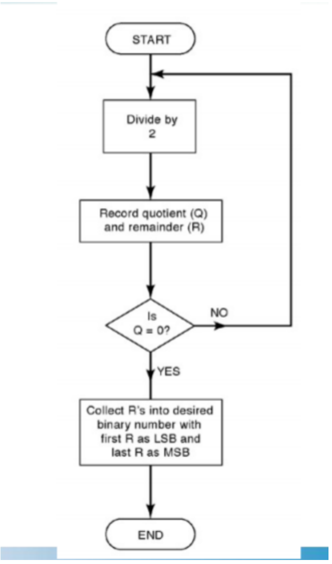
DIGITAL SYSTEM

1 Digital and Analog Systems

- **Digital System** is a combination of devices that manipulate values represented in digital form.
- **Analog System** is a combination of devices that manipulate values represented in analog form.

Analog-to-digital conversion(ADC) and digital-to-analog conversion(DAC) complicate circuitry


START
Divide by 2
Record quotient (Q) and remainder (R)
Is Q = 0?
NO
YES
Collect R's into desired binary number with first R as LSB and last R as MSB
END

2 Digital Number Systems

- Number system differ in the number of symbols they use

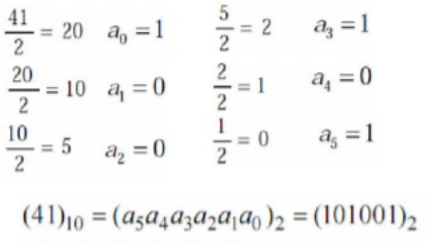
- Decimal - 10 symbols (base 10)
- Hexadecimal - 16 symbols (base 16)
- Octal - 8 symbols (base 8)
- Binary - 2 symbols (base 2)

3 Conversion

Binary to Decimal Conversion

1 0 0 1 0 1_2
5 4 3 2 1 0 We consider that:
$(100101)_2 = 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (37)_{10}$

Decimal to Binary Conversion



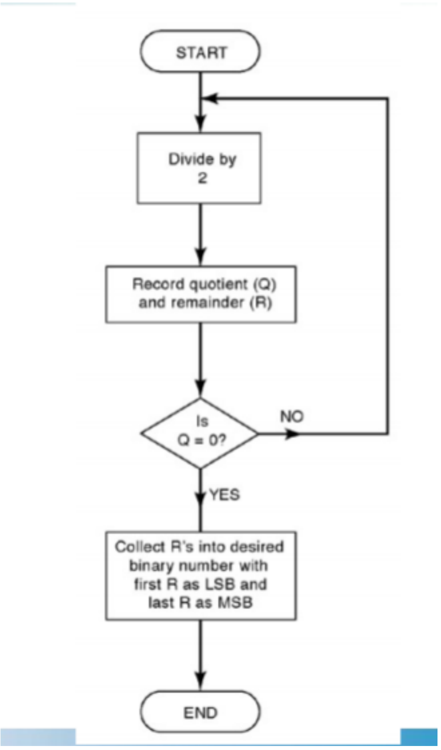
$\frac{41}{2} = 20 \quad a_0 = 1 \qquad \frac{5}{2} = 2 \quad a_3 = 1$

$\frac{20}{2} = 10 \quad a_1 = 0 \qquad \frac{2}{2} = 1 \quad a_4 = 0$

$\frac{10}{2} = 5 \quad a_2 = 0 \qquad \frac{1}{2} = 0 \quad a_5 = 1$

$$(41)_{10} = (a_5 a_4 a_3 a_2 a_1 a_0)_2 = (101001)_2$$

Flow chart:


START
Divide by 2
Record quotient (Q) and remainder (R)
Is Q = 0?
NO
YES
Collect R's into desired binary number with first R as LSB and last R as MSB
END

Convert from Hexadecimal to Decimal

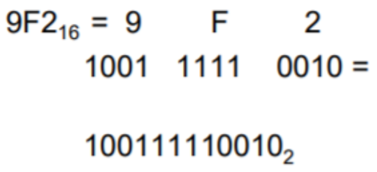
- Convert from Hexadecimal to decimal by multiplying each hex digit by its positional weight

$$\begin{aligned} 163_{16} &= 1\times(16^{2})+6\times(16^{1})+3\times(16^{0}) \\ &= 1\times 256+6\times 16+3\times 1 \\ &= 355_{10} \end{aligned}$$

Hexadecimal to Decimal Conversion

- Convert form Hexadecimal to Decimal by deviding the decimal number by 16
Same method to convertion from binary to decimal

Hexadecimal to Binary Conversion



$9F2_{16}$ = 9 F 2
1001 1111 0010 =

100111110010_2

Binary to Hexadecimal Convertion

- To convert Binary to Hexadecimal, first we should group bits in four, then we convert each group to hexadecimal.
Ex:
$1110100110_2 = 0011\ 1010\ 0110$
$= 3 \quad A \quad 6$
$= 3A6_{16}$

4 BCD

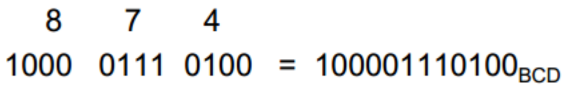
- **Binary Coded Decimal** (BCD) is a another way to present decimal numbers in binary form.

- BCD is widely uses and combines features of both decimal and binary systems.
- Each digit is converted to a binary equivalent.

Bảng 1: BCD Table

Decimal	**BCD**
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

- To convert the number 874_{10} to BCD:



8 7 4

1000 0111 0100 = 100001110100_{BCD}

- Each decimal digit is represented using 4 bits
- Each 4-bit group can never be greater than 9.
- Reverse the process to convert BCD to decimal.

5 Flip Flop

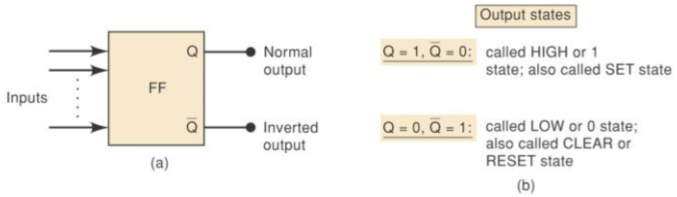
- So far we has seen Combinational Logic that the output(s) depends only on the value of the input variables.
- Here we look at Sequential Logic circuit that the output(s) can depend on

present and also past values of the input and output variables.
- Sequential circuits exist only in one of a defined number of state at any one time.

5.1 Synchronous and Asynchronous Sequence Logic:

- Synchronous:
  - The timing of all state transitions is controlled by a common clock.
  - Changes in all variables occur simultaneously.
- Asynchronous:
  - State transistions occur independently of any clock and normally dependent on the timing of transitions in the input variables.
  - Changes in more than one output do not neccessarily occur simultaneously.
- Clock:
  - A clock signal is a square wave of fixed frequency.
  - Often, transitions will occur on one of the clock pulses

5.2 General Flip Flop Symbol and definition of its two possible output states


Inputs
FF
Q
$\bar{Q}$
Normal output
Inverted output
(a)
Output states
$Q = 1, \bar{Q} = 0$: called HIGH or 1 state; also called SET state
$Q = 0, \bar{Q} = 1$: called LOW or 0 state; also called CLEAR or RESET state
(b)

- The Flip Flop, abbreviated FF, is a key memory element.
- The output of the Flip Flop is Q and $\bar{Q}$.
- Q is understood to be the normal output, $\bar{Q}$ is always the opposite.
- The most basic circuit can be constructed from either two NAND gates or NOR gates.

5.3 NAND gate latch

- The inputs are **set** and **clear** (reset).
- The inputs are active low, that is, the output will change when the input is pulsed low.
- When the latch is set:

$$\text{Q} = 1 \text{ and } \bar{Q} = 0$$

- When the latch is clear or reset:

$$\text{Q} = 0 \text{ and } \bar{Q} = 1$$

Summary of NAND latch

- SET = RESET = 1: Normal resting state, outputs remain in state prior to output.
- SET = 0, RESET = 1: Q will go high and remain high even if the SET input goes high.
- SET = 1, RESET = 0: Q will go low and remain low even if the SET input goes high.
- SET = RESET = 0: Output is unpredictable because the latch is being set and reset at the same time.

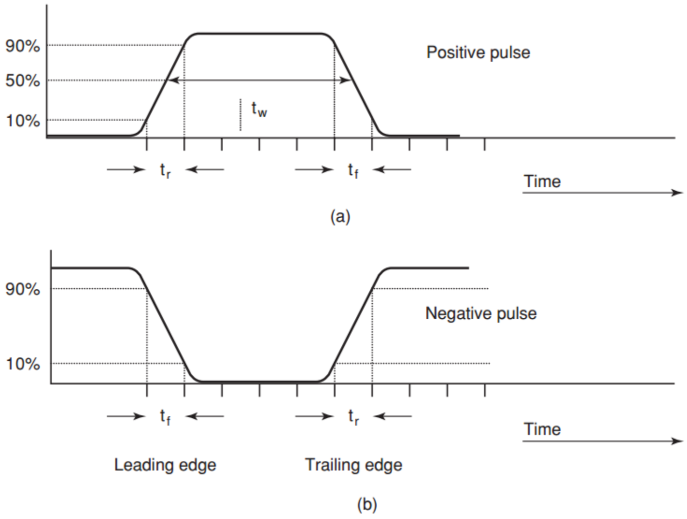
5.4 NOR Gate Latch

- The NOR Gate Latch is similar to the NAND Gate Latch, except that the output Q and $\bar{Q}$ is reserved.
- **The inputs are active high**, that is, the outputs will change only the inputs **are pulsed high**.
- In order to ensure that a FF begins operation at a known level, a pulse may be applied to SET and RESET inputs when a device is powered up.

5.5 Digital Pulses

- In digital system, when a signal switches from a normal inactive state to the oposite (active) state, thus causing something happen in the circuit. Then the signal returns to its inactive state while the effect of the recently activated signal remains in the system. These signals are called pulses.

FIGURE 5-14 (a) A positive pulse and (b) a negative pulse.



- A pulse that performs its intended function when it goes HIGH is called a positive pulse.
- A pulse that performs its intended funtion when it goes LOW is called a negative pulse.
- The transition from low to high on a positive pulse is called rise time (t_r).
- The transition from high to low on a positive pulse is called fall time (t_f).
- The transition from the beginning of the pulse is called the leading edge and the transition at the end of the pulse is called the trailing edge.

5.6 Clock Signal and CLocked Flip Flop

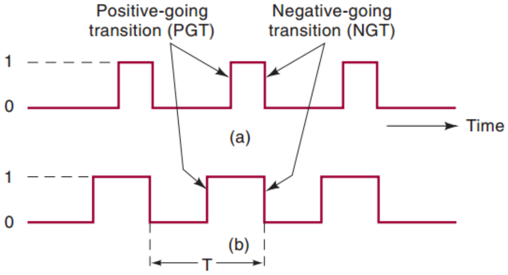
- Digital system can be operate either asynchronously and synchoronously:

- In asynchronous system, the outputs of circuits can change state anytime one or more of the inputs change.

- In synchronous system, the outputs can change states at the exact time by a signal called clock. This clock signal is generally a rectangular pulse train or a square wave.

Clock Signal

- Most (if not all) of the system outputs can change state only when the clock make a transition.


Positive-going transition (PGT)
Negative-going transition (NGT)
1
0
(a)
Time
1
0
(b)
T

- A transition (also called *edges*):

- When the clock changes from 0 to 1, this is called the **positive-going transition (PGT)**.

- When the clock changes from 1 to 0, this called the **negative-going transition (NGT)**.

- Most digital system are pricipally synchronous (although there are always some asynchronous parts) because synchronous circuits are easier to design and troubleshoot.

Clocked Flip Flop

- CLocked FFs can change state on one or other clock transitions. Some common characteristics:
  - CLock inputs are labeled as CLK, CK, or CP.
  - A **small triangle** at the CLK input indicates that CLK input is

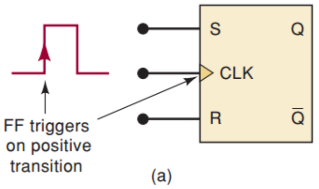
activated with a PGT (positive-going transition).
- A **bubble and a triangle** at the CLK input indicates that CLK input is activated with a NGT (negative-going transition).
- Control inputs have an effect on the output only at the active clock transition (PGT and NGT). There are also called synchronous control inputs.
- The control inputs get the FF outputs ready to change, but the change is not triggered until the CLK edge.

Setup Time and Hold Time

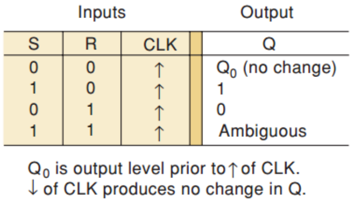
- Setup time (t_s) is the minimum time interval before the active CLK transition that the control input must be kept at the proper level.
- Hold time (t_h) is the time after the active CLK transition during which the control input must be kept at the proper level.

5.7 Clocked S-R Flip Flop

- In Clocked S-R Flip Flop system, FF only change states when when a signal applied to its clock input makes a transition from 0 to 1 or from 1 to 0 (PGT and NGT).
- The up arrow (↑) indicates that a PGT is required at CLK; the label Q_0 indicates the level at Q prior to the PGT.



(a)

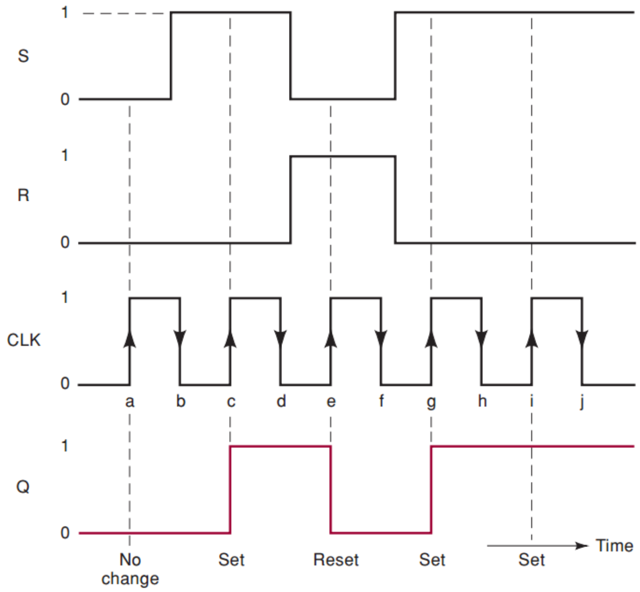


Inputs			Output
S	R	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

Q_0 is output level prior to ↑ of CLK.
↓ of CLK produces no change in Q.

(b)

- We can analyze these waveforms as follows:



(c)

1. Initiallly all inputs are 0 and the Q output is assumed to be 0; that is $Q_0 = 0$
2. When the PGT of the first clock pulse occurs (point a), the S and R inputs are both 0, so the FF is not effected and remains in the Q = 0 state (Q = Q_0)
3. When the PGT of the second clock pulse (point c), the S = 1 and R = 0. Thus the FF set to 1 state at the rising edge of this clock pulse.
4. When the third clock pulse makes its positive transition (point e) S = 0 and R = 1. Thus it causes the FF set to 0 state.
5. At the fourh pulse set the FF once again to the Q = 1 state (point g) because S = 1 and R = 0 when PGT occurs
6. The fifth pulse also find that S = 1 and R = 0 at PGT occurs so the FF remains state.
7. The case that S = R = 1 should not be used because it results in an ambiguous condition.

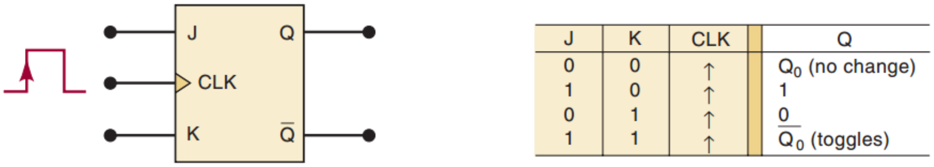
8. It should be noted that those waveforms that the FF is **not effected** by the NGT of the clock pulses. Also note that the S and R levels have no effects on the FF, except upon the occurence of the PGT **in PGT transitions**

9. For the NGT transition the FF operate at the same manner at PGT transition, except the output can change states only on the falling edge of the clock pulse

- As a general rule for stable flip flop triggering, the clock pulse rise and fall time must be very short, around typically 2 - 5 ns.

5.8 Clocked J-K Flip Flop

- The J and K inputs control the state of the FF in the same way as the S and R inputs do for the clocked S-R flip flop, except 1 difference that when J = K = 1 condition **does not result** in ambiguous output. In this condition, the FF will always go to its opposite side of the clock signal, this is called **toggle mode** of the operation.



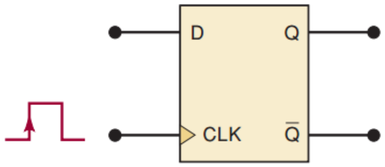
J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q_0}$ (toggles)

(a)

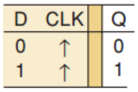
- The operation of the J-K Clocked Flip Flop is the same as the S-R Clocked Flip Flop , except when J = K = 1 at the PGT transition the output will change to the opposite state.
- Also in NGT, the operation is still same as PGT when clock is triggered in NGT.
- In edge-triggered J-K Flip Flop, CLK* must be high for FF to change states. This condition only occurs at the edge of a CLK transition

5.9 Clocked D Flip FLop

- Unlike the S-R and J-K Clocked FLip Flop, this flip flop has only one synchronous control input, *D*, which stands for data.
- The operation of the D CLocked Flip Flop is very simple: Q will go as the same state on the D input when a PGT occurs at CLK.
- A negative-edge-triggered D flip flop operates in the same way and Q will take on the value of D when a NGT occurs at CLK.



(a)

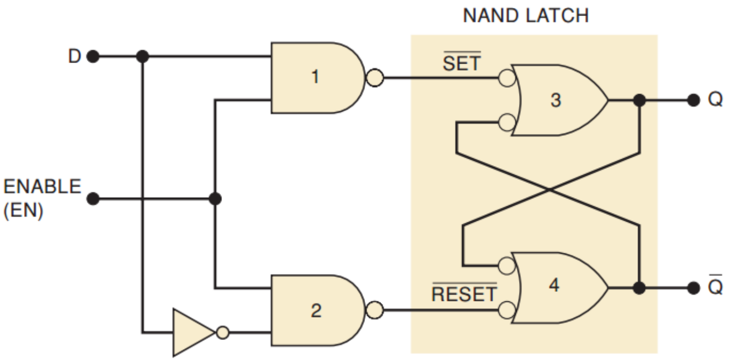


D	CLK	Q
0	↑	0
1	↑	1

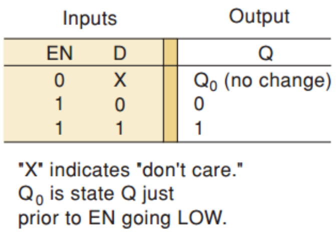
5.10 D Latch (Transparent Latch)

- The edge trigggered D Flip Flop uses an edge-detector circuit to ensure that the output will repond to the D input only when the active transition of the clock occurs. The latch that its edge-detector is not used called the D latch.

- One data input.
- The clock has been replaced by an ENABLE (EN).
- The device is not edge-triggered.
- When EN = 1; the output follows the change of input.
- When EN = 0; wheither D changes or not Q will stay at their current level (no change).



(a)

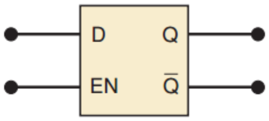


Inputs		Output
EN	D	Q
0	X	Q_0 (no change)
1	0	0
1	1	1

"X" indicates "don't care."
Q_0 is state Q just
prior to EN going LOW.

(b)

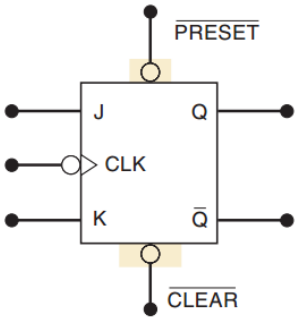
- Logic symbol of D latch:

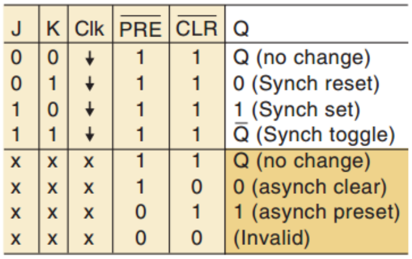


(c)

5.11 Asynchronous Inputs

- For the clocked flip flop, the S-R, J-K and D inputs have been reffered as **synchronous inputs**
- Most clocked FF also have one or more asynchronous inputs that operate independently of the synchronous inputs.
- These asynchronous inputs can be used to set FF to the 1 state of clear (reset) and FF to 0 state at **any time, regardless conditions of inputs**.
- State in another way, the asynchronous inputs are **overide inputs**





J	K	Clk	$\overline{PRE}$	$\overline{CLR}$	Q
0	0	↓	1	1	Q (no change)
0	1	↓	1	1	0 (Synch reset)
1	0	↓	1	1	1 (Synch set)
1	1	↓	1	1	$\overline{Q}$ (Synch toggle)
x	x	x	1	1	Q (no change)
x	x	x	1	0	0 (asynch clear)
x	x	x	0	1	1 (asynch preset)
x	x	x	0	0	(Invalid)

- $\overline{PRESET} = \overline{CLEAR} = 1$: The asynchronous inputs are inactive and the output depend only on CLK transitions.
- $\overline{PRESET} = 0; \overline{CLEAR} = 1$: The $\overline{PRESET}$ is activated and Q = 1 (no matter what conditions). The CLK inputs cannot affect the FF while $\overline{PRESET} = 0$.
- $\overline{PRESET} = 1; \overline{CLEAR} = 0$: The $\overline{CLEAR}$ is activated and Q = 0 (no matter what conditions). The CLK inputs cannot affect the FF while $\overline{CLEAR}$ = 0.
- $\overline{PRESET} = \overline{CLEAR} = 0$: Ambiguous response.

6 Fuzzy Logic

- Fuzzy Logic (FL) is a method of reasoning that resemble the human reasoning.
- Unlike Combinational Logic which only takes precise input and produce a definite output as 0 and 1.
- Fuzzy Logic provide the wide range of the output that surround the value of 0 and 1.

- To symlify the the result is 1, we can give the output of around 0.75 or 0.9 when the result is 0, the output can be a 0.34 or 0.2.

- The Fuzzy Logic works on the levels of possiblities of inputs to achieve the definite output.
- Fuzzy LOgic is useful for commercial and practical product.
  - It can control machines and consumer products.
  - It may not give a accurate result, but it still a acceptable result.
  - Fuzzy logic help to deal with the uncertainly in engineering.

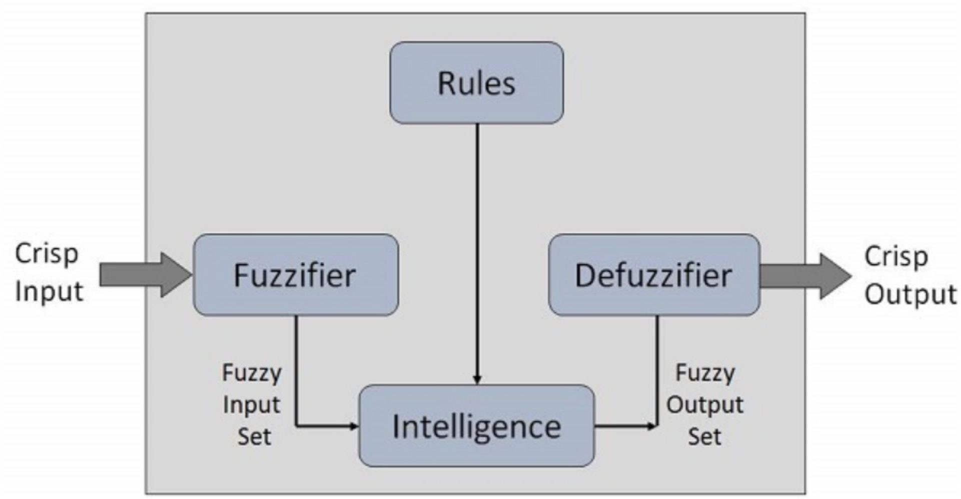
6.1 Fuzzy logic system

- It has 4 main parts:

- Fuzzification Module:
  - It transforms the system inputs, from crisp number to fuzzy set.
  - Crisp number is a single point, ex: 3, 5.5, 6; when fuzzy number is a fuzzy set with different degree os closeness to a given crisp number.
  - It splits the input signal into 5 components:

LP	x is Large Positive
MP	x is Medium Positive
S	x is small
MN	x is Medium Negative
LN	x is Large Negative

- Knowledge Base:
  - It stores IF-THEN rules.
- Inference Engine:
  - it simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN.
- Defuzzification Module:
  - It transforms the fuzzy set obtained by the inference engine into crisp value.


Rules
Crisp
Input
Fuzzifier
Defuzzifier
Crisp
Output
Fuzzy
Input
Set
Intelligence
Fuzzy
Output
Set