

CO20-320241

# **Computer Architecture and Programming Languages**

CAPL

**Lecture 5 & 6**

Dr. Kinga Lipskoch

Fall 2019

## Looping Groups of Eight (Octets) (1)

Looping an octet of adjacent 1s eliminates the three variables that appear in both complemented and uncomplemented form

	$\bar{C} \bar{D}$	$\bar{C} D$	$C \bar{D}$	$C D$
$\bar{A} \bar{B}$	0	0	0	0
$\bar{A} B$	1	1	1	1
$A \bar{B}$	1	1	1	1
$A B$	0	0	0	0

$$x = B$$

## Looping Groups of Eight (Octets) (2)

Looping an octet of adjacent 1s eliminates the three variables that appear in both complemented and uncomplemented form

	$\bar{C} \bar{D}$	$\bar{C} D$	$C \bar{D}$	$C D$
$\bar{A} \bar{B}$	1	1	0	0
$\bar{A} B$	1	1	0	0
$A \bar{B}$	1	1	0	0
$A B$	1	1	0	0

$$x = \bar{C}$$

## Looping Groups of Eight (Octets) (3)

Looping an octet of adjacent 1s eliminates the three variables that appear in both complemented and uncomplemented form

	$\bar{C} \bar{D}$	$\bar{C} D$	$C \bar{D}$	$C D$
$\bar{A} \bar{B}$	1	0	0	1
$\bar{A} B$	1	0	0	1
$A \bar{B}$	1	0	0	1
$A B$	1	0	0	1

$$x = \bar{D}$$

## Looping Groups of Eight (Octets) (4)

Looping an octet of adjacent 1s eliminates the three variables that appear in both complemented and uncomplemented form

	$\bar{C} \bar{D}$	$\bar{C} D$	$C \bar{D}$	$C D$
$\bar{A} \bar{B}$	1	1	1	1
$\bar{A} B$	0	0	0	0
$A \bar{B}$	0	0	0	0
$A B$	1	1	1	1

$$x = \bar{B}$$

## Looping Groups of Four (Quads) (1)

	$\bar{C} \bar{D}$	$\bar{C} D$	$C \bar{D}$	$C D$
$\bar{A} \bar{B}$	0 <sub>1</sub>	0	0	1
$\bar{A} B$	0 <sub>5</sub>	1	1	0
$A \bar{B}$	0 <sub>9</sub>	1	1	0
$A B$	0 <sub>13</sub>	0	1	0

$$x = \bar{A} \bar{B} \bar{C} \bar{D} + A C D + B D$$

loop 4

loop 11, 15

loop 6, 7, 10, 11

Each loop generates a term in the expression for  $x$

## Looping Groups of Four (Quads) (2)

	$\bar{C} \bar{D}$	$\bar{C} D$	$C \bar{D}$	$C D$
$\bar{A} \bar{B}$	0 <sub>1</sub>	0	1	0
$\bar{A} B$	1 <sub>5</sub>	1	1	1
$A \bar{B}$	1 <sub>9</sub>	1	0	0
$A B$	0 <sub>13</sub>	0	0	0

$$X = \bar{A} \bar{B} + B \bar{C} + \bar{A} C D$$

loop 5, 6, 7, 8

loop 5, 6, 9, 10

loop 3, 7

Each loop generates a term in the expression for  $x$

# Looping Groups

	$\bar{C} \bar{D}$	$\bar{C} D$	$C \bar{D}$	$C D$
$\bar{A} \bar{B}$	0 <sub>1</sub>	1	0	0
$\bar{A} B$	0 <sub>5</sub>	1	1	1
$A \bar{B}$	0 <sub>9</sub>	0	0	1
$A \bar{B}$	1 <sub>13</sub>	1	0	1

$$\begin{aligned}x = & \bar{A} \bar{C} \bar{D} + \bar{A} B C \\& + A \bar{B} \bar{C} + A C \bar{D}\end{aligned}$$

## Same K-map with Two Equivalent Solutions

	$\bar{C} \bar{D}$	$\bar{C} D$	$C \bar{D}$	$C D$
$\bar{A} \bar{B}$	0 <sub>1</sub>	1	0	0
$\bar{A} B$	0 <sub>5</sub>	1	1	1
$A \bar{B}$	0 <sub>9</sub>	0	0	1
$A \bar{B}$	1 <sub>13</sub>	1	0	1

$$x = \bar{A} \bar{C} \bar{D} + \bar{A} B C \\ + A \bar{B} \bar{C} + A C \bar{D}$$

	$\bar{C} \bar{D}$	$\bar{C} D$	$C \bar{D}$	$C D$
$\bar{A} \bar{B}$	0 <sub>1</sub>	1	0	0
$\bar{A} B$	0 <sub>5</sub>	1	1	1
$A \bar{B}$	0 <sub>9</sub>	0	0	1
$A \bar{B}$	1 <sub>13</sub>	1	0	1

$$x = \bar{A} B D + B C \bar{D} \\ + B \bar{C} D + A \bar{B} \bar{D}$$

## Karnaugh-Map Method (2)

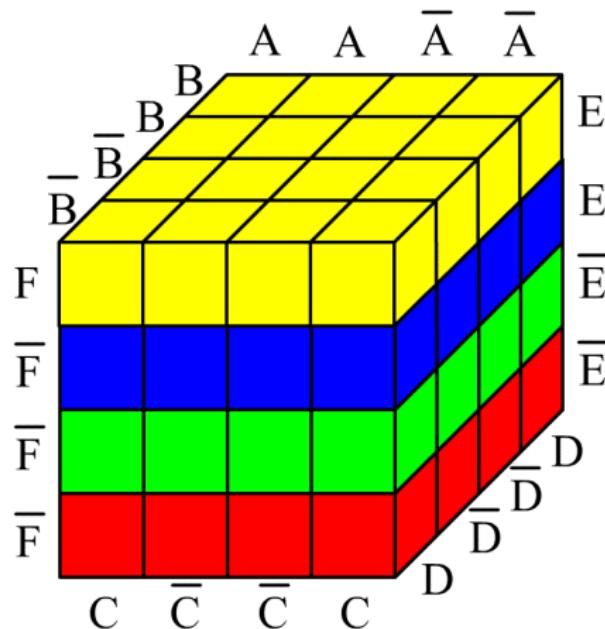
Complete K-map simplification process:

1. Construct the K-map, place 1s as indicated in the truth table
2. Loop 1s that are not adjacent to any other 1s (isolated 1s)
3. Loop 1s that are in pairs (isolated pairs)
4. Loop 1s in octets even if they have already been looped
5. Loop quads that have one or more 1s not already looped
6. Loop any pairs necessary to include 1s not already looped
7. Form the OR sum of terms generated by each loop

# Karnaugh-Map for 6 Variables (1)

		B'				B					
		E'F'	E'F	EF	EF'	E'F'	E'F	EF	EF'		
A'		C'D'	0	1	3	2	C'D'	16	17	19	18
		C'D	4	5	7	6	C'D	20	21	23	22
A		CD	12	13	15	14	CD	28	29	31	30
		CD'	8	9	11	10	CD'	24	25	27	26
A		C'D'	32	33	35	34	C'D'	48	49	51	50
		C'D	36	37	39	38	C'D	52	53	55	54
		CD	44	45	47	46	CD	60	61	63	62
		CD'	40	41	43	42	CD'	56	57	59	58

## Karnaugh-Map for 6 Variables (2)



## Exclusive OR and Exclusive NOR Circuits

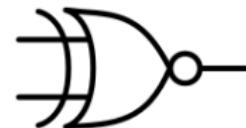
- ▶ The exclusive OR (XOR) produces a HIGH output if the two inputs are at opposite levels

$$x = \overline{A} B + A \overline{B} = A \oplus B$$

- ▶ The exclusive NOR (XNOR) produces a HIGH output if the two inputs are at the same level

$$x = A B + \overline{A} \overline{B} = \overline{A \oplus B}$$

- ▶ XOR and XNOR outputs are opposite

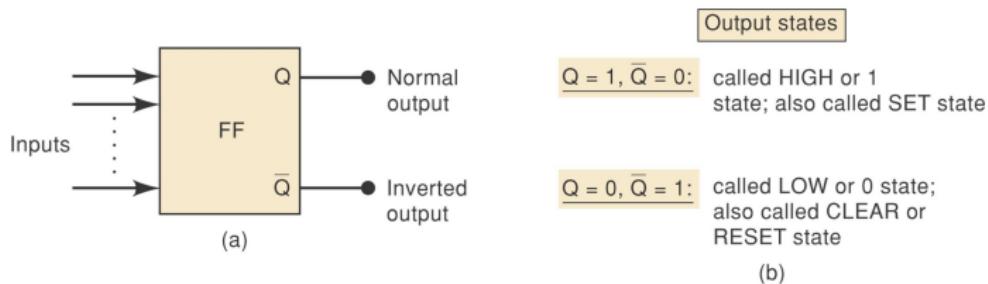


## Flip-Flops and Related Devices

- ▶ Logic circuits studied so far have outputs that respond immediately to inputs at some instant in time
- ▶ Many applications that use digital logic need to store information
- ▶ We now introduce the concept of memory: the flip-flop, abbreviated FF, is a key memory element
- ▶ The outputs of a flip flop are usually labeled  $Q$  and  $\overline{Q}$
- ▶  $Q$  is understood to be the normal output,  $\overline{Q}$  is always the opposite
- ▶ When the normal output ( $Q$ ) is placed in the high or 1 state we say the FF has been **set**
- ▶ When the normal output ( $Q$ ) is placed in the low or 0 state we say the FF has been **cleared** or **reset**

# Flip-Flops

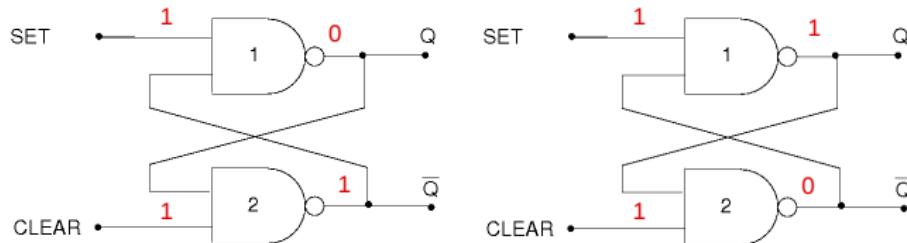
- ▶ The term **latch** is used for certain type of flip-flops
- ▶ The term **bistable multivibrator** is the technical name



# Latch

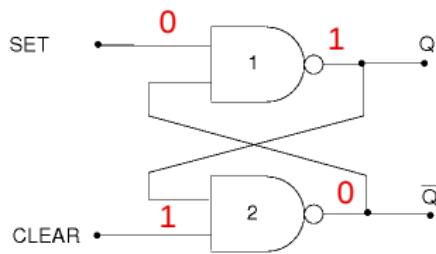
- ▶ The **latch** is a basic FF
- ▶ The inputs are labeled SET and CLEAR (RESET)
- ▶ If the output changes when the input is pulsed low, then we call this **active low**
- ▶ If the latch is set then  
 $Q = 1$  and  $\overline{Q} = 0$
- ▶ If the latch is clear or reset then  
 $Q = 0$  and  $\overline{Q} = 1$

## NAND Gate Latch



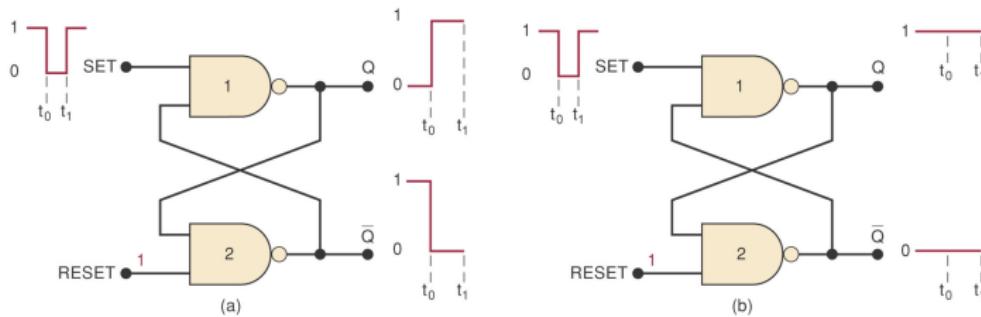
- ▶ NAND gate has two possible states if  $\text{SET} = \text{CLEAR} = 1$
- ▶ Depends on previous state of  $Q$  and  $\bar{Q}$

## Setting the NAND Gate Latch (1)



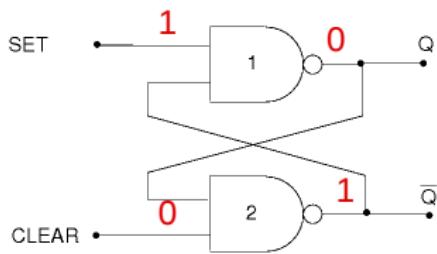
- ▶ Setting a LOW pulse on SET (while CLEAR remains HIGH) will always turn  $Q = 1$ , while  $\bar{Q} = 0$
- ▶ This operation is called **setting the latch**

## Setting the NAND Gate Latch (2)



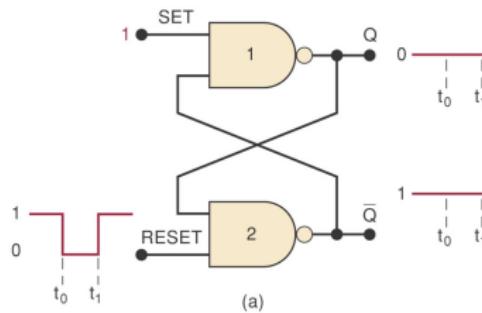
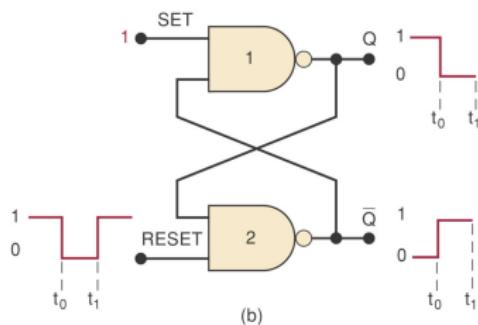
- ▶ Pulsing the SET input to the 0 state when
  - ▶  $Q = 0$  prior to SET pulse
  - ▶  $Q = 1$  prior to SET pulse
- ▶ Note that, in both cases,  $Q$  results in HIGH

## Clearing the NAND Gate Latch (1)



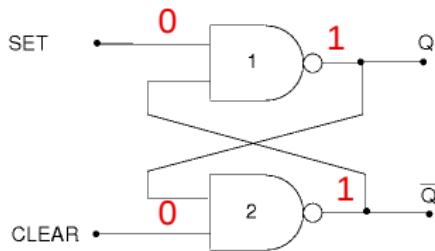
- ▶ Setting a LOW pulse on CLEAR (while SET remains HIGH) will always turn  $Q = 0$ , while  $\bar{Q} = 1$
- ▶ This operation is called **clearing** or **resetting** the latch

## Clearing the NAND Gate Latch (2)



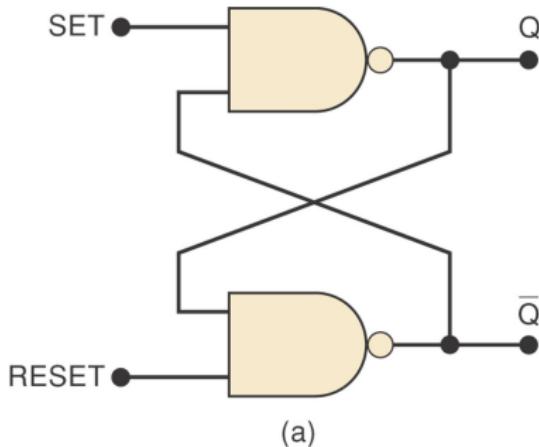
- ▶ Pulsing the RESET input to the LOW state when
  - ▶  $Q = 0$  prior to RESET pulse
  - ▶  $Q = 1$  prior to RESET pulse
- ▶ In each case,  $Q$  results in LOW

## Simultaneous LOW



- ▶ Simultaneously setting LOW on CLEAR and SET leads to  $Q = 1$ , while  $\bar{Q} = 1$
- ▶ Undesired state
- ▶ Output then depends on order of setting CLEAR or SET to 1
- ▶ Not being used for NAND latches

## NAND Gate Latch Summary (1)



Set	Reset	Output
1	1	No change
0	1	$Q = 1$
1	0	$Q = 0$
0	0	Invalid*

\*Produces  $Q = \bar{Q} = 1$ .

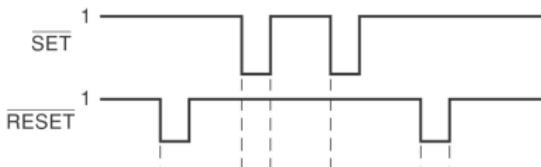
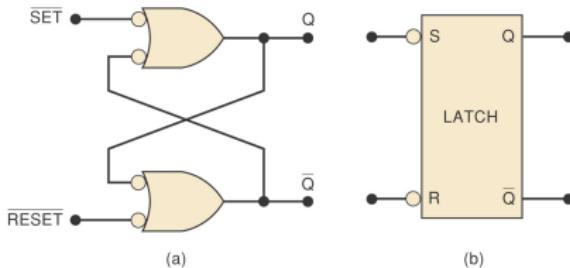
(b)

## NAND Gate Latch Summary (2)

Summary of the NAND latch:

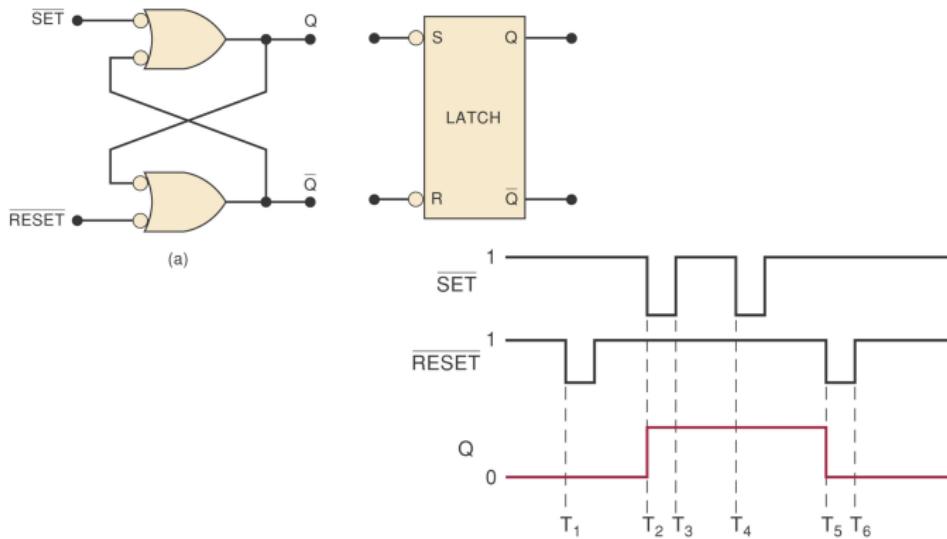
- ▶ SET = RESET = 1, normal resting state, outputs remain in state prior to input
- ▶ SET = 0, RESET = 1,  $Q$  will go high and remain high even if the SET input goes high
- ▶ SET = 1, RESET = 0,  $Q$  will go low and remain low even if the RESET input goes high
- ▶ SET = RESET = 0, output is unpredictable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

## Latch Alternate Representation (1)

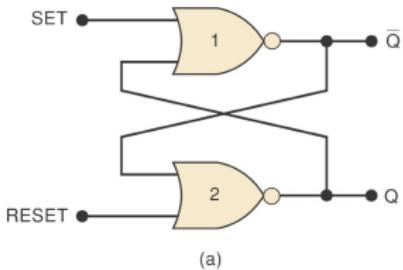


Assume initially  $Q = 0$ , determine the waveform for  $Q$

## Latch Alternate Representation (2)



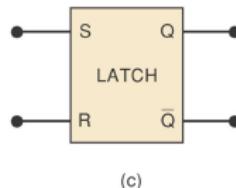
## NOR Gate Latch



Set	Reset	Output
0	0	No change
1	0	$Q = 1$
0	1	$Q = 0$
1	1	Invalid*

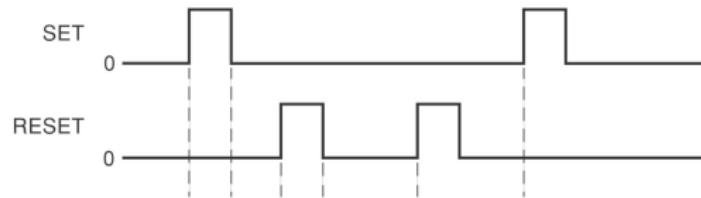
\*Produces  $Q = \bar{Q} = 0$ .

(b)



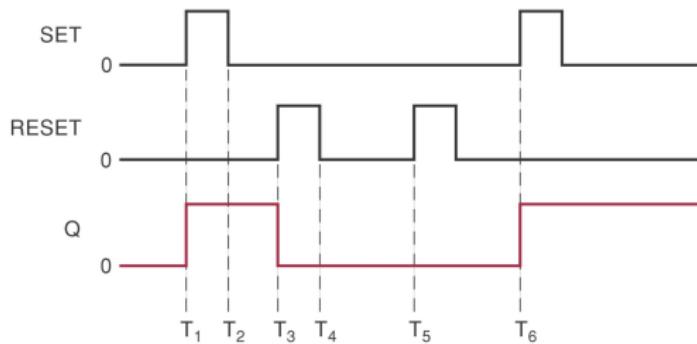
- ▶ The NOR latch is similar to the NAND latch except that the  $Q$  and  $\bar{Q}$  outputs are reversed
- ▶ The SET and RESET inputs are active high, that is, the output will change if input is pulsed high
- ▶ In order to ensure that a FF begins operation at a known level, a pulse may be applied to the SET or RESET inputs when a device is powered up

## Q Waveform for the NOR Latch (1)



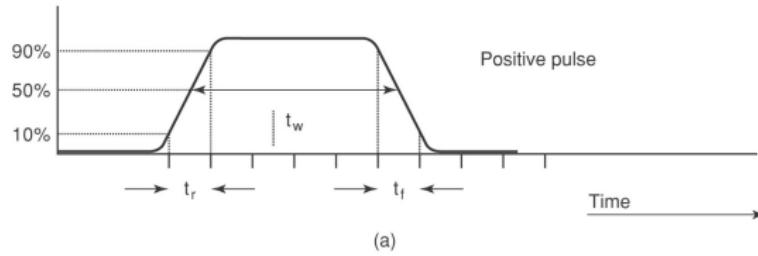
Determine the waveform for  $Q$  ( $Q_0 = 0$ )

## Q Waveform for the NOR Latch (2)



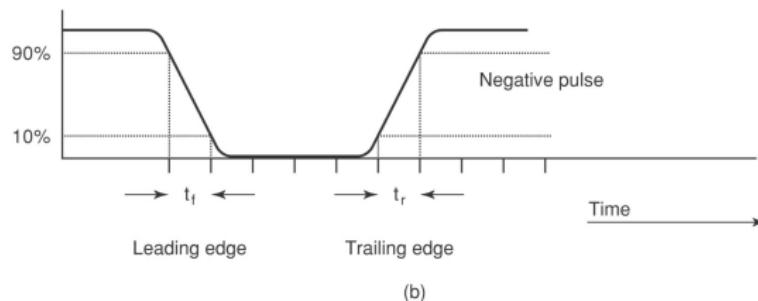
- ▶ CLEAR at  $T_5$  has no effect on  $Q$  since  $Q$  is already LOW
- ▶ SET pulse at  $T_6$  again sets  $Q$  back to 1, where it stays

## Digital Pulses (1)



(a)

- ▶ Signals that switch between active and inactive states are called pulse waveforms



(b)

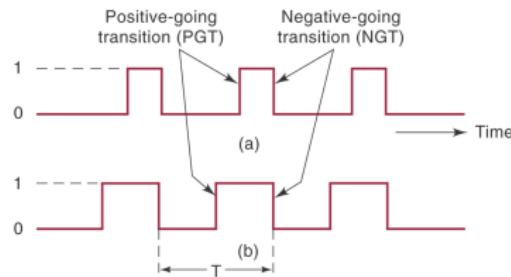
- ▶ A **positive pulse** has an active high level
- ▶ A **negative pulse** has an active low level

## Digital Pulses (2)

- ▶ The transition from low to high on a positive pulse is called rise time ( $t_r$ )
  - ▶ Rise time is measured between the 10% and 90% points on the leading edge of the voltage waveform
- ▶ The transition from high to low on a positive pulse is called fall time ( $t_f$ )
  - ▶ Fall time is measured between the 90% and 10% points on the trailing edge of the voltage waveform

## Clock Signals and Clocked Flip-Flops (1)

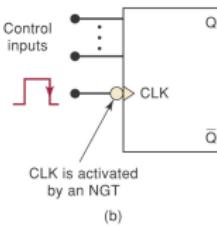
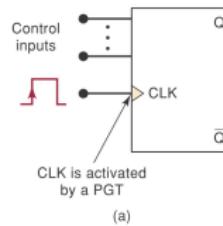
- ▶ **Asynchronous system** – outputs can change state at any time the input(s) change
- ▶ **Synchronous system** – output can change state only at a specific time in the clock cycle
- ▶ The clock signal is a rectangular pulse train or square wave
- ▶ **Positive going transition (PGT)** – when clock pulse goes from 0 to 1
- ▶ **Negative going transition (NGT)** – when clock pulse goes from 1 to 0
- ▶ Transitions are also called **edges**



## Clock Signals and Clocked Flip-Flops (2)

Clocked FFs change state on one or the other clock transitions. Some common characteristics:

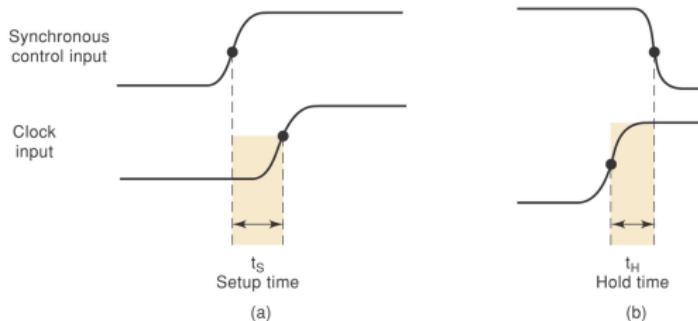
- ▶ Clock inputs are labeled CLK, CK, or CP
- ▶ A small triangle at the CLK input indicates that the input is activated with a PGT
- ▶ A bubble and a triangle indicates that the CLK input is activated with a NGT
- ▶ Control inputs have an effect on the output only at the active clock transition (PGT or NGT)



- ▶ These 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

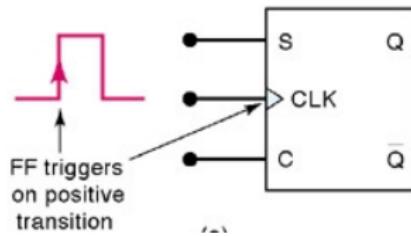
## Clock Signals and Clocked Flip-Flops (3)

- ▶ **Setup time ( $t_S$ )** is the minimum time before the active CLK transition while the control input must be kept at the proper level; ensures that the system moves to next state smoothly
- ▶ **Hold time ( $t_H$ )** is the minimum time following the active CLK transition during which the control input must be kept at the proper level; ensures that the system does not deviate from the current state and go into an invalid state



## Clocked S-R Flip-Flop (1)

The SET-RESET (or SET-CLEAR) FF will change states at the positive going or negative going clock edge

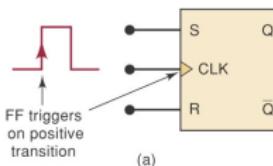


Inputs			Output
S	C	CLK	Q
0	0	↑	$Q_0$ (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

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

## Clocked S-R Flip-Flop (2)

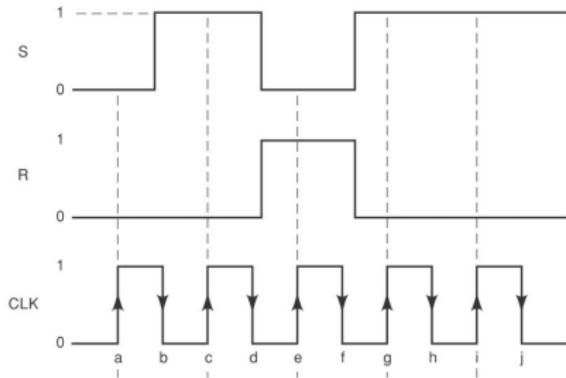
- ▶ Clocked S-R flip-flop that responds only to the positive-going edge of a clock pulse
- ▶ Determine the waveform for  $Q$  ( $Q_0 = 0$ )



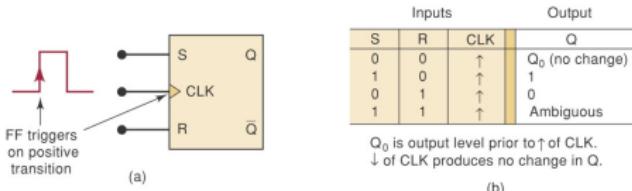
Inputs			Output
S	R	CLK	Q
0	0	↑	$Q_0$ (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

Q<sub>0</sub> is output level prior to ↑ of CLK.  
↓ of CLK produces no change in Q.

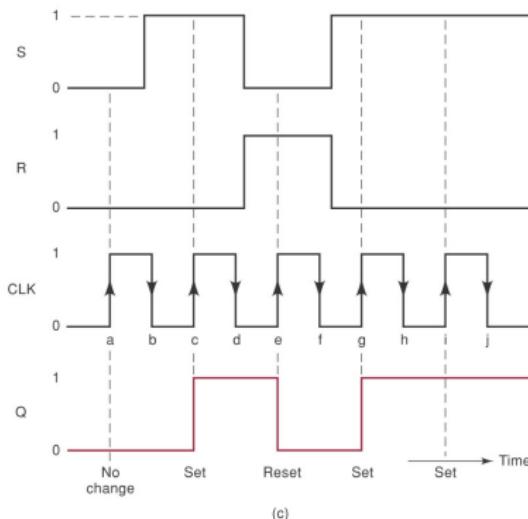
(b)



## Clocked S-R Flip-Flop (3)

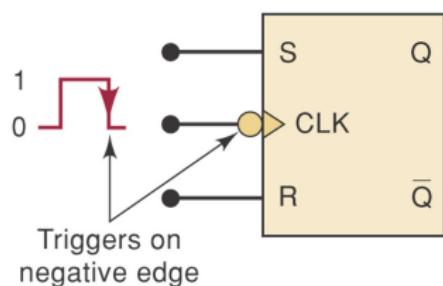


Clocked S-R  
flip-flop that  
responds only to the  
positive-going edge  
of a clock pulse



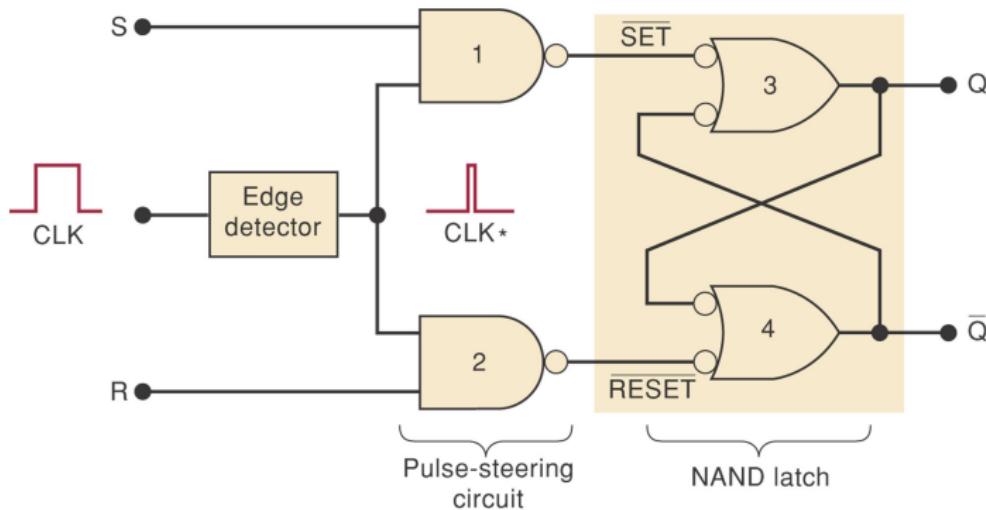
## Clocked S-R Flip-Flop (4)

Clocked S-R flip-flop that responds only to the negative-going edge of a clock pulse

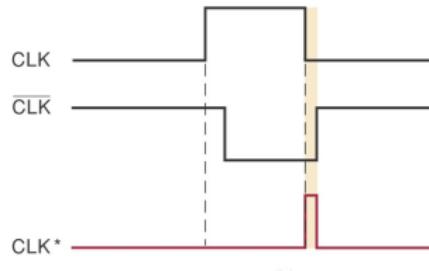
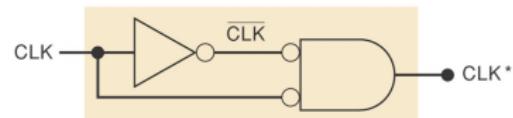
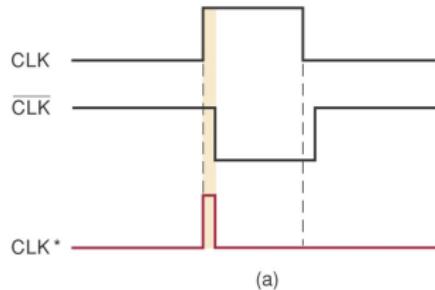
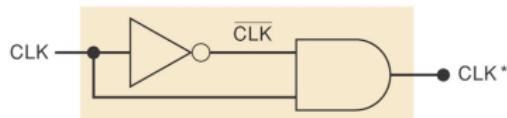


Inputs			Output
S	R	CLK	Q
0	0	↓	$Q_0$ (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambiguous

## Internal Circuitry of Edge-triggered S-R Flip-Flop



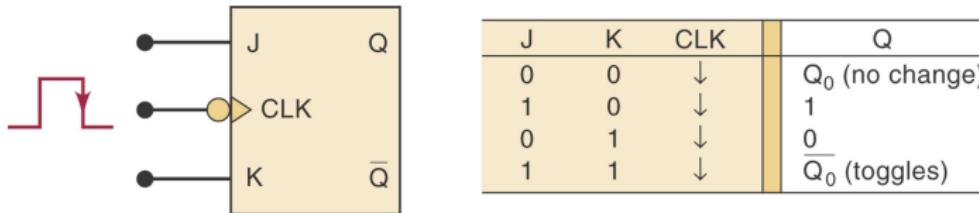
## Implementation of Edge-detector Circuit



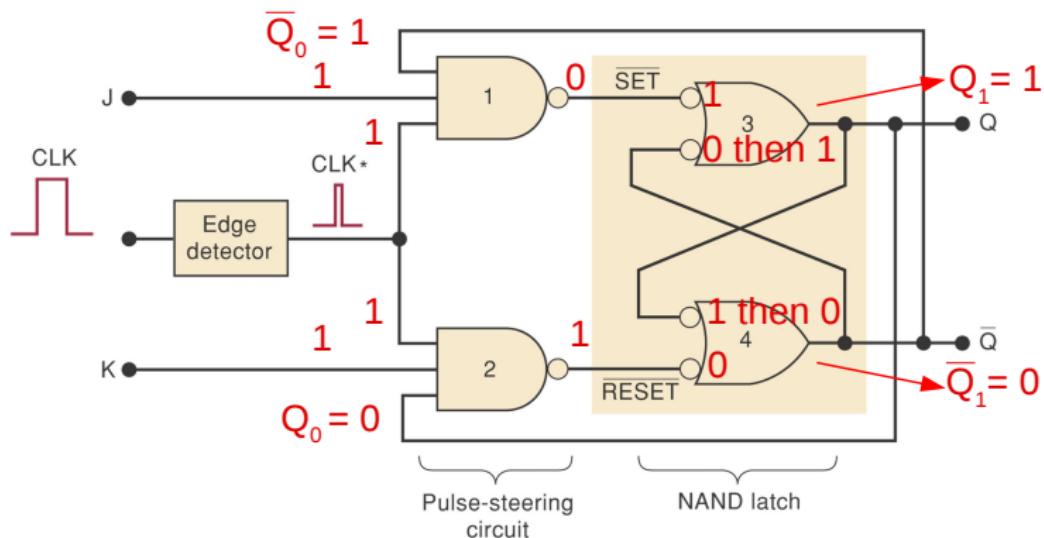
The INVERTER produces a delay of such that the transitions of  $\overline{\text{CLK}}$  occur a little bit after those of CLK

## Clocked J-K Flip-Flop (1)

- ▶ Operates like the S-R FF, J is set, K is clear
- ▶ When J and K are both high the output is  **toggled** from whatever state it is into the opposite state
- ▶ May be positive going or negative going clock trigger
- ▶ Has the ability to do everything the S-R FF does, plus operate in toggle mode



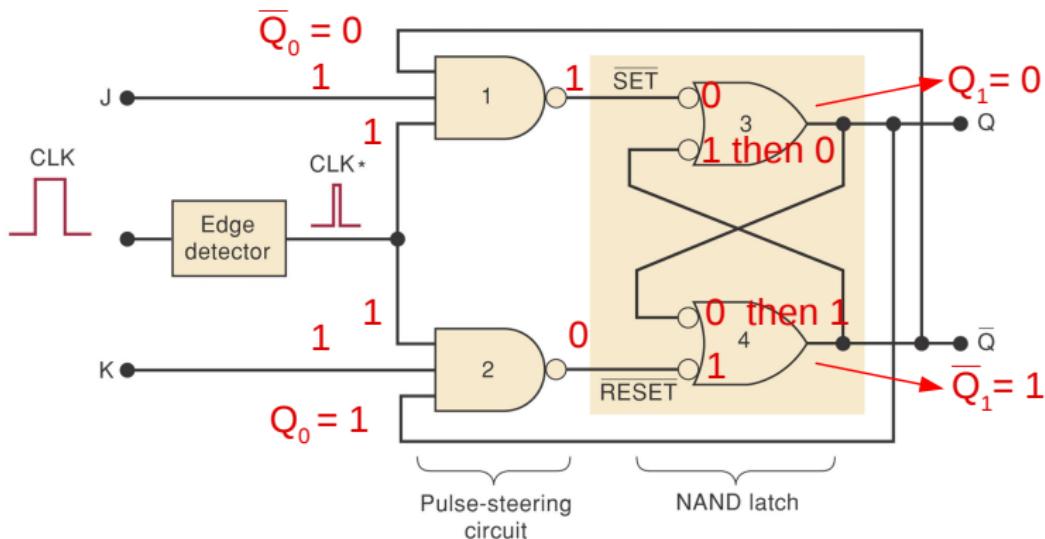
Toggle mode:  $Q_0 = 0, \overline{Q}_0 = 1$



$Q_0$  is the initial value,  $Q_1$  is the resulting value

Output OR gate 4 becomes 0 as soon as  $Q_1 = 1$

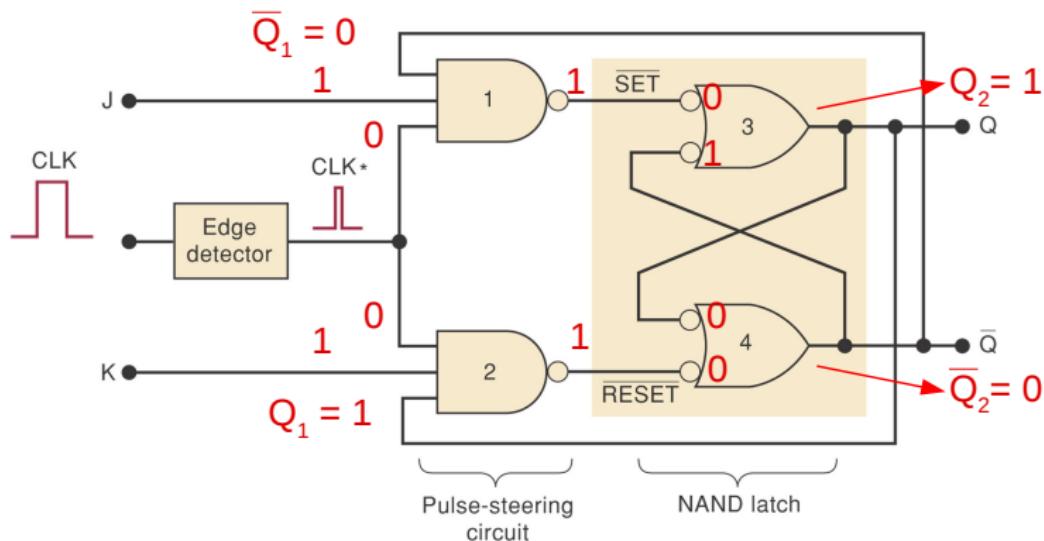
Toggle mode:  $Q_0 = 1, \overline{Q}_0 = 0$



$Q_0$  is the initial value,  $Q_1$  is the resulting value

Output OR gate 3 becomes 0 as soon as  $\overline{Q}_1 = 1$

$Q_1 = 1$  and No Edge

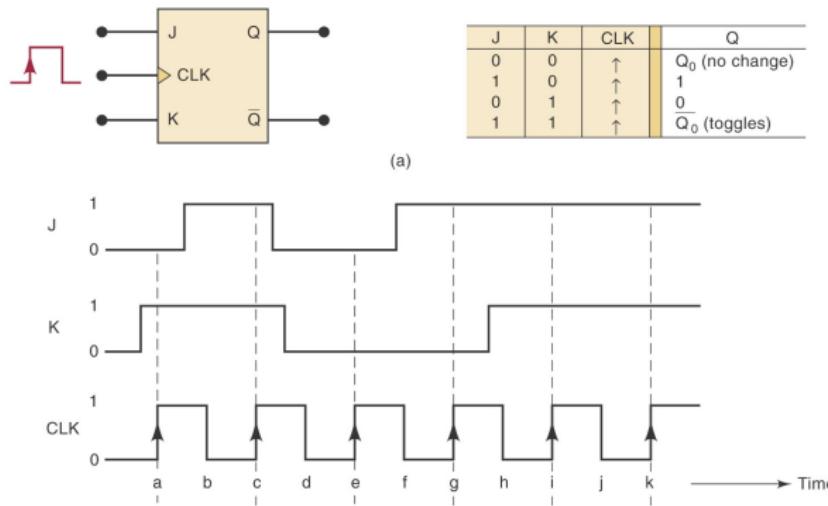


$Q_1$  is now the initial value,  $Q_2$  is the resulting value

## Toggle Mode J-K Flip-Flop

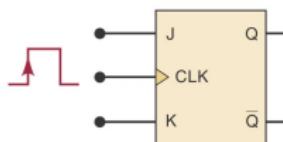
- ▶ Clock spike signal  $\text{CLK}^*$  (the narrow positive signal the edge detector has produced) must be very narrow
- ▶  $\text{CLK}^*$  needs to be already 0 before  $Q$  and  $\bar{Q}$  toggle to new values, otherwise new values of  $Q$  and  $\bar{Q}$  will toggle (if  $\text{CLK}^*$  is still high) outputs again

## Clocked J-K Flip-Flop Example (1)



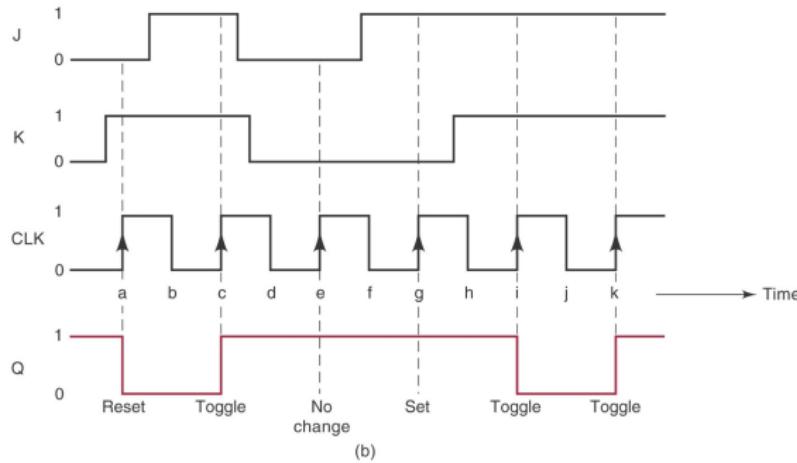
- ▶ Toggle mode
- ▶ Determine the waveform for  $Q$  ( $Q_0 = 1$ )

## Clocked J-K Flip-Flop Example (2)



(a)

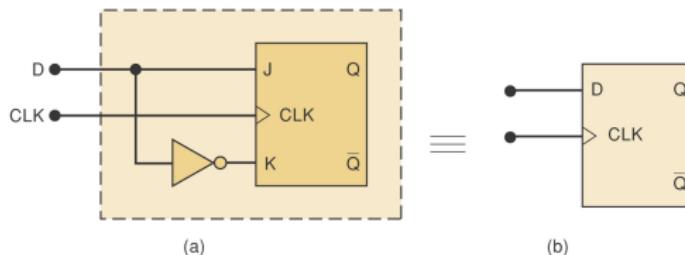
J	K	CLK	Q
0	0	↑	$Q_0$ (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$Q_0$ (toggles)



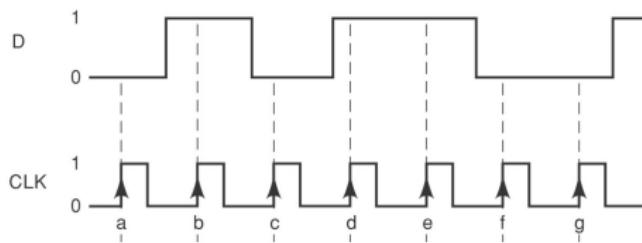
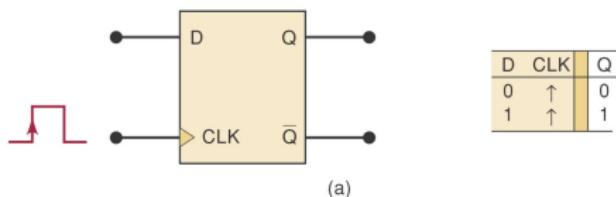
(b)

## Clocked D Flip-Flop (1)

- ▶ One data input
- ▶ The output changes to the value of the input at either the positive going or negative going clock trigger
- ▶ May be implemented with a J-K FF by tying the J input to the K input through an inverter
- ▶ Useful for parallel data transfer

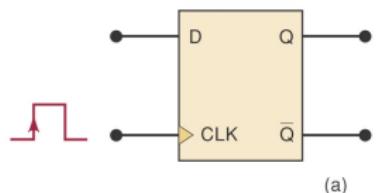


## Clocked D Flip-Flop (2)



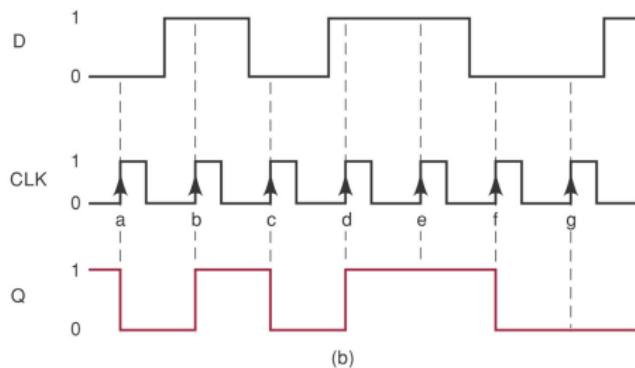
- ▶ The level present at D will be **stored** in the flip-flop when PGT occurs at CLK
- ▶ Determine the waveform for  $Q$  ( $Q_0 = 1$ )

## Clocked D Flip-Flop (3)



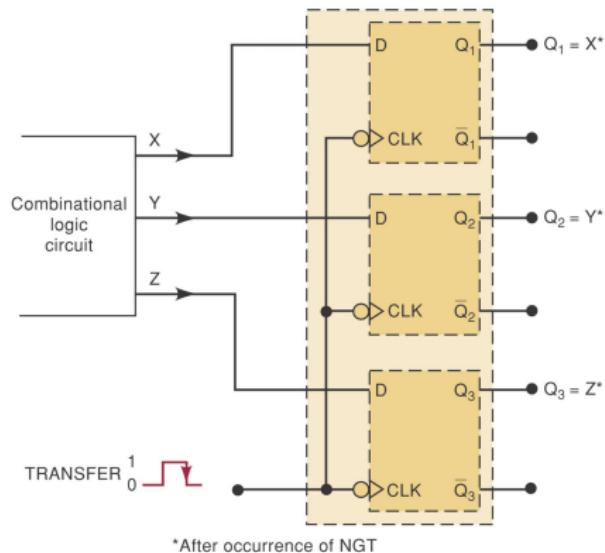
D	CLK	Q
0	↑	0
1	↑	1

(a)



- ▶  $D$  is low  $\rightarrow Q = 0$
- ▶  $Q$  goes high because  $D$  is high at PGT
- ▶  $D$  is low  $\rightarrow Q = 0$
- ▶ ...
- ▶  $Q$  can only change when PGT occurs

## Parallel Data Transfer

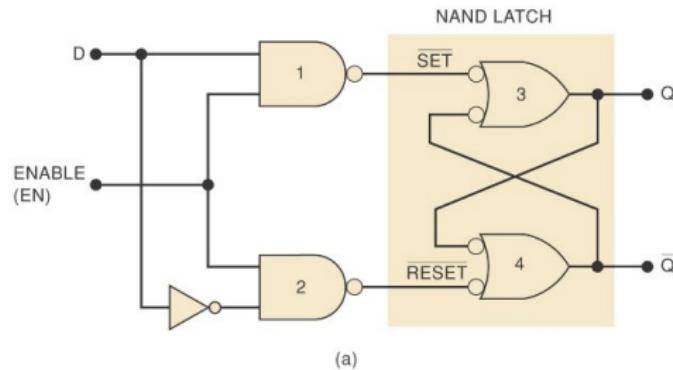


- ▶ Parallel transfer of binary data:
- ▶ Q output takes on value at its input D only at specific times
- ▶ Inputs X, Y, Z are transferred to Q<sub>1</sub>, Q<sub>2</sub>, Q<sub>3</sub> upon TRANSFER pulse at CLK inputs

## D Latch (Transparent Latch)

- ▶ D flip-flop uses edge detector so output only responds to D-input on transition of clock
- ▶ For the D Latch, clock has been replaced by an enable line
- ▶ The device is NOT edge triggered
- ▶ The output follows the input only if EN is high

# Transparent D Latch

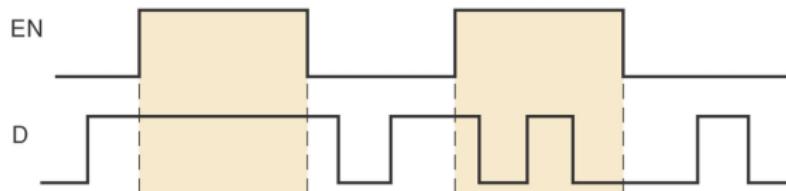


Inputs		Output
EN	D	Q
0	X	$Q_0$ (no change)
1	0	0
1	1	1

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

- ▶ If EN is high Q will be exactly D-input
  - ▶ therefore the D latch is transparent
- ▶ If EN goes LOW D-input is inhibited
  - ▶ output is latched to current level and cannot change while EN is low

## Waveform of Transparent D latch (1)



Determine the waveform for  $Q$  for a D-latch with EN and D inputs ( $Q_0 = 0$ )

## Waveform of Transparent D latch (2)

