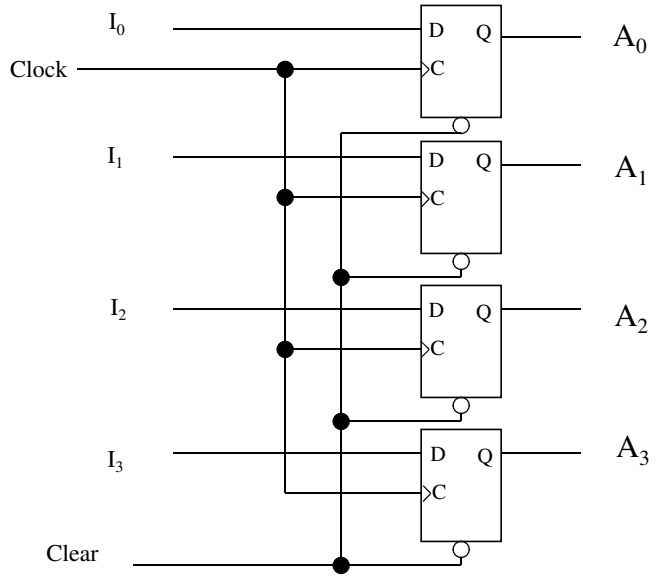
CSC 370 - Computer Architecture and Organization

Lecture 6: Registers and Counters

Registers

- A register is a group of flip-flops.
  - Each flip-flop stores one bit of data; *n* flip-flops are required to store *n* bits of data.
  - There are several different types of registers available commercially.
  - The simplest design is a register consisting only of flip-flops, with no other gates in the circuit.
- ***Loading the register*** – transfer of new data into the register.
- The flip-flops share a common clock pulse (frequently using a buffer to reduce power requirements).
- *Output* could be sampled at any time.
- *Clearing* the flip-flop (placing zeroes in all its bit) can be done through a special terminal on the flip-flop.

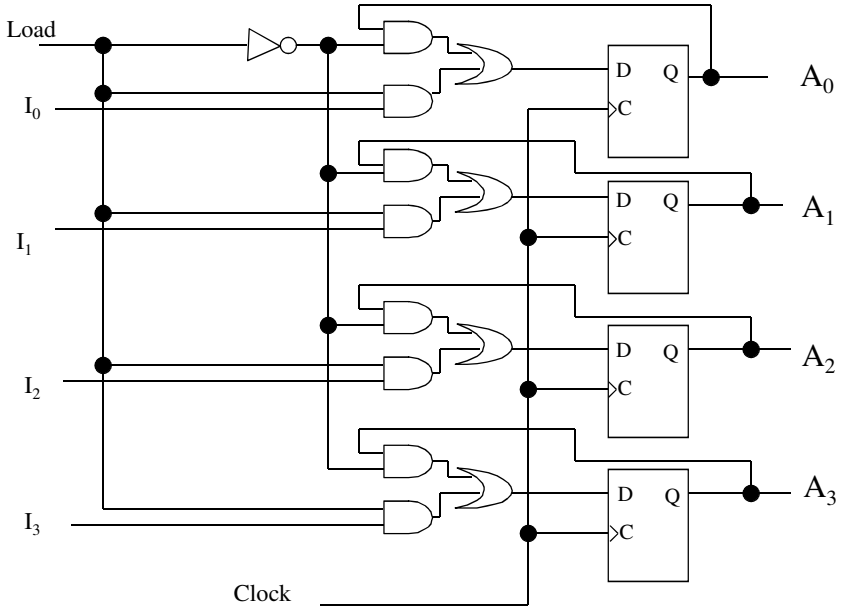
4-bit Register



Registers With Parallel Load

- The clock usually provides a steady stream of pulses which are applied to all flip-flops in the system.
- A separate control system is needed to determine when to load a particular register.
- The Register with Parallel Load has a separate load input.
  - When it is cleared, the register receives it output as input.
  - When it is set, it received the load input.

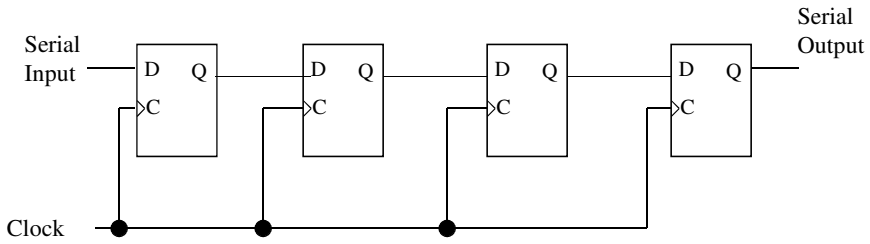
4-bit Register With Parallel Load



Shift Registers

- A shift register is a register which can shift its data in one or both directions.
- The simplest shift register simply connects the flip-flops to their respective neighbor with the clock controlling the operation.
- If we wish to shift on some clock pulses but not others, we can inhibit the clock pulses on which we do not to shift.

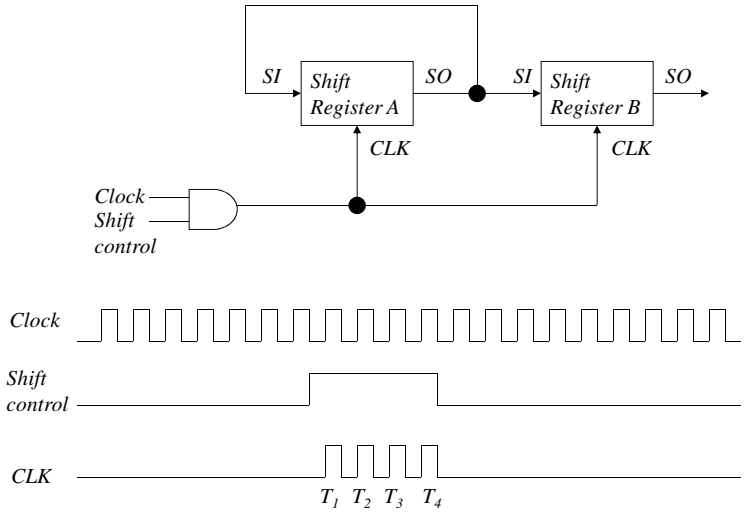
4-Bit Shift Register



Serial Transfer

- A digital system is operating in a serial mode when information is transferred and manipulated one bit at a time, with bits transferred out of the source register into the destination register.
- This is different from parallel transfer where all the bits of a register are transferred at once.
- Serial transfer of information from register A to register B is done with shift registers, where the serial output from register A serves as the serial input for register B.

Serial Transfer From Register A to Register B



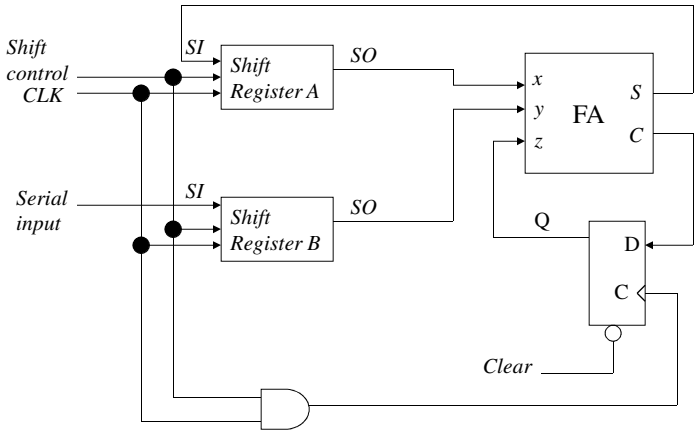
Serial Transfer – State Table

Timing Pulse	Shift Register A	Shift Register B
Initial value	1 0 1 1	0 0 1 0
After T_1	1 1 0 1	1 0 0 1
After T_2	1 1 1 0	1 1 0 0
After T_3	0 1 1 1	0 1 1 0
After T_4	1 0 1 1	1 0 1 1

Serial Addition

- While operations are usually parallel because it is faster, serial operations require less equipment.
- Serial addition can demonstrate this point, allowing us to perform addition with a single full adder and our two addends stored in a pair of shift registers, with the carry-out going into a D-type flip-flop (and going back in as the carry-in).
- Initially, register A and the carry flip-flop are cleared to 0.
- The augend is initially placed in register B and serial addition is used to place it into register A.
- The sum is placed in shift register A, replacing the augend.

Serial Adder



Redesigning the Serial Adder

- We will use a pair of shift registers whose outputs will be x and y respectively. These are corresponding bits of the addends.
- S is the sum bit produced and a flip-flop will hold the carry bit as the flip-flop's state *Q*.
- We will implement it using a *JK* flip-flop

State Table for the Serial Adder

Present State	Inputs		Next State	Output	Flip-flop Inputs	
Q(t)	**X**	**Y**	**Q(t+1)**	**S**	**J_Q**	**K_Q**
0	0	0	0	0	0	X
0	0	1	0	1	0	X
0	1	0	0	1	0	X
0	1	1	1	0	1	X
1	0	0	0	1	X	1
1	0	1	1	0	X	0
1	1	0	1	0	X	0
1	1	1	1	1	X	0

Karnaugh Map for J_Q

Q \ xy	00	01	11	10
0			1	
1	X	X	X	X

$J_Q = xy$

Karnaugh Map for K_Q

Q \ xy	00	01	11	10
0	X	X	X	X
1	1			

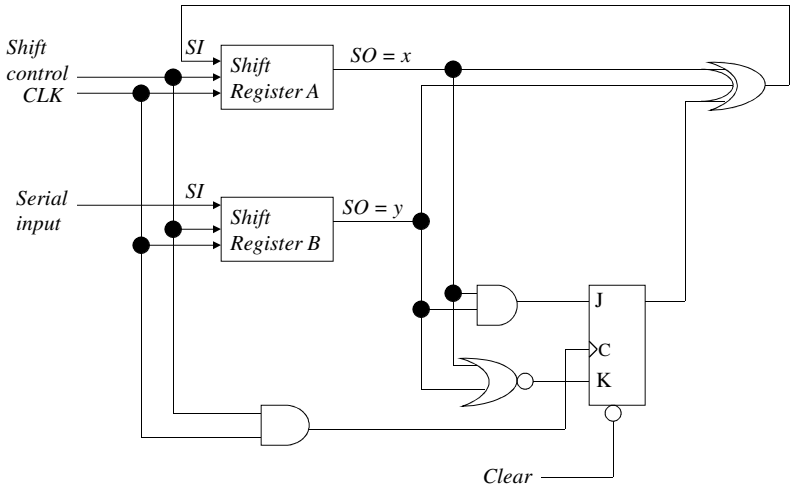
$K_Q = x'y' = (x + y)'$

Karnaugh Map for S

Q \ xy	00	01	11	10
0		1		1
1	1		1	

$$S = x \oplus y \oplus Q$$

Serial Adder



Bi-directional Shift Registers

- A shift register that can shift in one direction is called a ***unidirectional shift register***.
- A shift register that can shift in either direction is called a ***bi-directional shift register***.
- Some shift register also allow for the simple transfer of data.

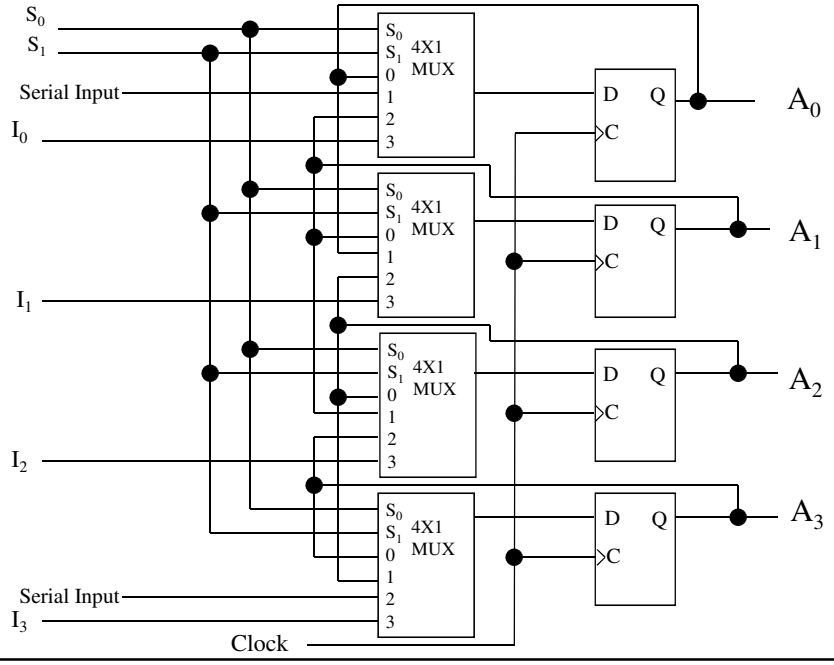
General Shift Register

- The most general shift register have all of these capabilities:
  - An input for clock pulses to synchronize all operations.
  - A shift-right operation and serial line input line associated with the shift-right.
  - A shift-left operation and serial line input line associated with the shift-left.
  - A parallel load operation and n input lines associated with the parallel transfer.
  - n parallel output lines.
  - A control state that leaves the information in the register unchanged even though the clock pulses are applied.

Function Table For General Shift Register

S_1	S_0	**Register operation**
0	0	No change
0	1	Shift Right (down)
1	0	Shift Left (up)
1	1	Parallel load

Bidirectional Shift Register With Parallel Load



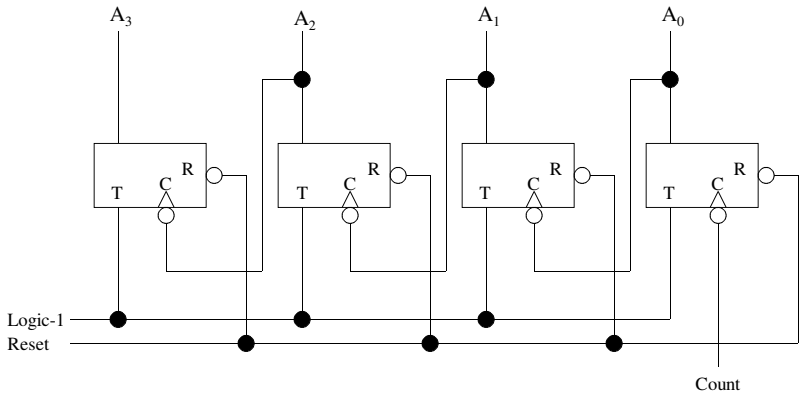
Counters

- A register that goes through a prescribed sequences of states upon the application of an input pulse is called a ***counter***.
- The input pulse may be a clock pulse or may have some other origin.
- A counter that goes through a binary sequence is called a ***binary counter***.
- An n-bit binary counter uses n flip-flops and can count from 0 to 2^n - 1.

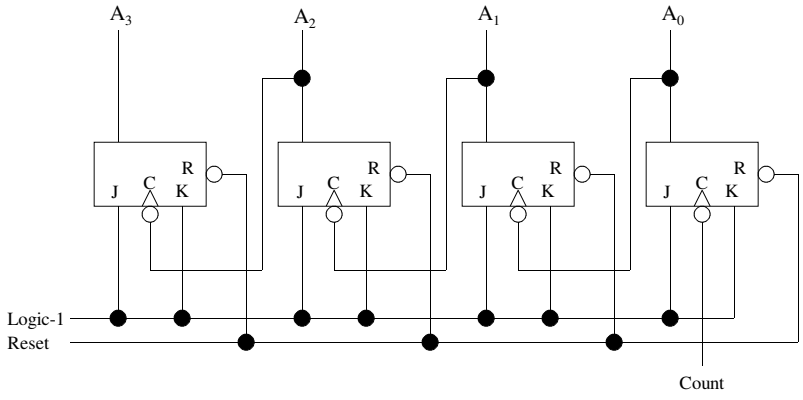
Ripple Counters

- Counters are either ripple counters or synchronous counters.
- In synchronous counters, all flip-flops receive the common clock pulse; therefore they change at the same time.
- In ripple counters, the output of one flip-flop is used as a source for triggering others.

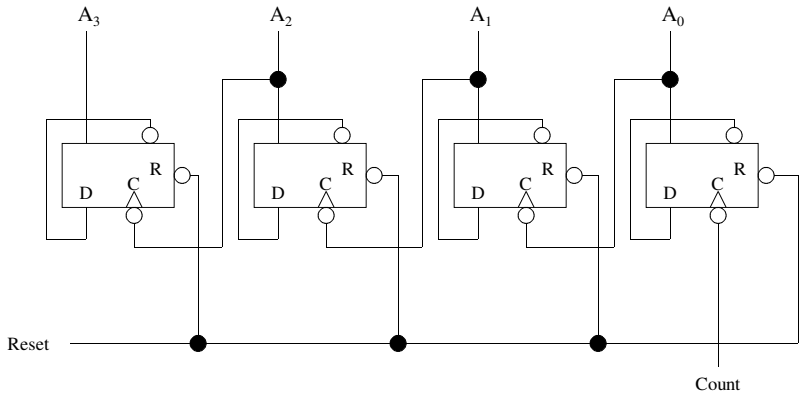
4-Bit Ripple Counter Using T-Type Flip-flop



4-Bit Ripple Counter Using JK-Type Flip-flop



4-Bit Ripple Counter Using D-Type Flip-flop

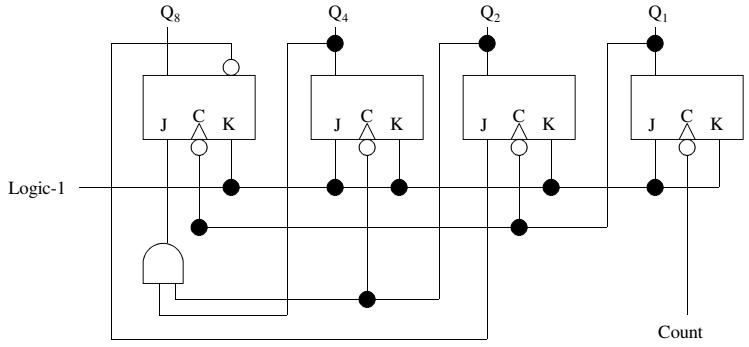


Binary Count Sequence

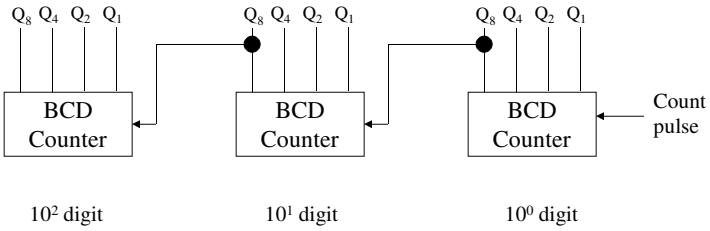
A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

BCD Ripple Counter

- A binary-coded decimal ripple counter will return to 0 after it reaches 9, this necessarily changes the logic



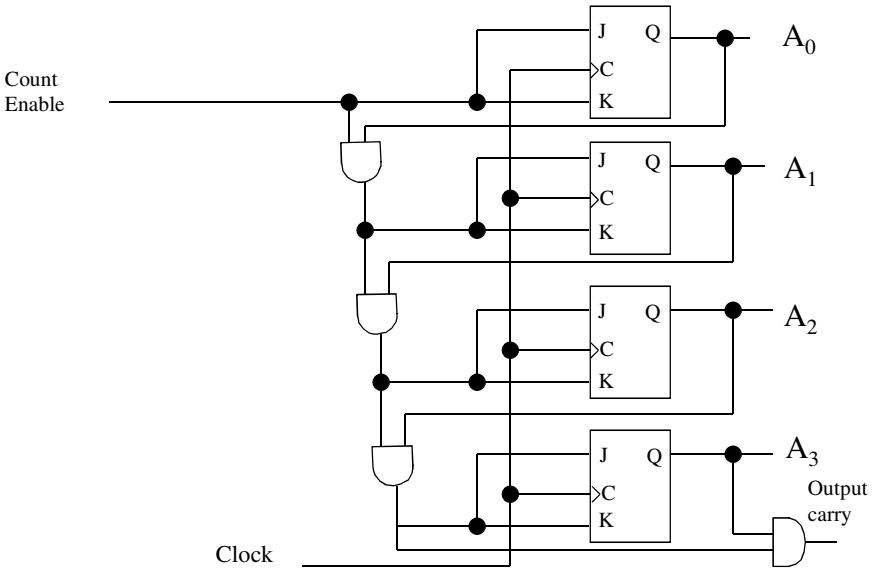
Three-Decade BCD Counter



Binary Counters

- A ***counter*** is a register that goes through a predetermined sequence of states as input pulses are applied.
- Almost all digital equipment will contain counters; they are used for counting the occurrences of a particular event and are useful in generating timing signals.
- An n-bit counter uses n flip-flops and are have any value in the range 0 to 2^n-1.
- We notice in our sequences that the lowest significance bit is complemented with every count and the other bits are complemented from one count to the next when all the lower bits are set.

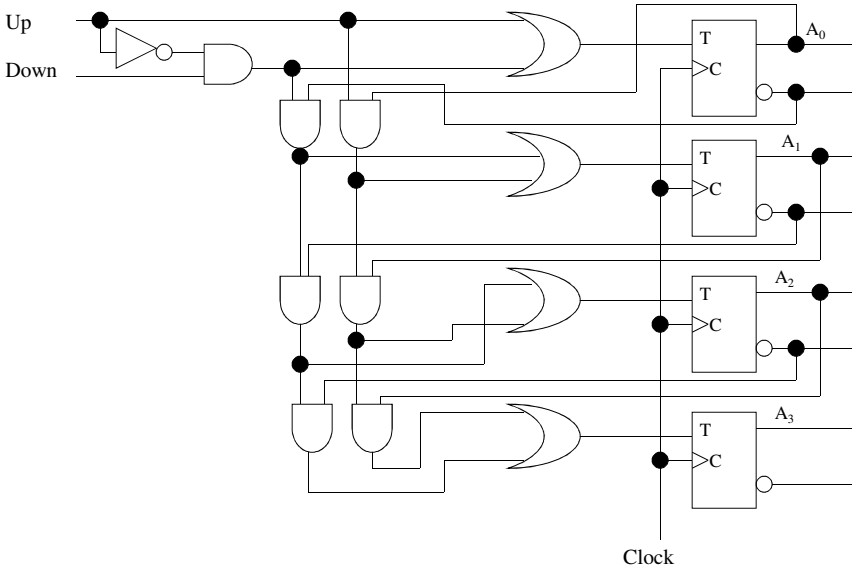
4-bit Synchronous Binary Counter



Up-Down Binary Counter

- A count-down binary counter will go through binary states in reverse order.
- E.g., a 4-bit count-down binary counter will start at 0000, go to 1111, then 1110, and so on down to 0000.
- As in regular counters, the least significant bit is always complemented. But higher bits are complemented only if the lower bits are all 0.
- We can design a counter that can go in either direction, depending on the control inputs.

4-Bit Up-Down Binary Counter



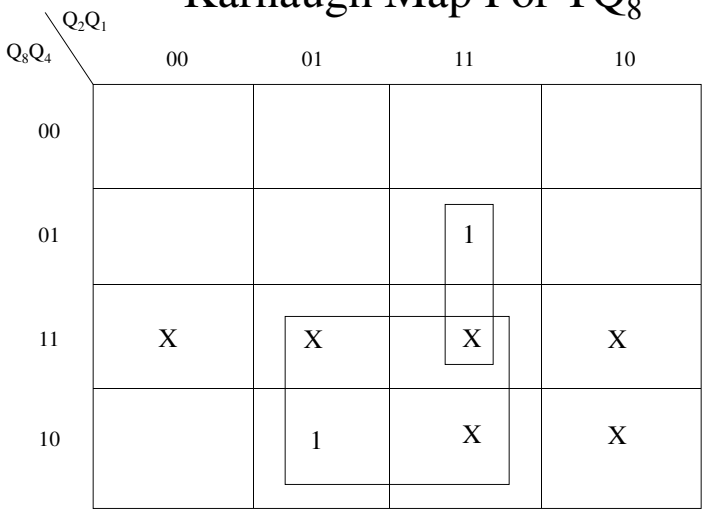
BCD Counter

- BCD counters only go through states 0000 to 0001 up to 1001.
- The pattern is as regular as binary counter, so we must go through the design process.

State Table for BCD Counter

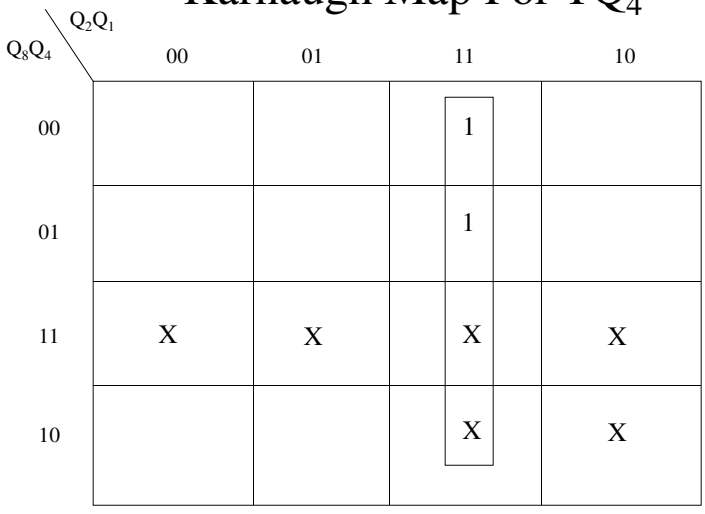
Present State				Next State				Output	Flip-flop Inputs			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	TQ_8	TQ_4	TQ_2	TQ_1
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1
1	0	1	0	X	X	X	X	0	X	X	X	X
1	0	1	1	X	X	X	X	0	X	X	X	X
1	1	0	0	X	X	X	X	0	X	X	X	X
1	1	0	1	X	X	X	X	0	X	X	X	X
1	1	1	0	X	X	X	X	0	X	X	X	X
1	1	1	1	X	X	X	X	0	X	X	X	X

Karnaugh Map For TQ_8



$T_{Q8} = Q_8Q_1 + Q_4Q_2Q_1$

Karnaugh Map For TQ_4



$T_{Q4} = Q_2Q_1$

Karnaugh Map For TQ_2

Q_8Q_4 \ Q_2Q_1	00	01	11	10
00		1	1	
01		1	1	
11	X	X	X	X
10			X	X

$T_{Q2} = Q_8'Q_1$

Karnaugh Map For TQ_1

Q_8Q_4 \ Q_2Q_1	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

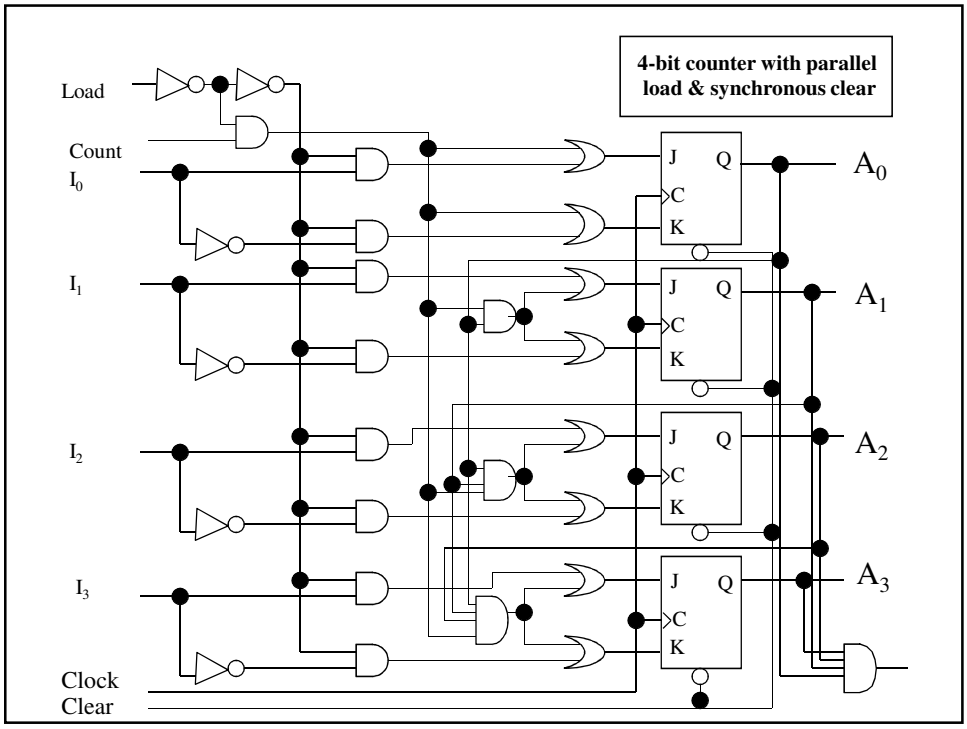
$T_{Q1} = 1$

4-Bit Counter With Parallel Load & Synchronous Clear

- Counters often need to be preset with a pre-specified value before counting begins.
- We also need the capability of clearing all bits simultaneously.

Function Table For 4-Bit Parallel Load Counter

Clock	**Clear**	**Load**	**Count**	**Operation**
	1	0	0	No change
	1	0	1	Increment count
	1	1	X	Load inputs I_0 through I_3
	0	X	X	Clear outputs



Counters With Unused States

- An n-bit counter has 2^n states, but there are occasions when we wish to use less than the total number of states available.
- The unused states may be treated as "don't-care" conditions (or assigned to specific next states).
- Because outside interference may land the counter in these states, we must ensure that the counter can find its way back to a valid state.

State Table For Counter With Unused States

Present State			Next State			Flip-flop Inputs					
A	**B**	**C**	**A**	**B**	**C**	**J_A**	**K_A**	**J_B**	**K_B**	**J_C**	**K_C**
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
0	*1*	*1*	*1*	*0*	*0*	*1*	*X*	*X*	*1*	*X*	*1*
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X
1	*1*	*1*	*0*	*0*	*0*	*X*	*1*	*X*	*1*	*X*	*1*

Karnaugh Maps for J_A and K_A

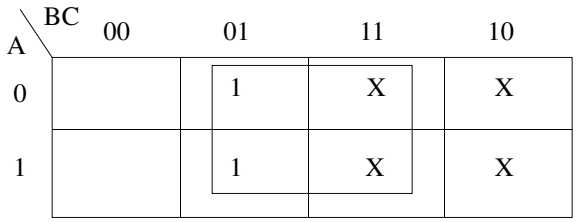
A \ BC	00	01	11	10
0			1	1
1	X	X	X	X

$J_A = B$

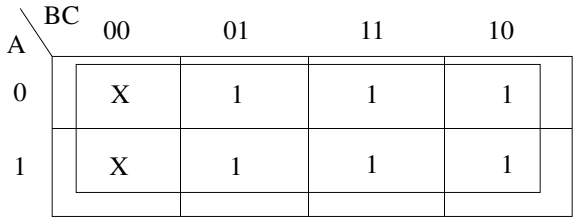
A \ BC	00	01	11	10
0	X	X	X	X
1			1	1

$K_A = B$

Karnaugh Maps for J_B and K_B

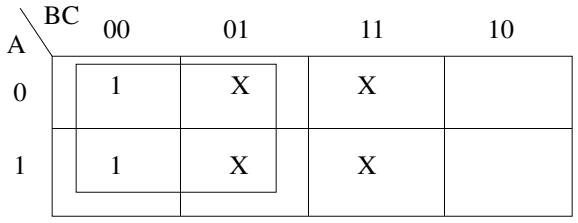


$J_B = C$

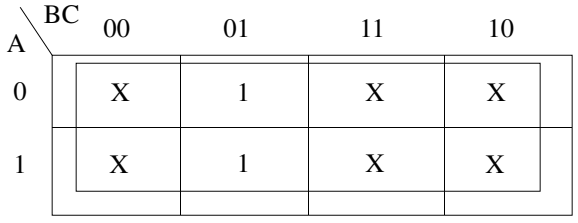


$K_B = 1$

Karnaugh Maps for J_C and K_C

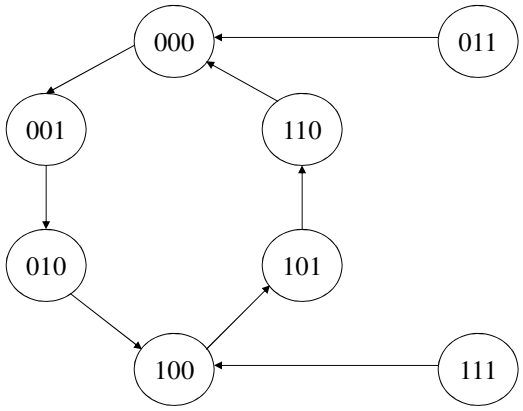


$J_C = B'$

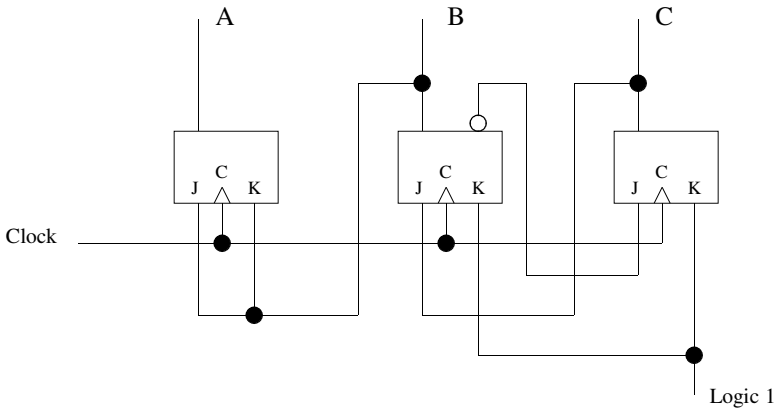


$K_C = 1$

State Diagram For Counter With Unused States



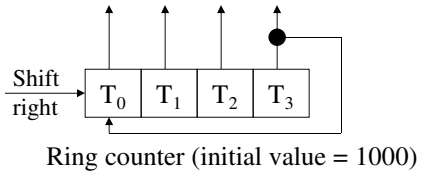
Logic Diagram For Counter With Unused States



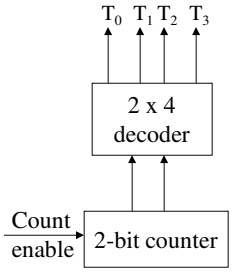
Ring Counter

- Computers need timing signals that indicate the sequence in which certain operations will take place.
- These can be generated by ring counters, circular shift registers with only one flip-flop set at any time.
- The alternative to a 4-bit ring counter is a 2-bit counter that goes through 4 distinct states and uses a decoder.

Ring Counter Vs. Counter and Decoder

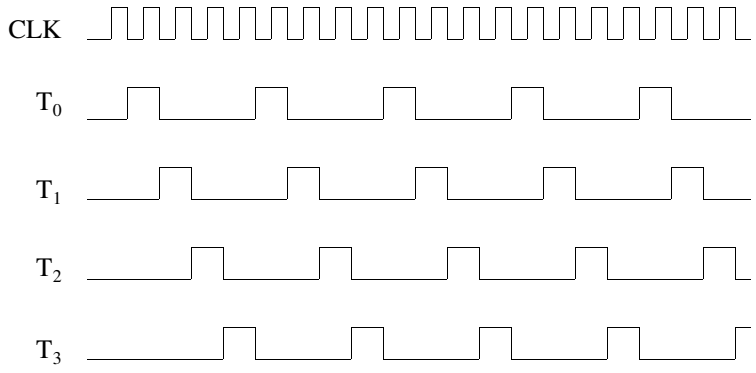


Ring counter (initial value = 1000)



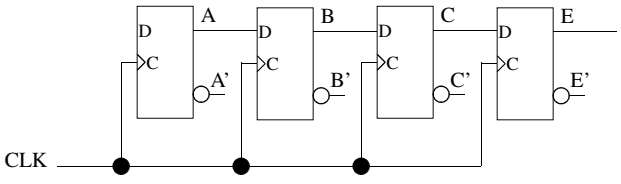
Counter and decoder

Sequence of Timing Signals From the Ring Counter



Switch-tail Ring Counters

- Switch-tail ring counter can double the number of states that a ring counter can provide.



4-stage switch-tail ring counter

State Table For Switch-tail Counter

Flip-flop Outputs

Seq. Num.	A	B	C	E	AND gate required for output
1	0	0	0	0	A'E'
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	A'B
7	0	0	1	1	B'C
8	0	0	0	1	C'E

Johnson Counters