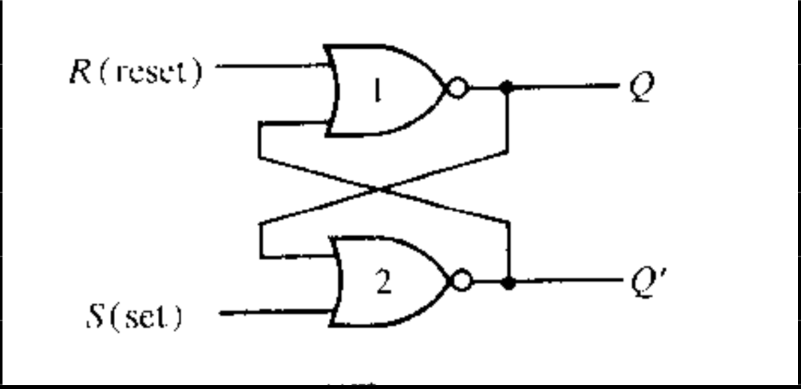
^ A LATCH / GATED LATCH / FLIP-FLOP is a MEMORY element (CELL) capable of storing 1 BIT of information, and can MAINTAIN its state INDEFINITELY (as long as power is delivered to the circuit) UNTIL directed by its input(s) to SWITCH states.

Every LATCH / GATED LATCH / FLIP-FLOP has 2 outputs, Q & Q'.
When Q = 1 & Q' = 0, the state is SET, and when Q = 0 & Q' = 1, the state is CLEAR.

Since a LATCH / GATED LATCH / FLIP-FLOP has 2 STABLE states, therefore it is also known as a BISTABLE MULTIVIBRATOR.

^ NOR-Gate SR Latch (also known INCORRECTLY as a DIRECT-COUPLED SR/RS Flip-Flop)

(See Building-an-8-bit-Breadboard-Computer.pdf (NOR-Gate SR Latch) before moving on)



After the circuit is CLOSED and after the initial PROPAGATIONS get completed, the following EXTREME cases MAY occur -

1. Let S, R, Q & Q' all be FLOATING somewhere between 0 & 1.

   Now, if S & R both become 0, even then Q & Q' both will remain FLOATING somewhere between 0 & 1.

   This behaviour is SIMILAR to the one discussed earlier for an OR gate.

2. Let S & R both be FLOATING somewhere between 0 & 1, and let Q & Q' both be FLOATING at 0.

   Now, if S & R both become 0 SIMULTANEOUSLY, then Q & Q' both will IDEALLY start OSCILLATING between 0 & 1, even if the PROPAGATION delays of the 2 NOR gates are DIFFERENT.

   This behaviour is SIMILAR to S & R both becoming 1 for a LONG ENOUGH time for the PROPAGATIONS to get completed, making Q & Q' both become 0, and then S & R both becoming 0 SIMULTANEOUSLY (discussed in Building-an-8-bit-Breadboard-Computer.pdf).

So, in EXTREME cases, after the circuit is CLOSED, after the initial PROPAGATIONS get completed and after S & R both become 0, the Q & Q' outputs of a NOR-Gate SR latch MAY remain FLOATING somewhere between 0 & 1, or they MAY start OSCILLATING between 0 & 1.

In REAL–WORLD circuits, such EXTREME cases RARELY occur, because

1. After the circuit is CLOSED and after the initial PROPAGATIONS get completed, BUT before S & R both become 0, if S & R become FLOATING at 1/0 & 0/1, respectively, for a LONG ENOUGH time for the PROPAGATIONS to get completed, instead of FLOATING somewhere between 0 & 1, then the latch will result in the SET/CLEAR state.
2. After the circuit is CLOSED, after the initial PROPAGATIONS get completed and after S & R both become 0, if Q & Q' become FLOATING at 1/0 and 0/1, respectively, for a LONG ENOUGH time for the PROPAGATIONS to get completed, instead of FLOATING somewhere between 0 & 1, then the latch will result in the SET/CLEAR state.
3. After the circuit is CLOSED, after the initial PROPAGATIONS get completed and after S & R both become 0, even if Q & Q' both start OSCILLATING between 0 & 1, they MAY NOT keep oscillating FOREVER and the latch MAY eventually result in the SET/CLEAR state after a FEW cycles of OSCILLATIONS, due to the TIME it takes to turn on and off TRANSISTORS, to charge internal CAPACITORS, etc.

   In other words, in REAL–WORLD circuits, a CHANGE in the inputs for a SMALL enough time does NOT contain enough ENERGY to cause a LOGIC GATE to SWITCH its output.

So, the STARTUP BEHAVIOUR of a NOR–Gate SR latch can be described as follows –
After the circuit is CLOSED and after the initial PROPAGATIONS get completed, the Q & Q' outputs of the latch will be ARBITRARY, i.e. 0, 1, OSCILLATING between 0 & 1 or FLOATING somewhere between 0 & 1, even after S & R both become 0.

ALMOST all LATCHES / GATED LATCHES / FLIP–FLOPS have similar ARBITRARY startup behaviours.

The ARBITRARY startup behaviour of a LATCH / GATED LATCH / FLIP–FLOP does NOT matter, as every LATCH / GATED LATCH / FLIP–FLOP is made to go into the SET/CLEAR state before it is used, for eg., by using DIRECT PRESET/CLEAR inputs (discussed later).

Under NORMAL operation, the S & R inputs of a NOR-Gate SR latch both REMAIN 0 unless the STATE of the latch has to be CHANGED.

1. Let S become 1 and let R remain 0.

   IRRESPECTIVE of the CURRENT values of Q & Q' (i.e. 0, 1, OSCILLATING between 0 & 1 or FLOATING somewhere between 0 & 1), the latch WILL result in the SET state (i.e. if Q & Q' were ALREADY 1 & 0, respectively, then they will REMAIN 1 & 0, and if Q & Q' were NOT 1 & 0, respectively, then they will BECOME 1 & 0) after a LONG ENOUGH time for the PROPAGATIONS to get completed.

   After the latch results in the SET state, S can become 0 again and the latch will REMAIN in the SET state.

2. Let S remain 0 and let R become 1.

   IRRESPECTIVE of the CURRENT values of Q & Q' (i.e. 0, 1, OSCILLATING between 0 & 1 or FLOATING somewhere between 0 & 1), the latch WILL result in the CLEAR state (i.e. if Q & Q' were ALREADY 0 & 1, respectively, then they will REMAIN 0 & 1, and if Q & Q' were NOT 0 & 1, respectively, then they will BECOME 0 & 1) after a LONG ENOUGH time for the PROPAGATIONS to get completed.

   After the latch results in the CLEAR state, R can become 0 again and the latch will REMAIN in the CLEAR state.

3. Let S & R both become 1.

   IRRESPECTIVE of the CURRENT values of Q & Q' (i.e. 0, 1, OSCILLATING between 0 & 1 or FLOATING somewhere between 0 & 1), the latch WILL result in an INVALID state with Q & Q' both becoming 0 after a LONG ENOUGH time for the PROPAGATIONS to get completed.

   After the latch results in this INVALID state, if S & R both become 0 SIMULTANEOUSLY, then Q & Q' both will IDEALLY start OSCILLATING between 0 & 1, even if the PROPAGATION delays of the 2 NOR gates are DIFFERENT (discussed earlier).

The LONG ENOUGH time for the PROPAGATIONS to get completed is equal to the SUM of the PROPAGATION delays of the two NOR gates (discussed in Building-an-8-bit-Breadboard-Computer.pdf).

The INVALID state of Q & Q' both becoming 0 VIOLATES the fact that Q & Q' are complements of each other.
Thus, under NORMAL operation, this condition MUST be AVOIDED by making sure that S & R both DON'T become 1 SIMULTANEOUSLY.

However, if S & R both become 1 SIMULTANEOUSLY for a LONG ENOUGH time, making the latch result in the INVALID state, due to a GLITCH, then also setting S & R to 1/0 & 0/1, respectively, will make the LATCH result in the SET/CLEAR state after a LONG ENOUGH time.
For eg., this GLITCH occurs in a GATED D latch due to the PROPAGATION delay of the NOT gate.

A GOOD design ENSURES that a digital ciruit behaves PROPERLY even though these kinds of GLITCHES may happen.

For POSITIVE edge-triggered flip-flops,

1. The SETUP time is defined as the DURATION of time for which the input(s) must be kept CONSTANT while CP is at 0 IMMEDIATELY BEFORE CP gets changed from 0 to 1.
2. The HOLD time is defined as the DURATION of time for which the input(s) must be kept CONSTANT while CP is at 1 IMMEDIATELY AFTER CP gets changed from 0 to 1.

For eg., if the DURATION of a clock cycle is 20 units, with the clock staying at 1 & 0 for 10 units each and the clock transitioning from 0 to 1 at t = 20 T, where T is an integer, and if the SETUP & HOLD times of a flip-flop are 2 units & 1 unit, respectively, then the input(s) must be kept CONSTANT from t = (20 T − 2) to t = (20 T + 1).

It should be noted that the ARRIVAL of the input(s) EXACTLY at t = (20 T − 2) and the CHANGING of the inputs EXACTLY at t = (20 T + 1) may work IDEALLY, but MAY or MAY NOT work in REAL-WORLD circuits. So, the input(s) must arrive slightly BEFORE t = (20 T − 2) and must be allowed to change only slightly AFTER t = (20 T + 1).

For GATED latches and NEGATIVE edge-triggered flip-flops,

1. The SETUP time is defined as the DURATION of time for which the input(s) must be kept CONSTANT while CP is at 1 IMMEDIATELY BEFORE CP gets changed from 1 to 0.
2. The HOLD time is defined as the DURATION of time for which the input(s) must be kept CONSTANT while CP is at 0 IMMEDIATELY AFTER CP gets changed from 1 to 0.

In a circuit, the DURATION of a clock cycle is made LONG ENOUGH for signals to PROPAGATE and for the ENTIRE circuit to become STABLE before the MAIN CLOCK gets changed from 0/1 to 1/0.

The data to be loaded into GATED LATCHES / FLIP-FLOPS arrive at the inputs and remain CONSTANT such that the SETUP time and the HOLD time constraints are NOT violated.

^

^

In a REAL—WORLD computer, when the POWER SUPPLY is turned on, a very SMALL circuit gets completed whose job is to DETECT when the POWER BUTTON is pressed.
AFTER the POWER BUTTON is pressed, power gets supplied to the ENTIRE circuit of the computer.

IMMEDIATELY AFTER the POWER BUTTON is pressed, the MAIN CLOCK starts (AFTER the initial PROPAGATIONS within the MASTER—CLOCK generator get completed), and a POWER—ON—RESET generator starts sending ASYNCHRONOUS RESET signals to CLEAR ALL of the FLIP—FLOPS.
After a FEW clock cycles, when the power STABILIZES, the initial PROPAGATIONS for the ENTIRE computer get completed and ALL of the FLIP—FLOPS get CLEARED, the POWER—ON—RESET GENERATOR stops sending the ASYNCHRONOUS RESET signals, and the NORMAL operation of the computer begins from the NEXT clock cycle.

The DE—ASSERTION of the ASYNCHRONOUS RESET signals is generally SYNCHRONISED using a RESET SYNCHRONISER circuit in order to adhere to the RECOVERY time and the REMOVAL time constraints.