Chapter 6

Counters and Registers

Dinh Duc Anh Vu
International University – VNU HCM

Objectives

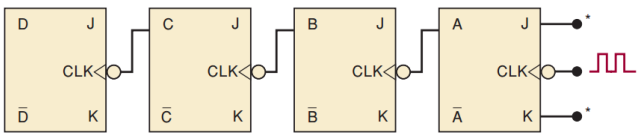
- Understand the operation and characteristics of synchronous and asynchronous counters.
- Construct counters with MOD numbers less than 2^N.
- Construct both up and down counters.
- Connect multistage counters.
- Analyze and evaluate various types of counters.
- Design arbitrary-sequence synchronous counters.
- Understand several types of schemes used to decode different types of counters.
- Compare the major differences between ring and Johnson counters.
- Recognize and understand the operation of various types of IC registers.

Definition for Counter

- Counter is a sequential logic circuit with:
  - The input is Clock used to count.
  - The output displays the numbers in the specified order.
  - After K Clock signals, the counter returns the beginning state.
- The counter is designed by FFs and combinational logic circuit.
  - Serial counter (asynchronous counter)
  - Parallel counter (synchronous counter)

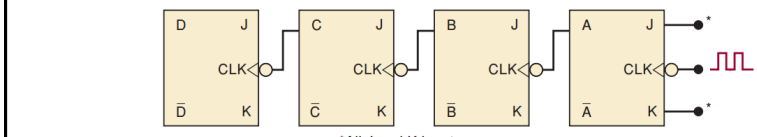
Asynchronous Counter

- An asynchronous (ripple) counter
  - is a counter in which all FFs are controlled by **different** synchronous signal
  - The output of the previous FF is the synchronous signals for the next FF
  - The unstable intermediate state can be appeared when states change
- Principles:
  - N JK FFs ⇒ design an up-counter MOD 2^N
  - Clock signal is excited at pin "CLK" of the first FF.
  - The output of the previous FF is the clock signal at pin "CLK" of the next FF. Continue to the last FF.
  - Pins "J" and "K" of all FFs are set at logic 1.
  - Pins "Clear" and "Preset" are not used (they are always not active).
    - Don't mention them in the circuit diagram to make the circuit clear.

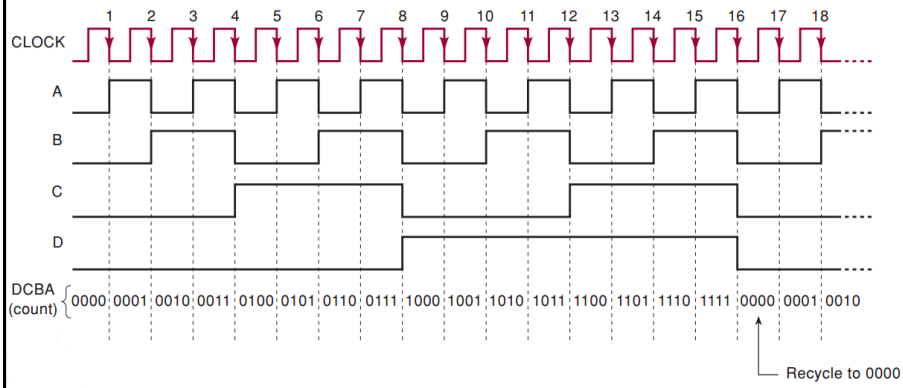


*All J and K inputs assumed to be 1.

MOD-16 Asynchronous Counter

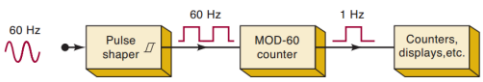


*All J and K inputs assumed to be 1.



Review questions

- The counter in the previous slide starts off in the 0000 state, and then clock pulses are applied. Some time later the clock pulses are removed, and the counter FFs read 0011. How many clock pulses have occurred?
  - 16n+3 (n: integer)
- A counter must be able to count as many as one thousand items. How many FFs are required?
  - to determine what value of N is needed so that $2^N >= 1000$, so 10 FFs are used

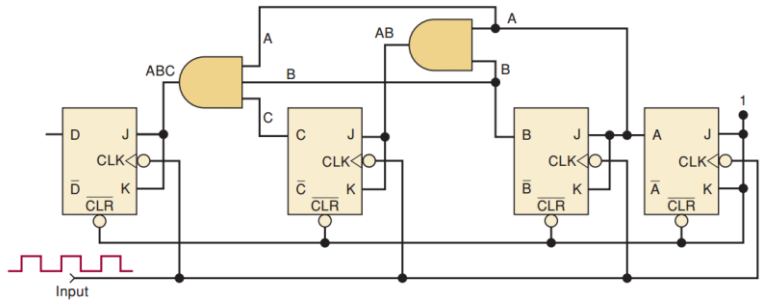


- Clock signal for digital clock: The 60-Hz square wave is put into a MOD-60 counter, which is used to divide the 60Hz frequency by exactly 60 to produce a 1Hz waveform. This 1Hz waveform is fed to a series of counters, which then count seconds, minutes, hours, and so on. How many FFs are required for the MOD-60 counter?
  - 6

Synchronous counter

- A synchronous (parallel) counter:
  - is a counter in which all FFs are controlled by **only one** synchronous signal, the input is clock.
  - The output's state changes only if the synchronous signal is excited.
  - The interval between two synchronous signals is set so that the output's state is stable before changing.
- Principle:
  - To design a synchronous counter, pins "Clear" and "Preset" are not used (they are always not active).
    - Don't mention them in the circuit diagram to make the circuit clear.
  - In this counter, the output Q's changes depend on the inputs J, K or D.

MOD-16 Synchronous Counter



- The CLK inputs of all of the FFs are connected together so that the input clock signal is applied to each FF simultaneously.
- Only flip-flop A, the LSB, has its J and K inputs permanently at the HIGH level. The J, K inputs of the other FFs are driven by some combination of FF outputs.
- The synchronous counter requires more circuitry than does the asynchronous counter.

MOD-16 Synchronous Counter

- Each FF should have its J and K inputs connected so that they are HIGH only when the outputs of all lower-order FFs are in the HIGH state
- The counting sequence shows that **the A flip-flop must change states at each NGT**. For this reason, its J and K inputs are permanently HIGH so that it will toggle on each NGT of the clock input.
- The counting sequence shows that **flip-flop B must change states on each NGT that occurs while A =1**. For example, when the count is 0001, the next NGT must toggle B to the 1 state. This operation is accomplished by connecting output A to the J and K inputs of flip-flop B so that J=K=1 only when A=1.
- The counting sequence shows that **flip-flop C must change states on each NGT that occurs while A=B=1**. For example, when the count is 0011, the next NGT must toggle C to the 1 state. By connecting the logic signal AB to FF C's J and K inputs, this FF will toggle only when A=B=1.
- In a like manner, we can see that **flip-flop D must toggle on each NGT that occurs while A=B=C=1**. When the count is 0111, the next NGT must toggle D to the 1 state. By connecting the logic signal ABC to FF D's J and K inputs, this FF will toggle only when A=B=C=1.

Count	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0
.	.	.	.	.
.	.	.	.	.
.	.	etc.	.	.

Counters and Registers 9

Synchronous Counter ICs

- Actual ICs
  - 74ALS160/162, 74HC160/162: Synchronous decade counters
  - 74ALS161/163, 74HC161/163: Synchronous MOD-16 counters

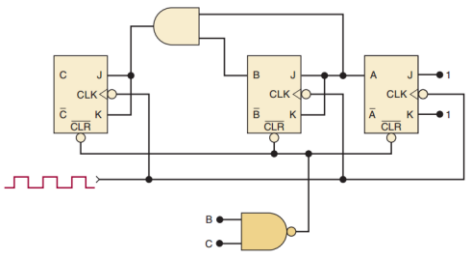
Counters and Registers 10

Review Questions

- What is the advantage of a synchronous counter over an asynchronous counter? What is the disadvantage?
  - can operate at higher clock frequencies and has more complex circuitry
- How many logic devices are required for a MOD-64 parallel counter?
  - 6 FFs and 4 AND gates
- What logic signal drives the J,K inputs of the MSB flip-flop for the counter of the above question?
  - ABCDE

Counters with MOD-K, K< 2^N

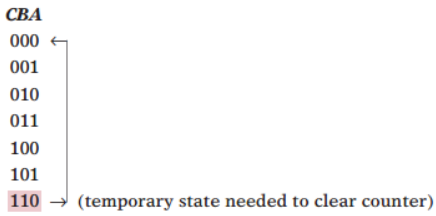
- Design an MOD-K up-counter, K<2^N is the same way that in MOD 2^N in which pins "CLR" are used to reset the counter to the beginning state.
  - E.g. MOD-6 counter produced by clearing a MOD-8 counter when a count of 6 occurs.

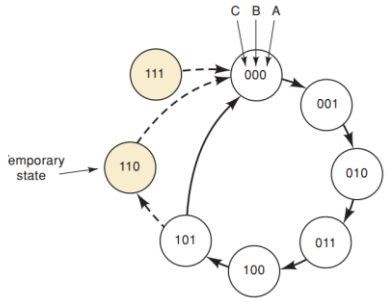


CBA	
000	←
001	
010	
011	
100	
101	
110	→ (temporary state needed to clear counter)

State Transition Diagram

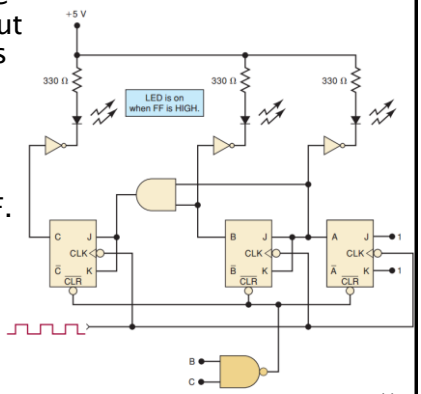
- If we assume a starting count of 000, the diagram shows that the states of the counter change normally up until the count of 101.
- When the next clock pulse occurs, the counter temporarily goes to the 110 count before going to the stable 000 count. The dotted lines indicate the temporary nature of the 110 state. As stated earlier, the duration of this temporary state is so short that for most purposes we can consider that the counter goes directly from 101 to 000 (solid arrow).





Displaying Counter States

- Each FF output is connected to an INVERTER whose output provides the current path for the LED.
  - For example, when output C is HIGH, the INVERTER output goes LOW and the LED turns ON. An LED that is turned on indicates C=1.
  - When output C is LOW, the INVERTER output is HIGH and the LED turns OFF. When the LED is turned off, it indicates C=0



Review questions

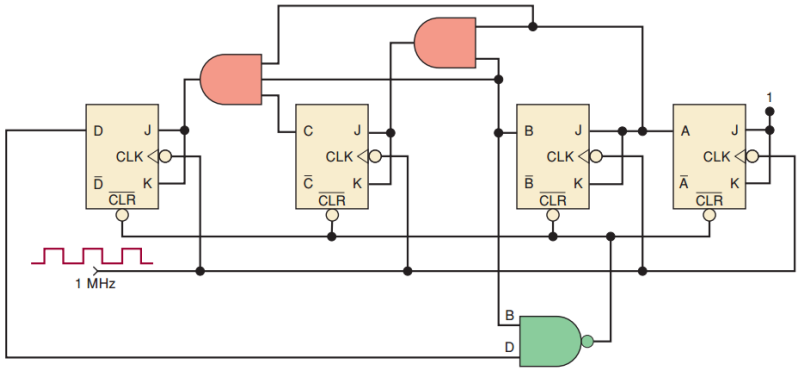
- What will be the status of the LEDs when the counter is holding the count of five?
  - 101_2 : ON, OFF, ON
- What will the LEDs display as the counter is clocked by a 1KHz input?
  - At 1 KHz, the LEDs will be switching ON and OFF so rapidly that they will appear to the human eye to be ON all the time at about half the normal brightness.
- Will the 110 state be visible on the LEDs?
  - No.

General Procedure to design MOD-K Synchronous Counter

- To construct a counter that starts counting from all 0s and has a MOD number of K:
  1. Find the smallest number of FFs such that $2^N \geq K$ and connect them as a counter. If $2^N = K$, donot do steps 2 and 3
  2. Connect a NAND gate to the asynchronous CLEAR inputs of all the FFs
  3. Determine which FFs will be in the HIGH state at a count = K; then connect the normal outputs of these FFs to the NAND gate inputs

Decade/BCD counter

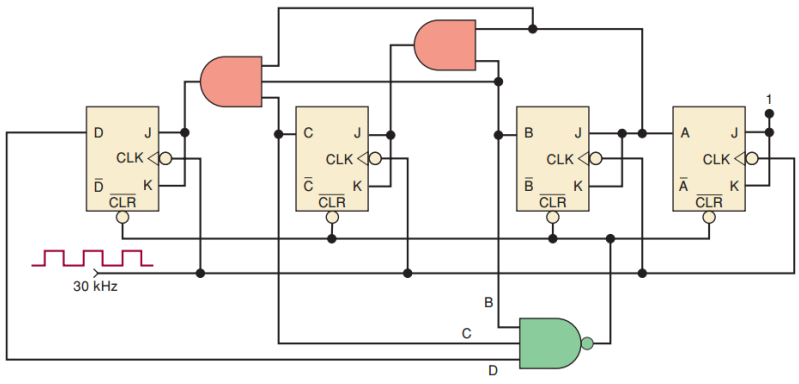
- Construct MOD-10 counter that will count from 0000 through 1001



Counters and Registers 17

Review questions

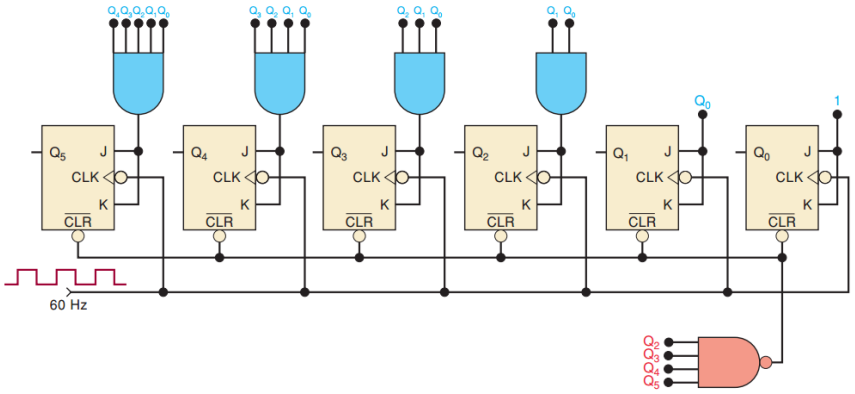
- Determine the MOD number of this counter



Counters and Registers 18

Review questions

- Construct a MOD-60 counter



Counters and Registers 19

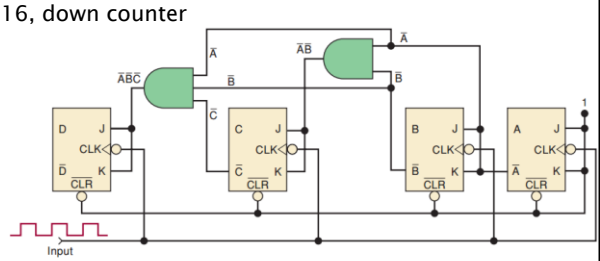
Review Questions

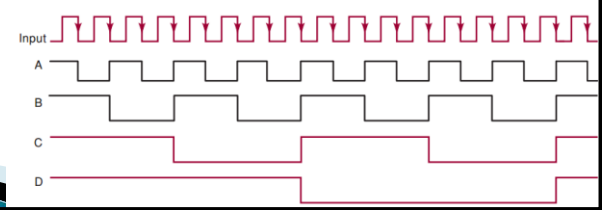
- What FF outputs should be connected to the clearing NAND gate to form a MOD-13 counter?
  - D, C, and A
- True of False: All BCD counters are decade counters.
  - True
- What is the output frequency of a decade counter that is clocked from a 50-KHz signal?
  - 5 KHz

Counters and Registers 20

Synchronous Down Counter

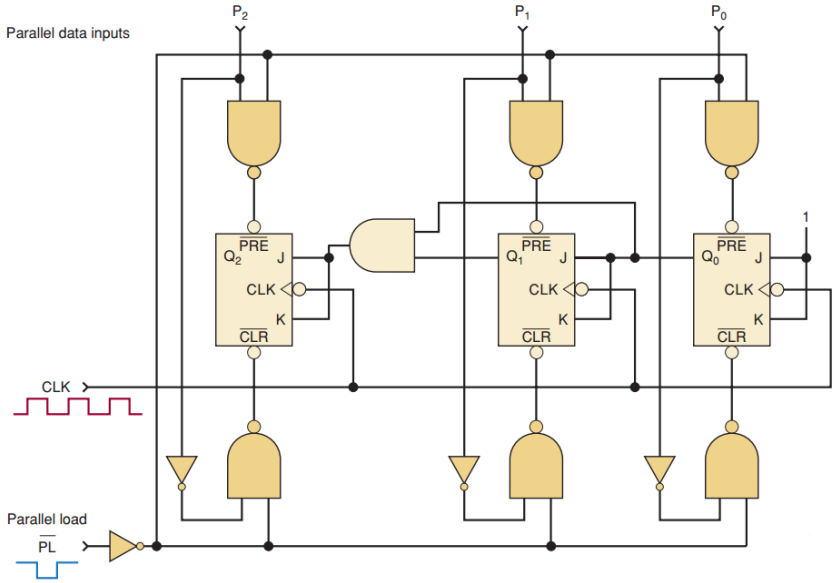
- A synchronous down counter is constructed in a similar manner except that we use the inverted FF outputs to control the higher-order J, K inputs.
- Synchronous, MOD-16, down counter





Counters and Registers

Presettable Counters



Synchronous Presetting

- Counter is preset on the active transition of the same clock signal that is used for counting
- Examples of IC counters that use synchronous presetting
  - 74ALS160, 74ALS161, 74ALS162, 74ALS163
  - 74HC160, 74HC161, 74HC162, 74HC163
- Examples of IC counters that use asynchronous presetting
  - 74ALS190, 74ALS191, 74ALS192, 74ALS193
  - 74HC190, 74HC191, 74HC192, 74HC193

Review Questions

- What is meant when we say that a counter is presettable?
  - It can be preset to any desired starting count.
- Describe the difference between asynchronous and synchronous presetting.
  - Asynchronous presetting is independent of the clock input, while synchronous presetting occurs on the active edge of the clock signal.

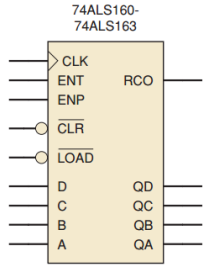
Synchronous/Asynchronous presetting IC counters

- IC counters that use synchronous presetting
  - 74ALS160, 74ALS161, 74ALS162, 74ALS163
  - 74HC160, 74HC161, 74HC162, 74HC163
- IC counters that use asynchronous presetting
  - 74ALS190, 74ALS191, 74ALS192, 74ALS193
  - 74HC190, 74HC191, 74HC192, 74HC193

74xx series: Standard TTL (Transistor-Transistor Logic)
ALS: Advanced Low-Power Schottky technology
HC: High-speed CMOS technology

74ALS160–163 Series

- 74ALS160 and 74ALS162: MOD–10 counters
- 74ALS161 and 74ALS163: MOD–16 counters
- 74ALS160 and 74ALS161 has an asynchronous clear input
- 74ALS162 and 74ALS163: IC counters are synchronously cleared
- The clear input has priority over all other functions
- The second priority function available in this IC series is the parallel loading of data into the counter's flip–flops
- RCO output is used in connecting two or more counter chips together in a multistage arrangement to create larger counters



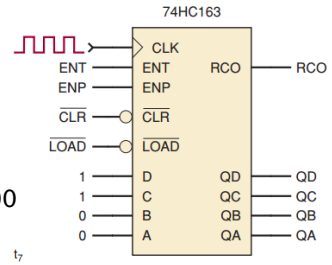
Part Number	Modulus
74ALS160	10
74ALS161	16
74ALS162	10
74ALS163	16

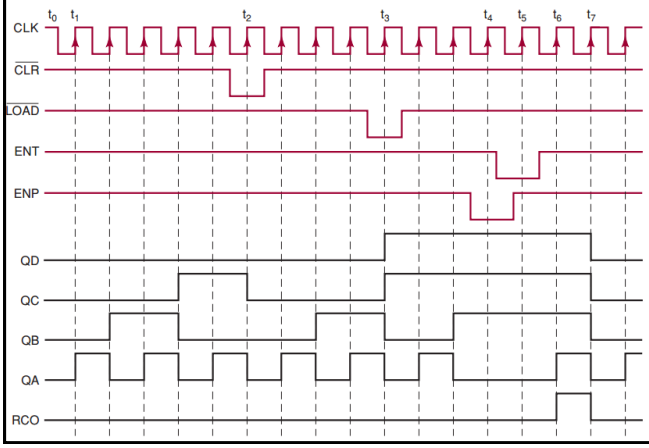
74ALS160-74ALS163 Function Table

$\overline{CLR}$	$\overline{LOAD}$	ENP	ENT	CLK	Function	Part Numbers
L	X	X	X	X	Asynch. Clear	74ALS160 & 74ALS161
L	X	X	X	↑	Synchr. Clear	74ALS162 & 74ALS163
H	L	X	X	↑	Synchr. Load	All
H	H	H	H	↑	Count up	All
H	H	L	X	X	No change	All
H	H	X	L	X	No change	All

Usage: 74HC163

- The parallel data inputs are permanently connected as 1100.
- Assume the counter is initially in the 0000 state

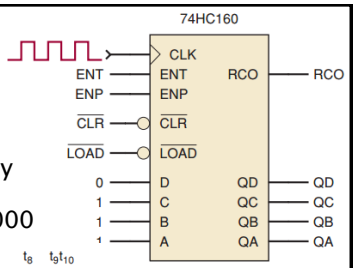


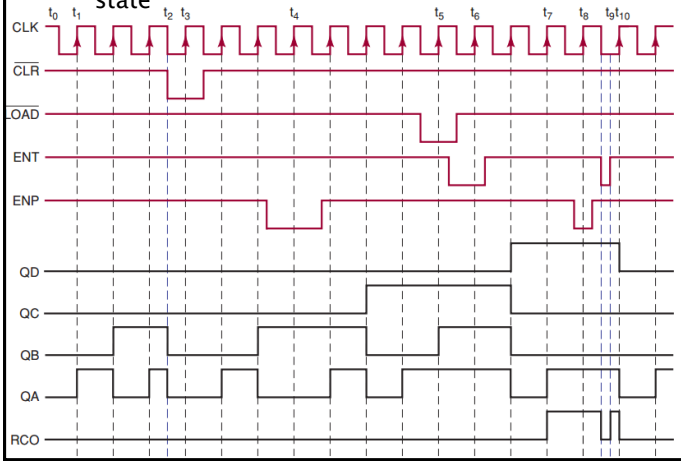


27

Usage: 74HC160

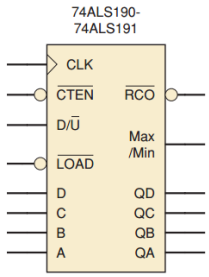
- The parallel data inputs are permanently connected as 0111.
- Assume the counter is initially in the 0000 state





28

74ALS190-191 Series



Part Number	Modulus
74ALS190	10
74ALS191	16

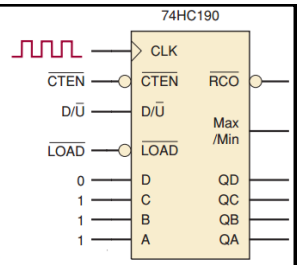
74ALS190-74ALS191 Function Table

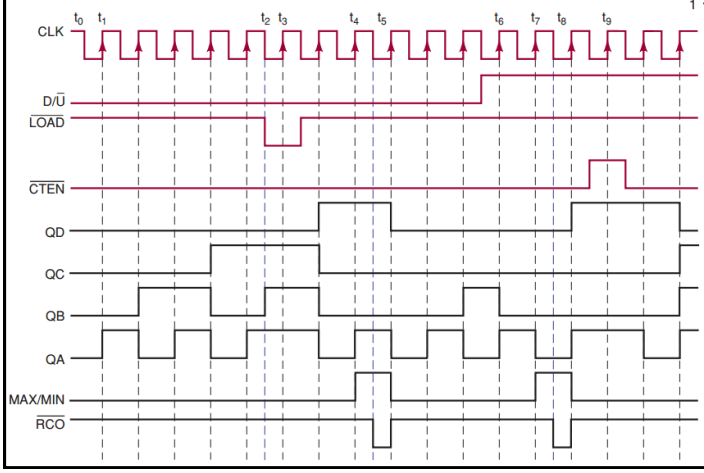
LOAD	CTEN	D/U	CLK	Function
L	X	X	X	Asynch. Load
H	L	L	↑	Count up
H	L	H	↑	Count down
H	H	X	X	No change

- Both chips are up/down counters and have an asynchronous, active-LOW load input
- 74ALS190: MOD-10 counter
- 74ALS191: MOD-16 counter
- The LOAD input has priority over counting functions. As soon as LOAD goes LOW, the counter will be preset to the parallel data on the D, C, B, A (A is LSB and D is MSB) input pins
- RCO output is used in connecting two or more counter chips together in a multistage arrangement to create larger counters

Counters and Registers 29

Usage: 74HC190

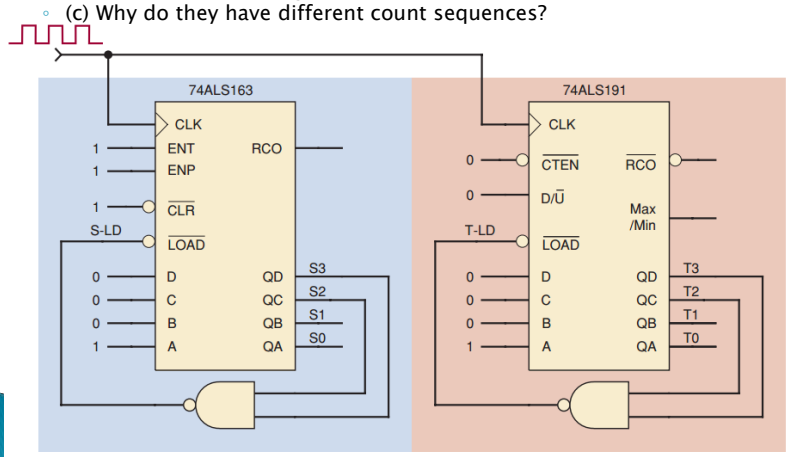




30

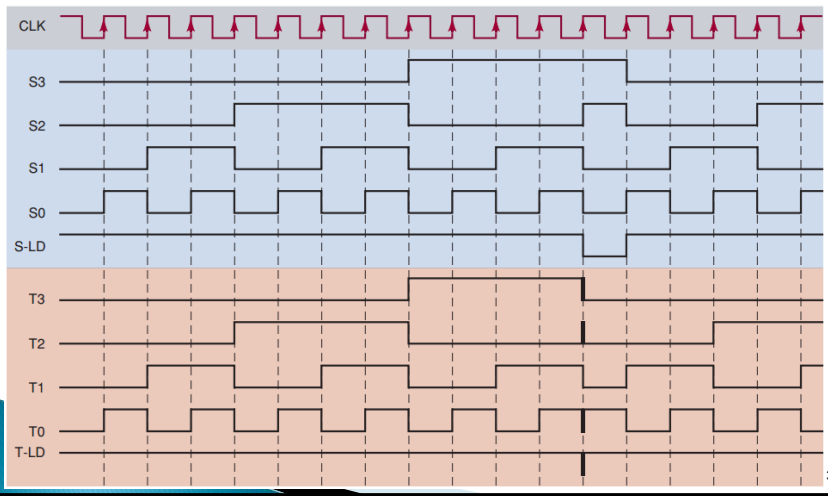
74HC163 vs 74HC191

- Compare the operation of two counters, one with synchronous load and the other with asynchronous load.
  - (a) Determine the output waveform for each counter.
  - (b) What is the recycling count sequence and modulus for each counter?
  - (c) Why do they have different count sequences?



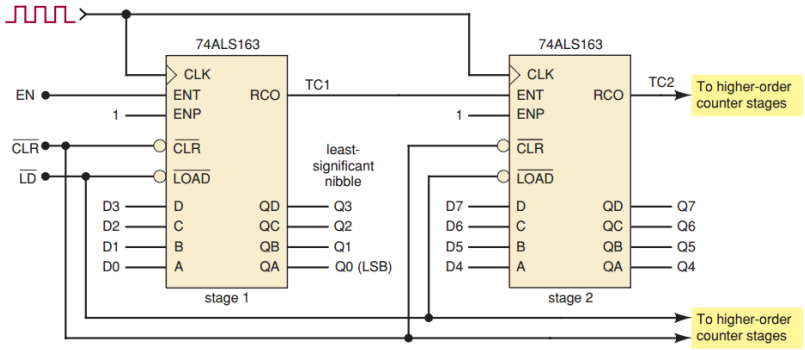
74HC163 vs 74HC191

(a) Determine the output waveform for each counter.
(b) What is the recycling count sequence and modulus for each counter?
(c) Why do they have different count sequences?



Multistage Arrangement

- Many standard IC counters have been designed to make it easy to connect multiple chips together to create circuits with a higher counting range.
- These previous counter chips can be simply connected in a **multistage** or **cascading** arrangement
- two 74ALS163s are connected in a two-stage counter arrangement that produces a recycling, binary sequence from 0 to 255 for a maximum modulus of 256



33

Review Questions

- Describe the function of the inputs LOAD and D, C, B, A.
  - LOAD is the control that enables the parallel loading of the data inputs D C B A (A: LSB).
- Describe the function of the CLR input.
  - CLR is the control that enables the resetting of the counter to 0000.
- True or false: The 74HC161 cannot be preset while CLR is active.
  - True
- What logic levels must be present on the control inputs in order for the 74ALS162 to count pulses that appear on the CLK?
  - All control inputs (CLR, LOAD, ENT, and ENP) on the 74162 must be HIGH.
- What logic levels must be present on the control inputs in order for the 74HC190 to count down with pulses that appear on the CLK?
  - LOAD=1, CTEN=0, and D/U=1 to count down.
- What would be the maximum counting range for a four-stage counter made up of 74HC163 ICs? What is the maximum counting range for 74ALS190 ICs?
  - 74HC163: 0 to 65,535; 74ALS190: 0 to 9999 or 9999 to 0

Counters and Registers

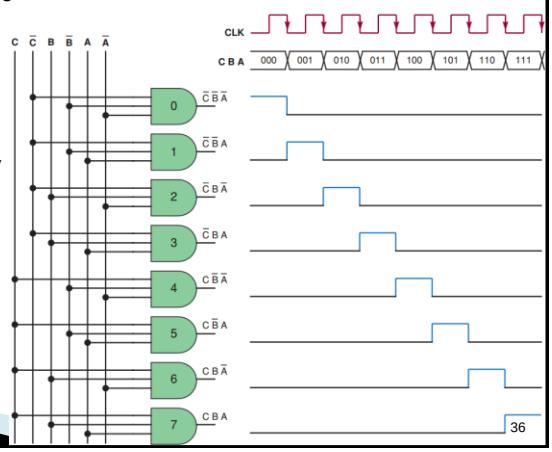
34

Decoding a Counter

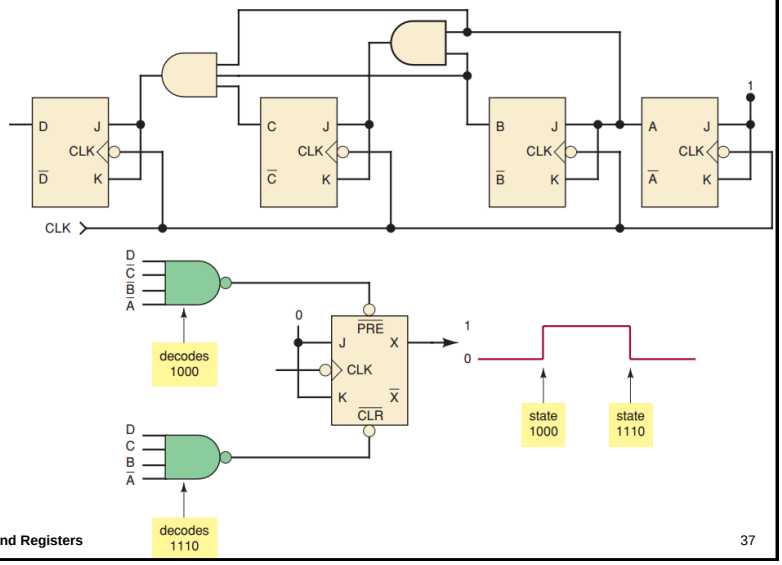
- One of the simplest means for displaying the contents of a counter involves just connecting the output of each FF to a small indicator LED
- Mentally decoding the binary states of the LEDs
  - Becomes inconvenient as the size of the counter increases
- Electronically decoding
  - To control the timing or sequencing of operations automatically without human intervention.
  - Active-High Decoding
  - Active-Low Decoding
  - BCD counter decoding

Active-High Decoding

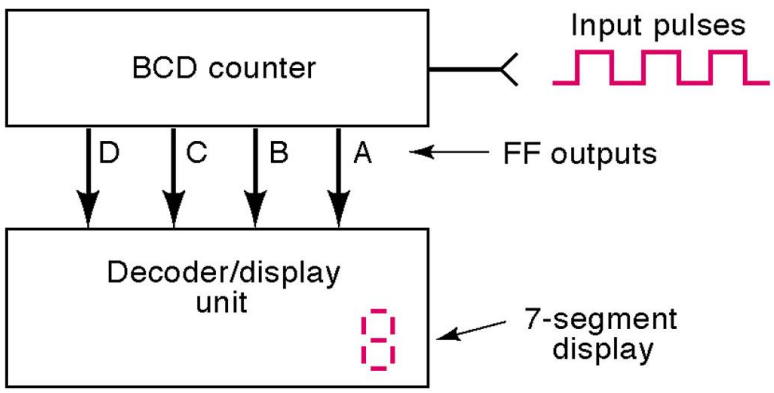
- A decoding network is a logic circuit that generates X different outputs, each of which detects (decodes) the presence of one particular state of the counter.
- The decoder outputs can be designed to produce either a HIGH level when the detection occurs
- Each AND gate produces a HIGH output for one particular state of the counter.
- How many AND gates are required to decode completely all of the states of a MOD-32 binary counter? What are the inputs to the gate that decodes for the count of 21?
  - The decoder requires 32 AND gates. Each gate will have five inputs, one from each FF.



Active-LOW Decoding



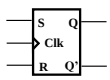
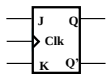
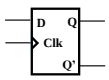
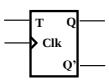
BCD Counter Decoding



Review Questions

- How many gates are needed to decode a six-bit counter fully?
  - 64
- Describe the decoding gate needed to produce a LOW output when a MOD-64 counter is at the counter of 23.
  - 23 = 010111
  - 6-input NAND gate with inputs $\overline{F}E\overline{D}CBA$

Flip-flop types

Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table
SR		S R Q(next) 0 0 Q 0 1 0 1 0 1 1 1 NA	*Q(next)=S+R'Q* *SR=0*	Q Q(next) S R 0 0 0 X 0 1 1 0 1 0 0 1 1 1 X 0
JK		J K Q(next) 0 0 Q 0 1 0 1 0 1 1 1 Q'	*Q(next)=JQ'+K'Q*	Q Q(next) J K 0 0 0 X 0 1 1 X 1 0 X 1 1 1 X 0
D		D Q(next) 0 0 1 1	*Q(next)=D*	Q Q(next) D 0 0 0 0 1 1 1 0 0 1 1 1
T		T Q(next) 0 Q 1 Q'	*Q(next)=TQ'+T'Q*	Q Q(next) T 0 0 0 0 1 1 1 0 1 1 1 0

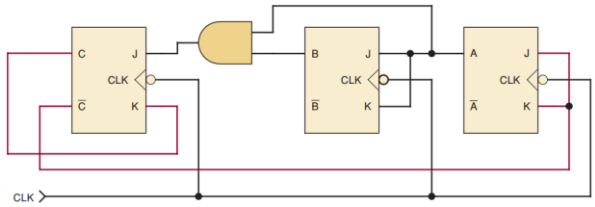
State transitions for FFs (Excitation Table)

Q Present state	Q Next state	S	R	J	K	D	T
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	0	1
1	1	X	0	X	0	1	0

Analyzing Synchronous Counters

- To analyze a synchronous counter design by predicting the FF control inputs for each state of the counter.
- A state transition table is a very useful tool in this analysis process.
- Procedure
  1. Write the logic expression for each FF control input.
  2. Assume a PRESENT state for the counter and apply that combination of bits to the control logic expressions.
  3. The outputs from the control expressions will allow us to predict the commands to each FF and the resulting NEXT state for the counter after clocking.
  4. Repeat the analysis process until the entire count sequence is determined.

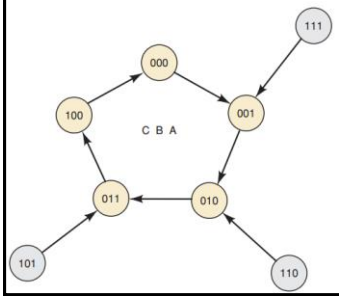
Analyzing Synchronous Counter – Example 1



- Control input expressions
  - $J_C = AB$ $K_C = C$
  - $J_B = \overline{A}$ $K_B = \overline{C}$
  - $J_A = \overline{C}$ $K_A = \overline{C}$
- Assume that the PRESENT state for the counter is CBA=000.
- Applying this combination to the control expressions above will yield J_C K_C=0 0, J_B K_B =0 0, and J_A K_A =1 1. These control inputs will tell FFs C and B to hold and FF A to toggle on the next NGT on CLK.
- Our predicted NEXT state is 001 for CBA.

Analyzing Synchronous Counter – Example 1

PRESENT State			Control Inputs						NEXT State		
C	B	A	J_C	K_C	J_B	K_B	J_A	K_A	C	B	A
0	0	0	0	0	0	0	1	1	0	0	1
0	0	1	0	0	1	1	1	1	0	1	0
0	1	0	0	0	0	0	1	1	0	1	1
0	1	1	1	0	1	1	1	1	1	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	1	0	1	1	1	0	0	0	1	1
1	1	0	0	1	0	0	0	0	0	1	0
1	1	1	1	1	1	1	0	0	0	0	1

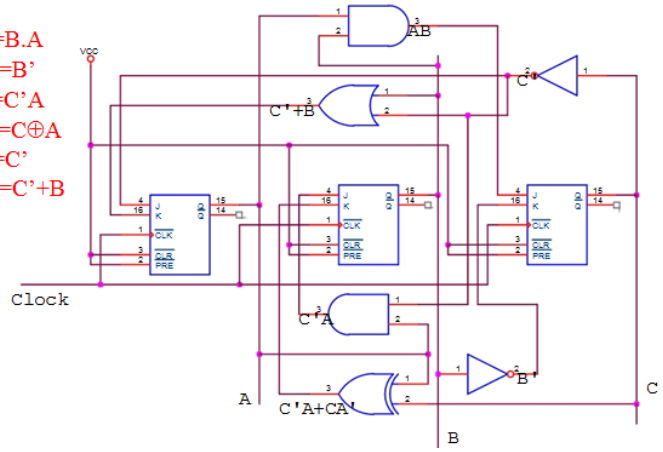


Control input expressions
- $J_C = AB$ $K_C = C$
- $J_B = \overline{A}$ $K_B = \overline{C}$
- $J_A = \overline{C}$ $K_A = \overline{C}$

- Continuing with this process will result in a recycling count sequence of 000, 001, 010, 011, 100, 000.
  - MOD-5 counter
- **Self-correcting** counter is one in which normally unused states will all somehow return to the normal count sequence.

Analyzing Synchronous Counter – Example 2

$J_C=B.A$
$K_C=B'$
$J_B=C'A$
$K_B=C\oplus A$
$J_A=C'$
$K_A=C'+B$



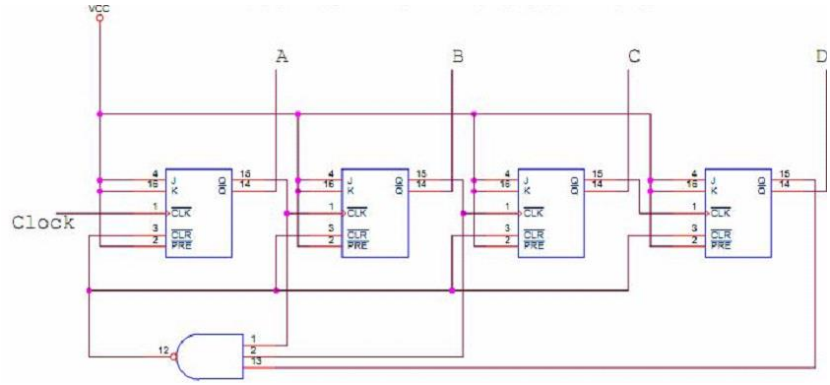
Transition table of the circuit

$J_C=BA$ $K_C=B'$ $J_B=C'A$ $K_B=C\oplus A$ $J_A=C'$ $K_A=C'+B$

PR	C	B	A	JC	KC	JB	KB	JA	KA	C+	B+	A+	N
0	0	0	0	0	1	0	0	1	1	0	0	1	1
1	0	0	1	0	1	1	1	1	1	0	1	0	2
2	0	1	0	0	0	0	0	1	1	0	1	1	3
3	0	1	1	1	0	1	1	1	1	1	0	0	4
4	1	0	0	0	1	0	1	0	0	0	0	0	0
5	1	0	1	0	1	0	0	0	0	0	0	1	1
6	1	1	0	0	0	0	1	0	1	1	0	0	4
7	1	1	1	1	0	0	0	0	1	1	1	0	6

$1\rightarrow2\rightarrow3\rightarrow4\rightarrow0\rightarrow1\ldots.$

Analyzing Synchronous Counter – Quizz



Review questions

- Why is it desirable to avoid having asynchronous controls on counters?
  - We will not have to deal with transient states and possible glitches in output waveforms.
- What tool is useful in the analysis of synchronous counters?
  - State transition table
- What determines the count sequence for a counter circuit?
  - The gates control the count sequence.
- What counter characteristic is described by saying that it is self-correcting?
  - Unused states all lead back to the count sequence of the counter

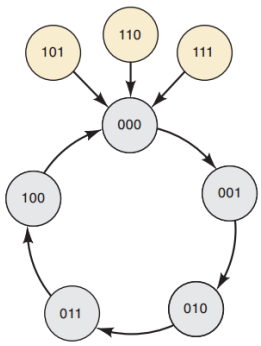
Synchronous Counter Design

- Design a custom counter that follows a sequence that is not a regular binary count pattern, e.g., 000, 010, 101, 001, 110, 000…
- The following steps are applied to design a synchronous counter:
  - Step 1: Define the number of FFs used. It is defined by the number of bits used to express the maximum number in the counter.
  - Step 2: Draw the count diagram including states outside the count circle.
  - Step 3: Based on the diagram in Step 2, display the states transitions between the present state and next state, then fill in the state transition table.
  - Step 4: Define the combinational functions connected to pins J, K or D of FFs.
  - Step 5: Simplify the combinational functions using K-map.
  - Step 6: Draw the circuit diagram.

Sync. Counter Design – Example 1

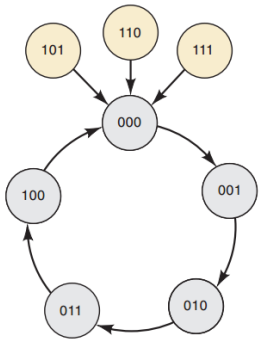
- Design a synchronous counter, count from 0 to 4, using JK FFs

- Step 1: 3 FFs are used.
- Step 2: the count circle consists of 5 states, 3 states outside the count circle.
  - Link the states outside the count circle to any state in the count circle.



Sync. Counter Design – Example 1

- Step 3: Draw the state transition table from the state transition diagram



	PRESENT State			NEXT State		
	C	*B*	*A*	*C*	*B*	*A*
Line 1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	1
4	0	1	1	1	0	0
5	1	0	0	0	0	0
6	1	0	1	0	0	0
7	1	1	0	0	0	0
8	1	1	1	0	0	0

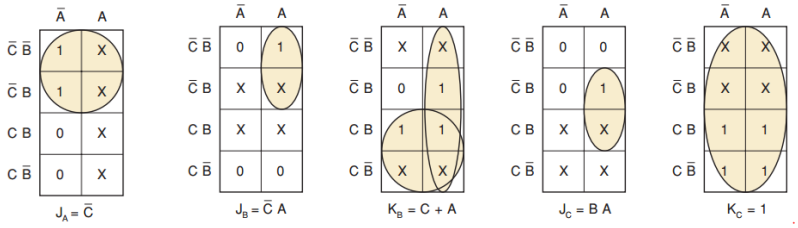
Sync. Counter Design – Example 1

- Step 4: Define the combinational functions connected to pins J,K

	PRESENT State			NEXT State								
	C	*B*	*A*	*C*	*B*	*A*	J_C	K_C	J_B	K_B	J_A	K_A
Line 1	0	0	0	0	0	1	0	*x*	0	*x*	1	*x*
2	0	0	1	0	1	0	0	*x*	1	*x*	*x*	1
3	0	1	0	0	1	1	0	*x*	*x*	0	1	*x*
4	0	1	1	1	0	0	1	*x*	*x*	1	*x*	1
5	1	0	0	0	0	0	*x*	1	0	*x*	0	*x*
6	1	0	1	0	0	0	*x*	1	0	*x*	*x*	1
7	1	1	0	0	0	0	*x*	1	*x*	1	0	*x*
8	1	1	1	0	0	0	*x*	1	*x*	1	*x*	1

Sync. Counter Design – Example 1

- Step 5: Simplify J_C , K_C , J_B , K_B , J_A , K_A using K-map

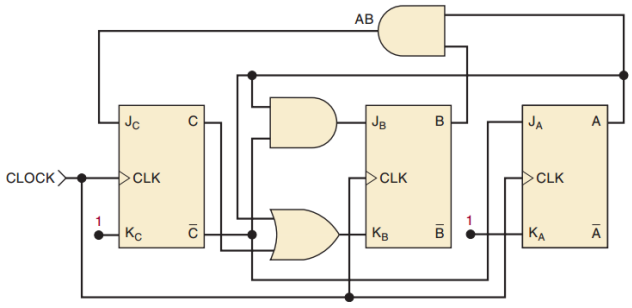


Counters and Registers 58

Sync. Counter Design – Example 1

- Step 6: Implement the final circuit diagram

$J_A = \overline{C}$
$K_A = 1$
$J_B = A\overline{C}$
$K_B = A + C$
$J_C = AB$
$K_C = 1$

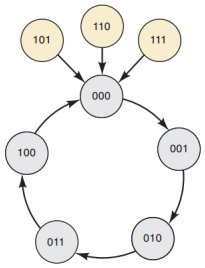


Counters and Registers 59

Sync. Counter Design – Example 2

- Design a synchronous counter, count from 0 to 4, using D FFs
- Step 1, 2, 3 are similar to the previous example
- Step 4:

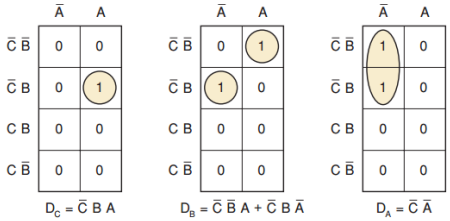
PRESENT State			NEXT State			Control Inputs		
C	B	A	C	B	A	D_C	D_B	D_A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0



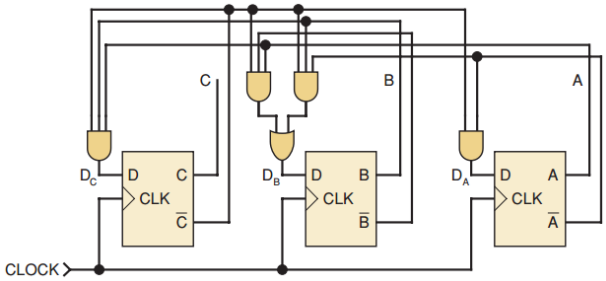
Counters and Registers 60

Sync. Counter Design – Example 2

- Step 5:



- Step 6:



Counters and Registers 61

Exercises

- Design a synchronous counter, count from 0 to 5 using
  - JK FFs.
  - D FFs

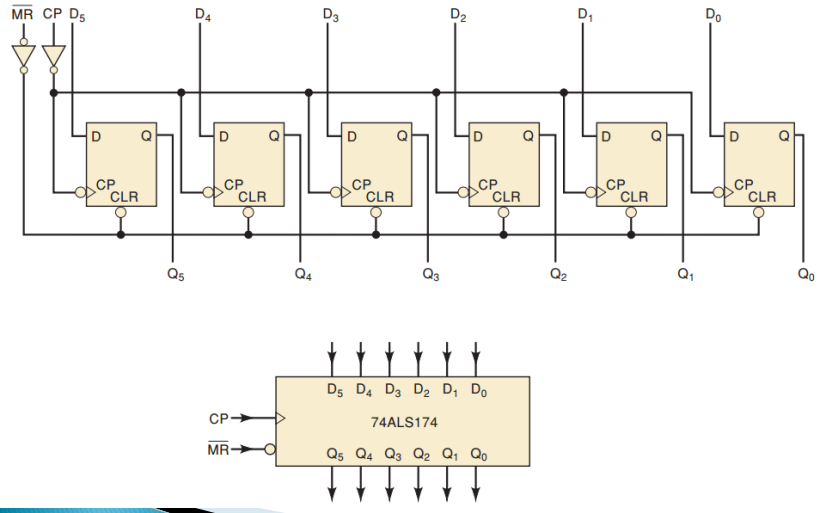
Review questions

- List the six steps in the procedure for designing a synchronous counter
  - See the slide 53.
- What information is contained in a state transition table?
  - It associates every possible PRESENT state with its desired NEXT state.
- What information is contained in the circuit excitation table?
  - It shows the necessary levels at each flip-flop's synchronous input to produce the counter's state transitions
- True or false:The synchronous counter design procedure can be used for the following sequence: 0010, 0011, 0100, 0111, 1010, 1110, 1111, and repeat.
  - True

Registers

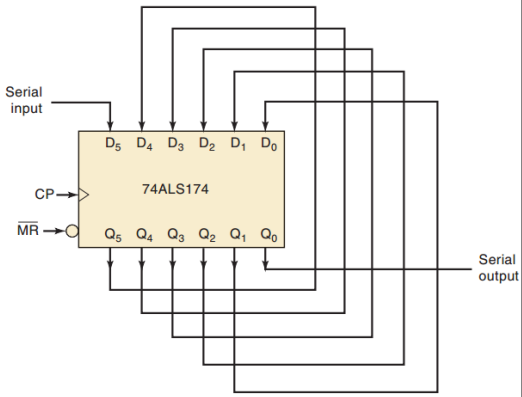
- One FF can save 1 data bit ⇒ a register comprised by n FFs is used to save n data bits.
- A register can shift data to left or right when the clock pulse is excited.
- The various types of registers can be classified according to the manner in which data can be entered into the register for storage and the manner in which data are outputted from the register:
  - Serial input – serial output register (SISO).
  - Serial input – parallel output register (SIPO): 74LS164.
  - Parallel input – serial output register (PISO): 74LS165 .
  - Parallel input – parallel output register (PIPO): 74LS174

PIPO (74ALS174)



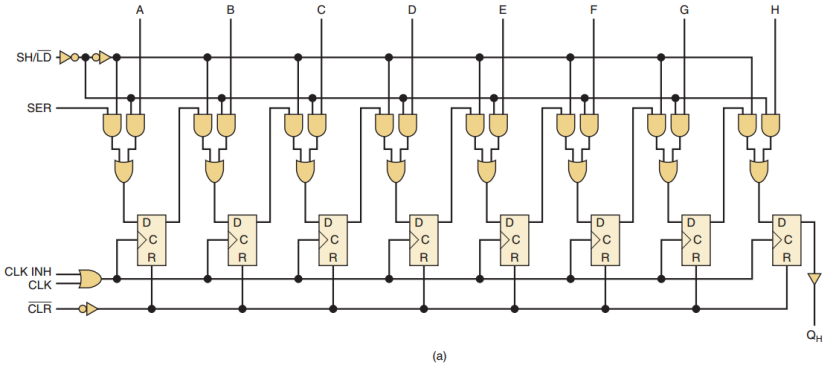
74ALS174

- Show how to connect the 74ALS174 so that it operates as a serial shift register with data shifting on each PGT of CP as follows: Serial input → Q5 → Q4 → Q3 → Q2 → Q1 → Q0. In other words, serial data will enter at D5 and will output at Q0.
- How would you connect two 74ALS174s to operate as a 12-bit shift register?

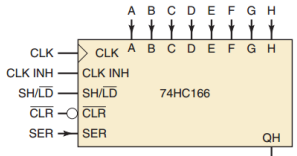


Counters and Registers

SISO (74ALS166)



(a)

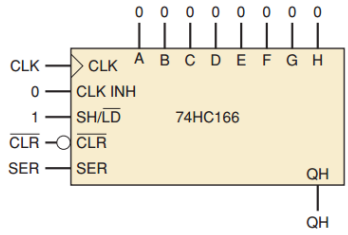


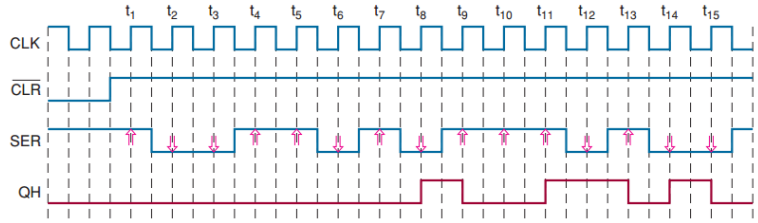
INPUTS						OUTPUTS		
						INTERNAL		
$\overline{CLR}$	SH/$\overline{LD}$	CLK INH	CLK	SER	PARALLEL A . . . H	Q_A	Q_B	Q_H
L	X	X	X	X	X	L	L	L
H	X	L	L	X	X	Q_{A0}	Q_{B0}	Q_{H0}
H	L	L	↑	X	a . . . h	a	b	h
H	H	L	↑	H	X	H	Q_{An}	Q_{Gn}
H	H	L	↑	L	X	L	Q_{An}	Q_{Gn}
H	X	H	↑	X	X	Q_{A0}	Q_{B0}	Q_{H0}

Counters and Registers

67

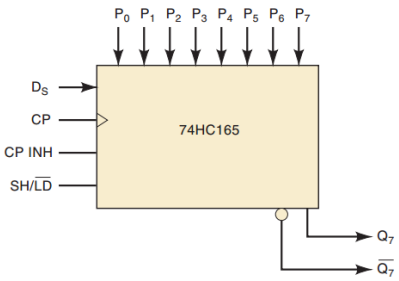
74ALS166





PISO (74ALS165)

- Eight-bit parallel in/serial out register
- It actually has serial data entry via DS and asynchronous parallel data entry via P0 through P7
- CP is the clock input used for the shifting operation.
  - The clock inhibit input, CP INH is used to inhibit the effect of the CP input.
- The shift/load input, controls which operation is taking place—shifting or parallel loading.
- Only accessible FF outputs are Q7



Function Table

Inputs			Operation
SH/$\overline{LD}$	CP	CP INH	
L	X	X	Parallel load
H	H	X	No change
H	X	H	No change
H	↓	L	Shifting
H	L	↓	Shifting

H = high level
L = low level
X = immaterial
↓ = PGT

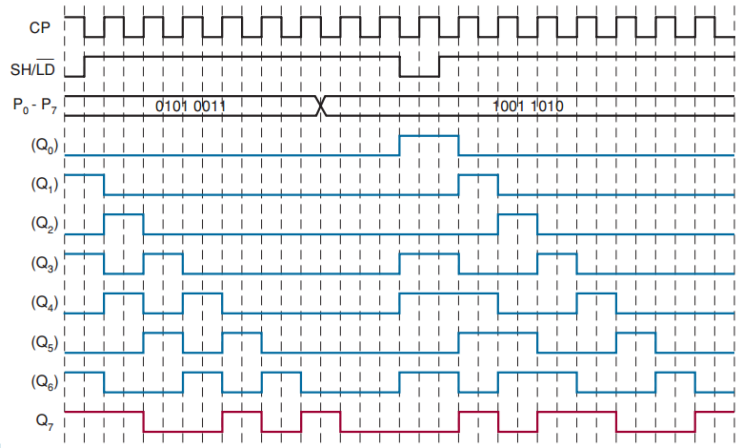
74ALS165

Function Table

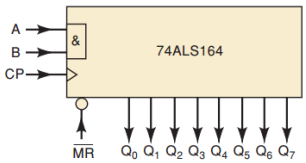
Inputs			Operation
SH/$\overline{LD}$	CP	CP INH	
L	X	X	Parallel load
H	H	X	No change
H	X	H	No change
H	↓	L	Shifting
H	L	↓	Shifting

H = high level
L = low level
X = immaterial
↓ = PGT

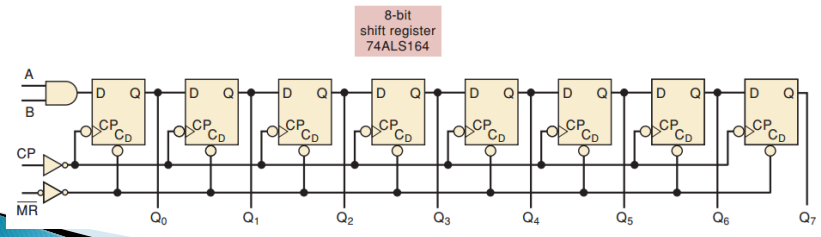
- DS=0 and CP INH = 0



SIPO (74ALS164)

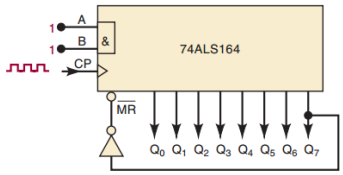


- Eight bit serial in/parallel out shift register with each FF output externally accessible.
- An AND gate combines inputs A and B to produce the serial input to flip-flop Q0
- The shift operation occurs on the PGTs of the clock input CP.
- The input provides asynchronous resetting of all FFs on a LOW level



74ALS164

- Assume that the initial contents of the 74ALS164 register are 00000000
- Determine the sequence of states as clock pulses are applied



Input pulse number	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0
4	1	1	1	1	0	0	0	0
5	1	1	1	1	1	0	0	0
6	1	1	1	1	1	1	0	0
7	1	1	1	1	1	1	1	0
8	1	1	1	1	1	1	1	1

Recycles

Temporary state

Review questions

- What kind of register can have a complete binary number loaded into it in one operation, and then have it shifted out one bit at a time?
  - Parallel in/serial out
- True or false: A serial in/parallel out register can have all of its bits displayed at one time.
  - True
- What type of register can have data entered into it only one bit at a time, but has all data bits available as outputs?
  - Serial in/parallel out
- In what type of register do we store data one bit at a time and have access to only one output bit at a time?
  - Serial in/serial out
- How does the parallel data entry differ for the 74165 and the 74174?
  - The 74165 uses asynchronous parallel data transfer; the 74174 uses synchronous parallel data transfer.
- How does the CP INH input of the 74ALS165 work?
  - A HIGH prevents shifting on CPs

Summary

- In asynchronous (ripple) counters, the clock signal is applied to the LSB FF, and all other FFs are clocked by the output of the preceding FF.
- A counter's MOD number is the number of stable states in its counting cycle; it is also the maximum frequency-division ratio.
- The normal (maximum) MOD number of a counter is 2^N. One way to modify a counter's MOD number is to add circuitry that will cause it to recycle before it reaches its normal last count.
- Counters can be cascaded (chained together) to produce greater counting ranges and frequency-division ratios.
- In a synchronous (parallel) counter, all of the FFs are simultaneously clocked from the input clock signal.
- The maximum clock frequency for an asynchronous counter, fmax, decreases as the number of bits increases. For a synchronous counter, fmax remains the same, regardless of the number of bits.
- A decade counter is any MOD-10 counter. A BCD counter is a decade counter that sequences through the 10 BCD codes (0-9).
- A presettable counter can be loaded with any desired starting count.

Summary (cont)

- An up/down counter can be commanded to count up or count down.
- Logic gates can be used to decode (detect) any or all states of a counter.
- The count sequence for a synchronous counter can be easily determined by using a PRESENT state/NEXT state table that lists all possible states, the flip-flop input control information, and the resulting NEXT states.
- Synchronous counters with arbitrary counting sequences can be implemented by following a standard design procedure.
- Numerous IC registers are available and can be classified according to whether their inputs are parallel (all bits entered simultaneously), serial (one bit at a time), or both. Likewise, registers can have outputs that are parallel (all bits available simultaneously) or serial (one bit available at a time).
- A sequential logic system uses FFs, counters, and registers, along with logic gates. Its outputs and sequence of operations depend on present and past inputs.
- A ring counter is actually an N-bit shift register that recirculates a single 1 continuously, thereby acting as a MOD-N counter. A Johnson counter is a modified ring counter that operates as MOD-2N counter