Chapter 5

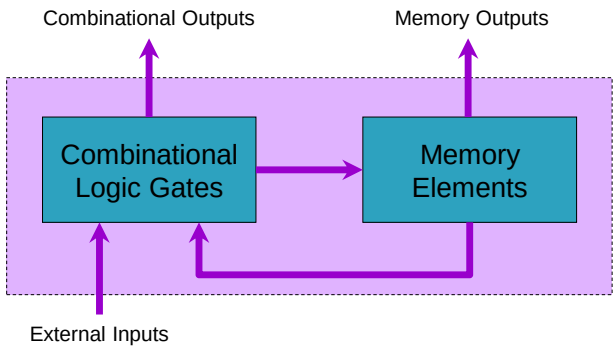
Flip-Flops

Dinh Duc Anh Vu
International University – VNU HCM

Objectives

- Construct & analyze the operation of a latch FF made from NAND or NOR gates
- Describe the difference between synchronous & asynchronous systems
- Understand the operation of edge-triggered FFs
- Understand the major differences between parallel and serial data transfers
- Use state transition diagrams to describe counter operation
- Use FFs in synchronization circuits
- Connect shift registers as data transfer circuits
- Employ FFs as frequency-division and counting circuits

General Digital System Diagram

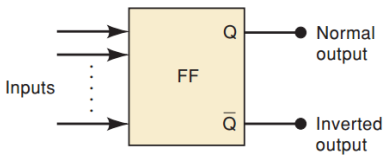


Flip-Flop

Flip-Flop (FF)

Latch

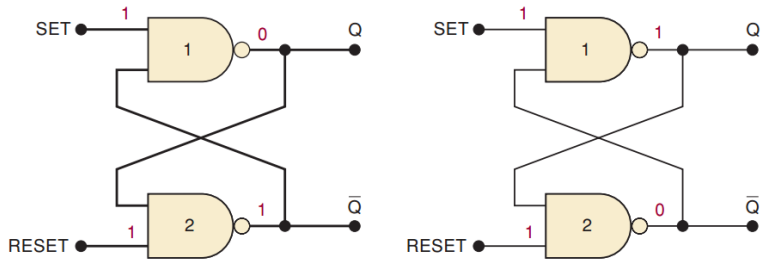
Bistable multivibrator



Output states	
$Q = 1, \overline{Q} = 0$:	called HIGH or 1 state; also called SET state
$Q = 0, \overline{Q} = 1$:	called LOW or 0 state; also called CLEAR or RESET state

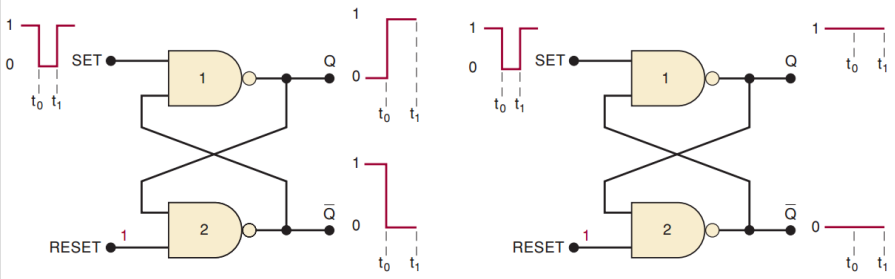
- A FF can have
  - one or more inputs: These inputs are used to cause the FF to switch back and forth ("flip-flop") between its possible output states.
  - two outputs, labeled Q and $\overline{Q}$, that are the inverse of each other
- The state of a FF is referred to the state of its normal output Q
- Most FF inputs need only to be momentarily activated (pulsed) in order to cause a change in the FF output state
- The output will remain in that new state even after the input pulse is over

NAND Gate Latch



- The two NAND gates are cross-coupled so that the output of NAND-1 is connected to one of the inputs of NAND-2, and vice versa.
- There are two latch inputs: the SET input is the input that sets Q to the 1 state; the RESET input is the input that resets Q to the 0 state.
- The SET and RESET inputs are both normally resting in the HIGH state, and one of them will be pulsed LOW whenever we want to change the latch outputs.

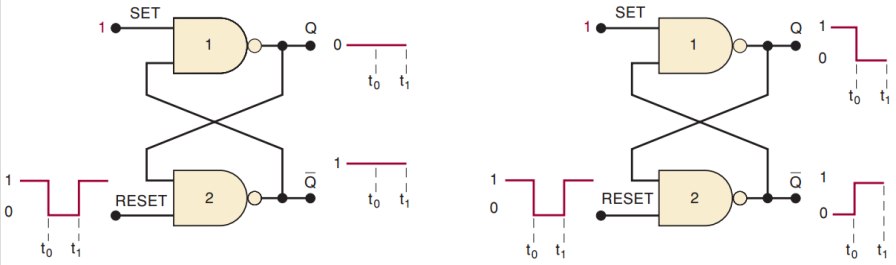
Setting the Latch



Pulsing the SET input to the 0 state while the RESET input is kept 1 when (a) Q = 0 prior to SET pulse; (b) Q = 1 prior to SET pulse.

Note that in both cases Q ends up HIGH

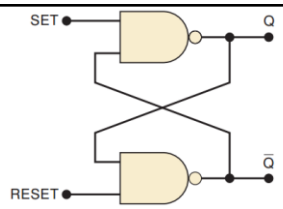
Clearing the Latch



Pulsing the CLEAR input to the LOW state while the SET input is kept 1 when (a) Q =0 prior to CLEAR pulse; (b) Q = prior to CLEAR pulse

In each case, Q ends up LOW

NAND Latch



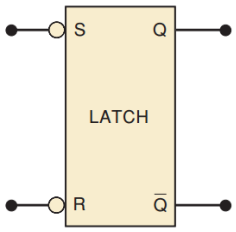
Set	Reset	Output
1	1	No change
0	1	Q = 1
1	0	Q = 0
0	0	Invalid*

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

- SET = RESET = 1
  - This condition is the normal resting state, and it has no effect on the output state. The Q and $\overline{Q}$ outputs will remain in whatever state they were in prior to this input condition
- SET =0; RESET = 1
  - This will always cause the output to go to the state Q=1, where it will remain even after SET returns HIGH. This is called setting the latch.
- SET = 1; RESET = 0
  - This will always produce the state Q=0, where the output will remain even after RESET returns HIGH. This is called clearing or resetting the latch
- SET = RESET = 0
  - This condition tries to set and clear the latch at the same time, and it produces $Q = \overline{Q} = 1$. If the inputs are returned to 1 simultaneously, the resulting state is unpredictable. This input condition should not be used.

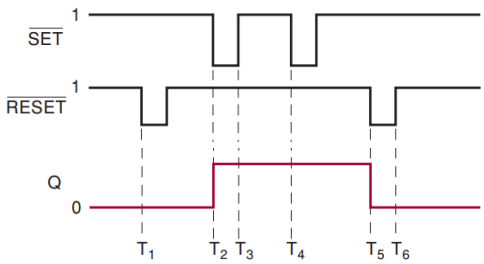
Alternate Representations

- From the description of the NAND latch operation, it should be clear that the SET and RESET inputs are active-LOW.
  - The SET input will set when SET goes LOW;
  - The RESET input will clear when RESET goes LOW.
- The bubbles on the inputs, as well as the labeling of the signals as $\overline{SET}$ and $\overline{RESET}$ indicate the active-LOW status of these inputs



NAND Latch – Example

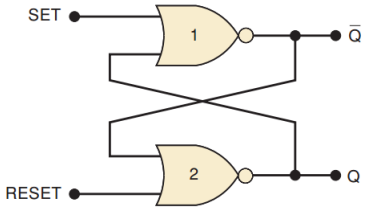
- The waveforms are applied to the inputs of the latch. Assume that initially Q=0, and determine the Q waveform.



Review questions

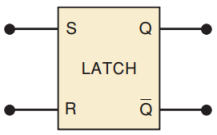
- What is the normal resting state of the SET and RESET inputs? What is the active state of each input?
  - HIGH; LOW
- What will be the states of Q and $\overline{Q}$ after a FF has been reset (cleared)?
  - Q=0
- The SET input can never be used to make Q=0. True or False?
  - True
- When power is first applied to any FF circuit, it is impossible to predict the initial states of Q and $\overline{Q}$. What can be done to ensure that a NAND latch always starts off in the state Q=1?
  - Apply a momentary LOW to SET input

NOR Gate Latch



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

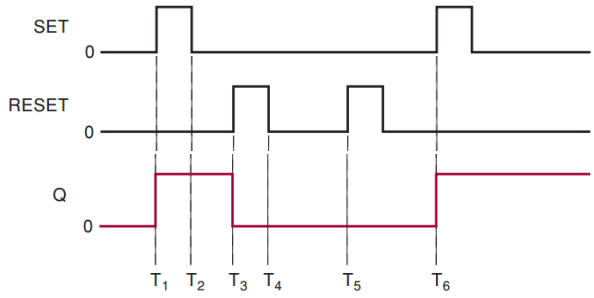
*Produces Q = $\overline{Q}$ = 0.



- Made of two cross-coupled NOR gates
- SET=RESET=0
  - This is the normal resting state for the NOR latch, and it has no effect on the output state. Q and will remain in whatever state they were in prior to the occurrence of this input condition
- SET=1; RESET=0
  - This will always set Q=1, where it will remain even after SET returns to 0
- SET=0; RESET=1
  - This will always clear Q=0, where it will remain even after RESET returns to 0
- SET=RESET=1
  - This condition tries to set and reset the latch at the same time, and it produces $Q = \overline{Q} = 0$. If the inputs are returned to 0 simultaneously, the resulting output state is unpredictable. This input condition should not be used

NOR Latch – Example

- Assume that Q=0 initially, and determine the Q waveform for the NOR latch

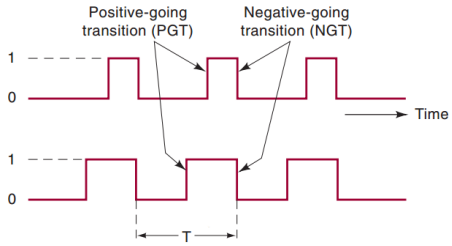


Review questions

- What is the normal resting state of the NOR latch inputs? What is the active state?
  - LOW; HIGH
- When a latch is set, what are the states of Q?
  - Q=1
- What is the only way to cause the Q output of a NOR latch to change from 1 to 0?
  - Make RESET=1

Clock Signals

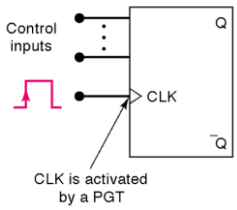
- Digital systems can operate
  - Asynchronously: output can change state whenever inputs change
  - Synchronously: output only change state at clock transitions (edges)

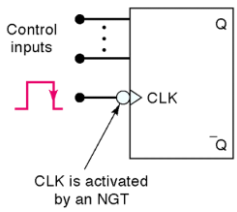


- Clock signal
  - Outputs change state at the transition (edge) of the input clock
  - Positive-going transitions (PGT): clock changes from 0 to 1
  - Negative-going transitions (NGT): clock changes from 1 to 0

Clocked FFs

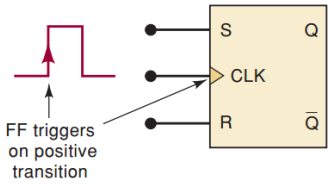
- Clocked FFs have a clock input that is typically labeled CLK, CK, or CP.
  - Normally use CLK
  - In most clocked FFs, the CLK input is edge-triggered, which means that it is activated by a signal transition; this is indicated by the presence of a small triangle on the CLK input.
- Clocked FFs also have one or more control inputs that can have various names, depending on their operation
- The control inputs get the FF outputs ready to change, while the active transition at the CLK input actually triggers the change.
  - The control inputs control the WHAT (i.e., what state the output will go to); the CLK input determines the WHEN.





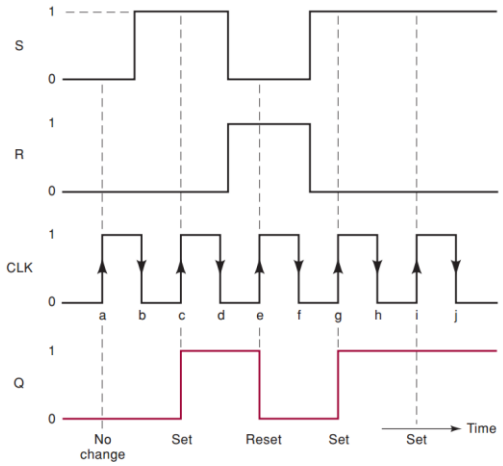
Clocked S–C FF

The *S* and *R* inputs control the state of the FF in the same manner as the NOR gate latch, but the FF does not respond to these inputs until the occurrence of the PGT of the clock signal.



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

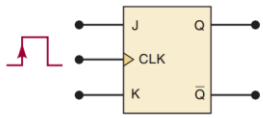
Q_0 is output level prior to ↑ of CLK.
↓ of CLK produces no change in Q.



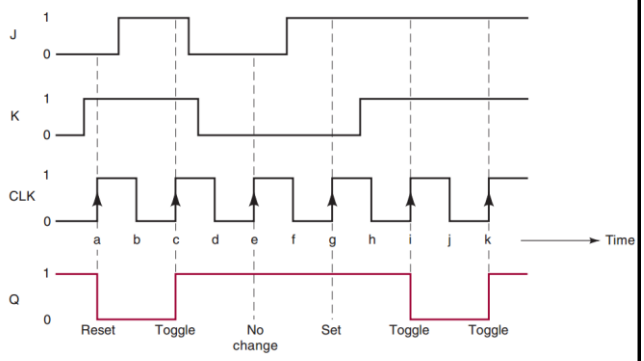
Clocked J–K FF

J=K=1 condition does not result in an ambiguous output as S–C FF.

This condition is toggle mode



J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)

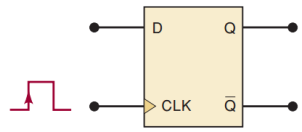


Review questions

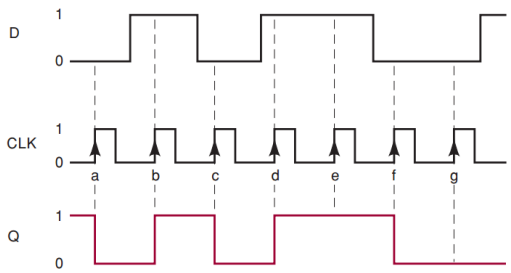
- True or false: A J-K flip-flop can be used as an S-R flip-flop, but an S-R flip-flop cannot be used as a J-K flip-flop.
  - True
- Does a J-K flip-flop have any ambiguous input conditions?
  - No
- What J-K input condition will always set Q upon the occurrence of the active CLK transition?
  - J=1 K=0

Clocked D FF

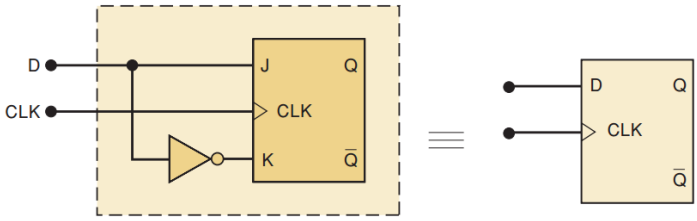
Q will go to the same state that is present on the D input when a PGT occurs at CLK



D	CLK	Q
0	↑	0
1	↑	1



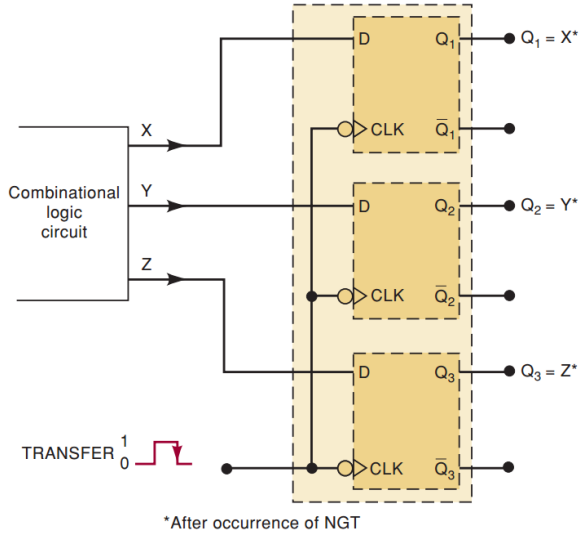
Edge-triggered D FF implementation from a J-K FF



Implement an edge-triggered D FF from a S-C FF?

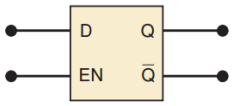
Parallel Data Transfer

Parallel transfer of binary d
using D flip-flops



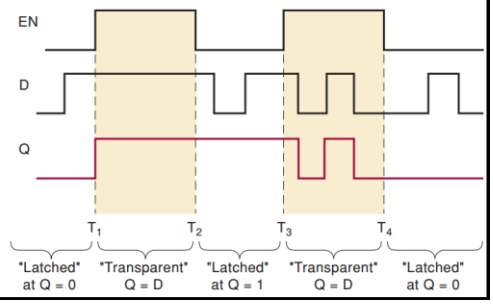
*After occurrence of NGT

D Latch (transparent 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.



Sequential Logic Circuits

Review questions

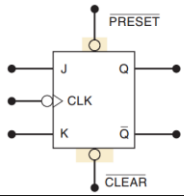
- True or false: A D latch is in its transparent mode when EN=0
  - False
- True or false: In a D latch, the D input can affect Q only when EN=1.
  - True

Sequential Logic Circuits 25

Asynchronous Inputs

- The S, C, J, K, and D inputs is called synchronous inputs because their effects on the output are synchronized with the CLK input
  - the synchronous control inputs must be used in conjunction with a clock signal to trigger the FF
- Asynchronous inputs (override inputs) operate independently of the synchronous inputs and clock and can be used to set the FF to 1/0 states at any time
  - the asynchronous inputs are override inputs, which can be used to override all the other inputs in order to place the FF in one state or the other

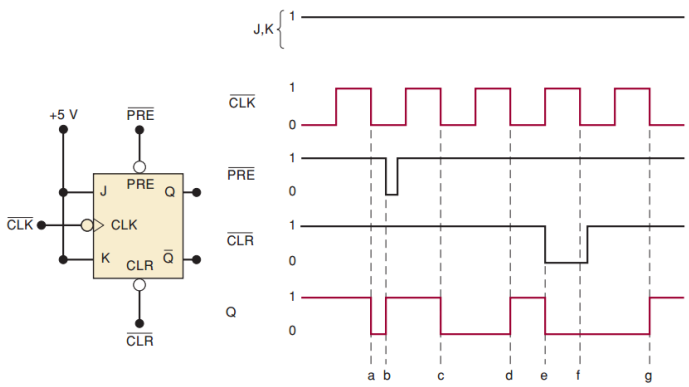
Clocked J-K flip-flop with asynchronous inputs



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	Q̄ (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)

Sequential Logic Circuits

Asynchronous Inputs (cont.)



Point	Operation
a	Synchronous toggle on NGT of $\overline{CLK}$
b	Asynchronous set on $\overline{PRE}$ = 0
c	Synchronous toggle
d	Synchronous toggle
e	Asynchronous clear on $\overline{CLR}$ = 0
f	$\overline{CLR}$ overrides the NGT of $\overline{CLK}$
g	Synchronous toggle

Sequential Logic Circuits 27

Review questions

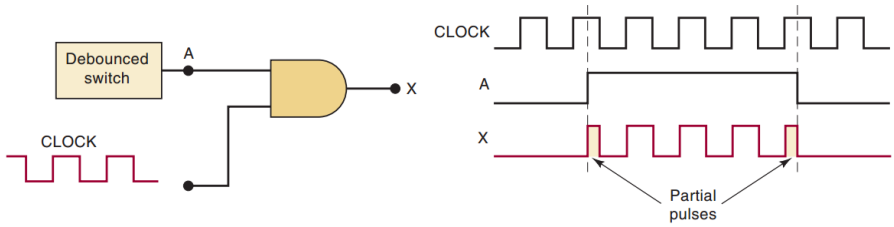
- How does the operation of an asynchronous input differ from that of a synchronous input?
  - Asynchronous inputs work independently of the CLK input.
- Can a D flip-flop respond to its D and CLK inputs while PRE=1?
  - Yes, because PRE is active-LOW
- List the conditions necessary for a positive-edge-triggered J-K flip-flop with active-LOW asynchronous inputs to toggle to its opposite state.
  - J=K=1, PRE=CLR=1 and a PGT at CLK

FF Applications

- Counting, storing of binary data, transferring binary data, and many more …
- Many applications fall into the category of **sequential circuits**, in which the outputs follow a predetermined sequence of states, with a new state occurring each time a clock pulse occurs.

Flip-Flop Synchronization

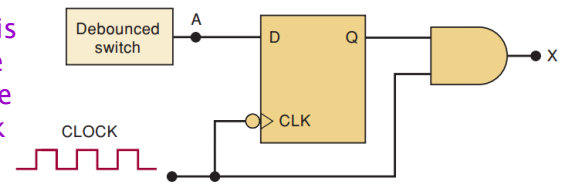
- A FF can be used to synchronize the effect of an asynchronous input whose randomness can produce the unpredictable and undesirable results in digital systems

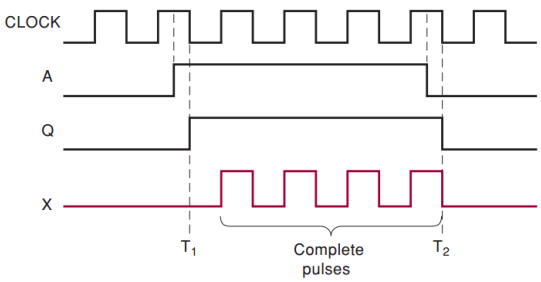


Asynchronous signal A can produce partial pulses at X

FF Synchronization (cont.)

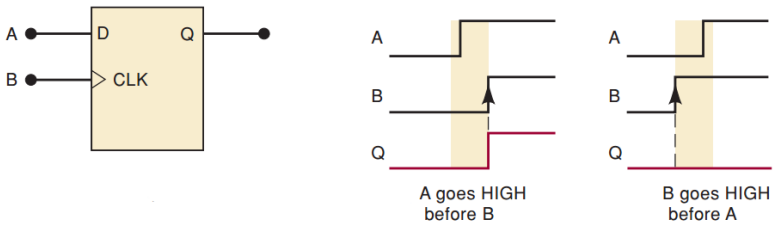
An edge-triggered D FF is used to synchronize the enabling of the AND gate to the NGTs of the clock





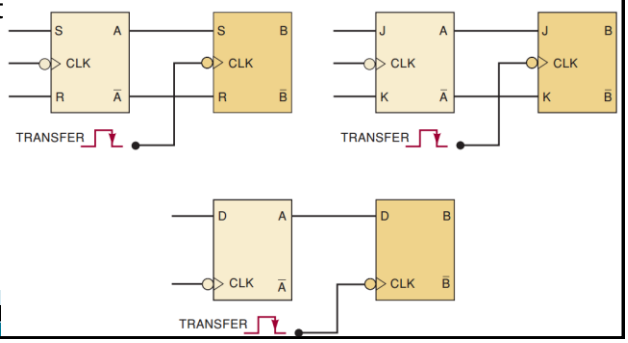
Detecting an Input Sequence

- In many situations an output is to be activated only when the inputs are activated in a certain sequence.
  - This can not be accomplished using pure combinational logic, but FFs can do it
- An AND gate can be used to determine when two inputs A and B are both HIGH, but its output will respond the same regardless of which input goes HIGH first. But suppose that we want to generate a HIGH output only if A goes HIGH and then B goes HIGH some time later.



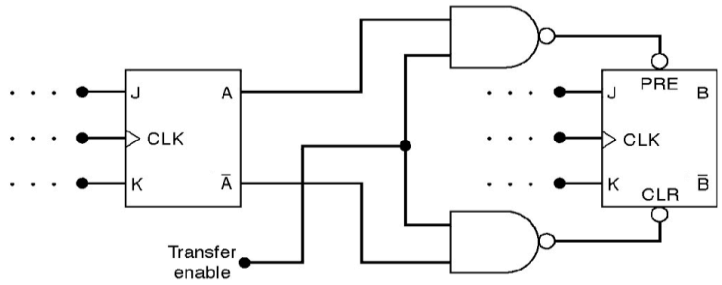
Data Storage and Transfer

- Registers are groups of FFs used to store data.
- Synchronous transfer
  - The logic value that is currently stored in FF A is transferred to FF B upon the NGT of the TRANSFER pulse.
  - Thus, after this NGT, the B output will be the same as the A output



Data Storage and Transfer (cont.)

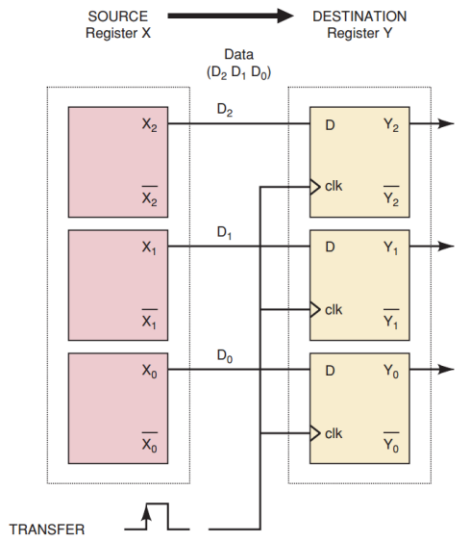
- Asynchronous transfer



Sequential Logic Circuits 34

Parallel Data Transfer

- Data transfer from one register to another using D FFs



Sequential Logic Circuits 35

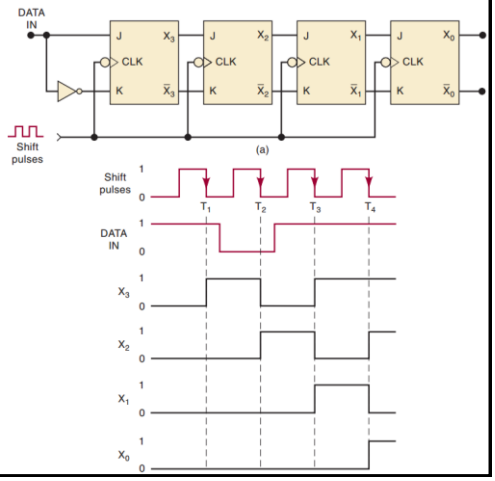
Review questions

- True or false: Asynchronous data transfer uses the CLK input.
  - False
- Which type of FF is best suited for synchronous transfer because it requires the fewest interconnections from one FF to the other?
  - D FF
- If JK flip-flops were used in the registers of the slide 38, how many total interconnections would be required from register X to register Y ?
  - 6
- True or false: Synchronous data transfer requires less circuitry than asynchronous transfer
  - True

Serial Data Transfer: Shift Registers

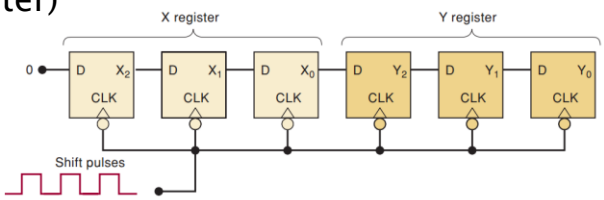
- A shift register is a group of FFs arranged so that the binary numbers stored in FFs are shifted from one FF to the next **for every clock pulse**

The FFs are connected so that the output of X3 transfers into X2, X2 into X1, and X1 into X0. Upon the occurrence of the NGT of a shift pulse, each FF takes on the value stored previously in the FF on its left



Serial Transfer Between Registers

- Serial transfer of information (from X register into Y register)



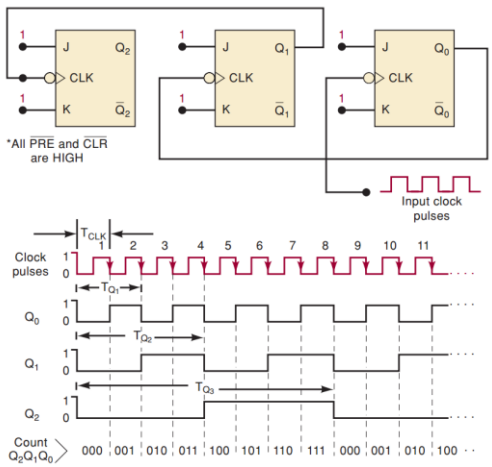
X_2	X_1	X_0	Y_2	Y_1	Y_0	
1	0	1	0	0	0	Before pulses applied
0	1	0	1	0	0	After first pulse
0	0	1	0	1	0	After second pulse
0	0	0	1	0	1	After third pulse

Review questions

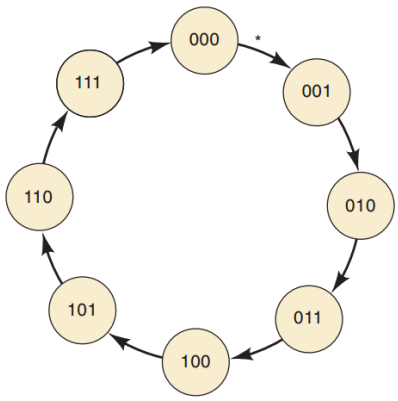
- True or false: The fastest method for transferring data from one register to another is parallel transfer.
  - True
- What is the major advantage of serial transfer over parallel transfer?
  - Fewer interconnections between registers
- Refer to Figure on the slide 41. Assume that the initial contents of the registers are . Also assume that the D input of is held HIGH. Determine the value of each FF output after the occurrence of the fourth shift pulse.
  - X2X1X0 = 111 Y2Y1Y0 = 101
- In which form of data transfer does the source of the data not lose its data?
  - Parallel

Frequency Division And Counting

- JK FFs wired as a 3-bit binary counter (MOD-8)
  - FF Q0 toggles on the NGT of each input clock pulse. Thus, the Q0 output waveform has a frequency that is exactly one half of the clock pulse frequency
  - FF Q1 toggles each time the output goes from HIGH to LOW. The Q1 waveform has a frequency equal to exactly one-half the frequency of the Q0 output and therefore one-fourth of the clock frequency
  - FF Q2 toggles each time the output goes from HIGH to LOW. Thus, the Q2 waveform has one-half the frequency of Q1 and therefore one-eighth of the clock frequency



State Transition Diagram



* **Note:** each arrow represents the occurrence of a clock pulse

State transition diagram shows how the states of the counter FFs change with each applied clock pulse.

MOD Number

- MOD number indicates the number of states in the counting sequence.
  - Counter in slide 40 has 8 different states (000 through 111), so it is Mod-8 counter
  - In general, if N flip-flops are connected in the arrangement of slide 43, the counter will have 2^N different states, and so it is a MOD-2^N counter
- The MOD number of a counter also indicates the frequency division obtained from the last FF.

Review questions

- Consider a counter circuit that contains six FFs wired in the arrangement of slide 43 (i.e. Q5Q4Q3Q2Q1Q0)
  - Determine the counter's MOD number.
    - $2^6 = 64$
  - Determine the frequency at the output of the last FF (Q5) when the input clock frequency is 1 MHz.
    - 1MHz/64 = 15.625KHz
  - What is the range of counting states for this counter?
    - 000000 through 111111 (i.e., 0 to 63)
  - Assume a starting state (count) of 000000. What will be the counter's state after 129 pulses
    - 000001 state
- A 20-kHz clock signal is applied to a JK flip-flop with J=K=1. What is the frequency of the FF output waveform?
  - 20KHz/2 = 10KHz
- How many FFs are required for a counter that will count 0 to 255?
  - $255 = 2^8-1$, so 8 FF are required
- What is the MOD number of the counter in question 3?
  - MOD-256
- What is the frequency of the output of the eighth FF when the input clock frequency is 512 kHz?
  - 512KHz/(2^8) = 2KHz

Summary

1. With a memory characteristics, a flip-flop's outputs will go to a new state in response to an input pulse and will remain in that new state after the input pulse is terminated.
2. A NAND latch and a NOR latch are simple FFs that respond to the logic levels on their SET and CLEAR inputs.
3. Clearing (resetting) a FF means that its output ends up in the Q=0/Q'=1 state. Setting a FF means that it ends up in the Q=1/Q'=0 state.
4. Clocked FFs are edge-triggered, meaning that it triggers the FF on a positive-going transition (PGT) or a negative-going transition (NGT)

Summary (cont.)

5. Edge-triggered (clocked) FFs can be triggered to a new state by the active edge of the clock input according to the state of the FF's synchronous control inputs (S, C or J, K or D)
6. Most clocked FFs have asynchronous inputs that can set or clear the FF independently of the clock input.
7. D latch is a modified NAND latch that operates like a D FF except that it is not edge-triggered.
8. Some of FF applications include data storage and transfer, data shifting ,counting, and frequency division.