

# Module 5 – Registers and Counters

**CSE-103 Digital Computer Logic**

**Athar Mahboob**

**WWW:** <http://www.atharmahboob.com>

**Email:** [athar@atharmahboob.com](mailto:athar@atharmahboob.com)

# Module Outline

---

- Definitions
- Registers
- Shift Registers
- Ripple Counter
- Synchronous Binary Counters
- Other Counters

# Definitions

---

- **Register** – a register consists of a set of flip-flops together with gates that implement their state transitions. In a narrower sense the term register is applied to a set of flip-flops, with added combinational gates that perform data processing tasks.
- **Counter** – a counter is a register that goes through a pre-determined sequence of states upon the application of clock pulses. The gates in a counter are connected in such a way as to produce the prescribed sequence of binary states.

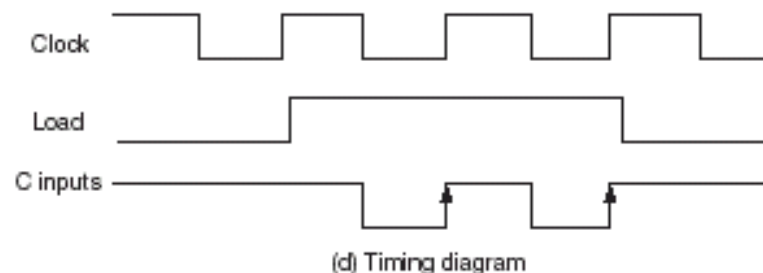
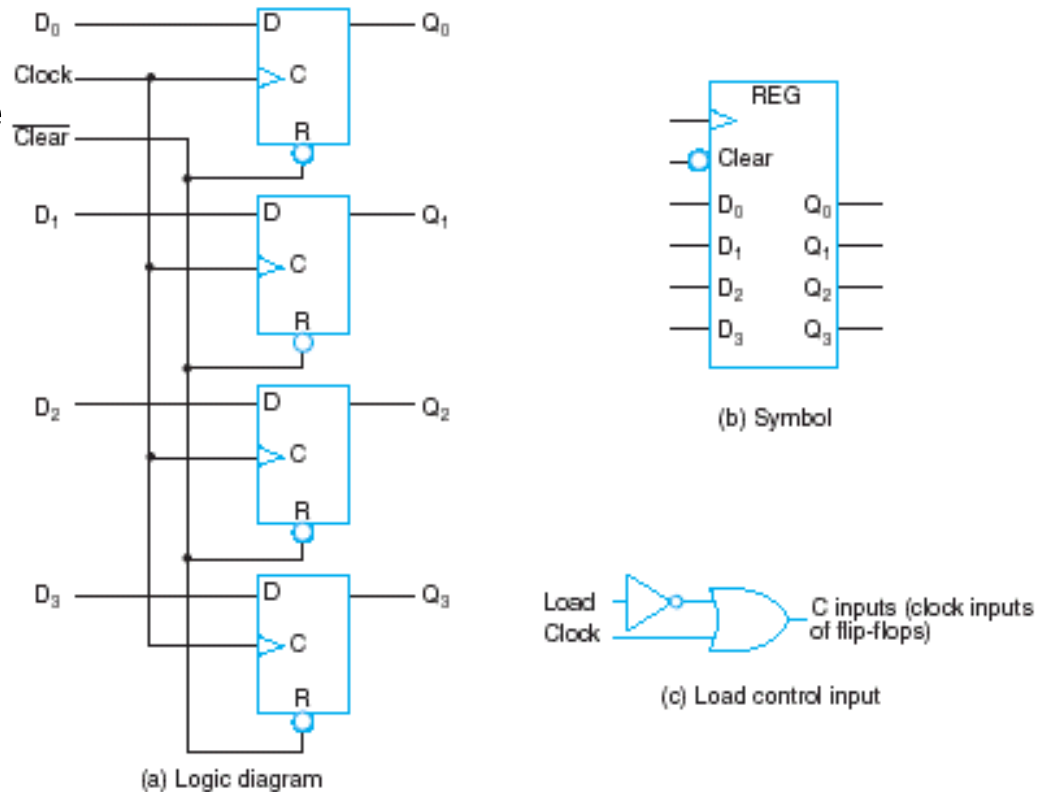
# 4-Bit Register

The common clock triggers all the flip-flops on the rising edge of each pulse and the binary data available at the four D inputs are transferred to 4-bit register.

The Clear input is used to asynchronously reset (clear to 0) the register.

The transfer of new information into the register is known as loading the register.

If all the bits of a register are loaded simultaneously with a common clock pulse, we say that loading is done in parallel.



# Load control input

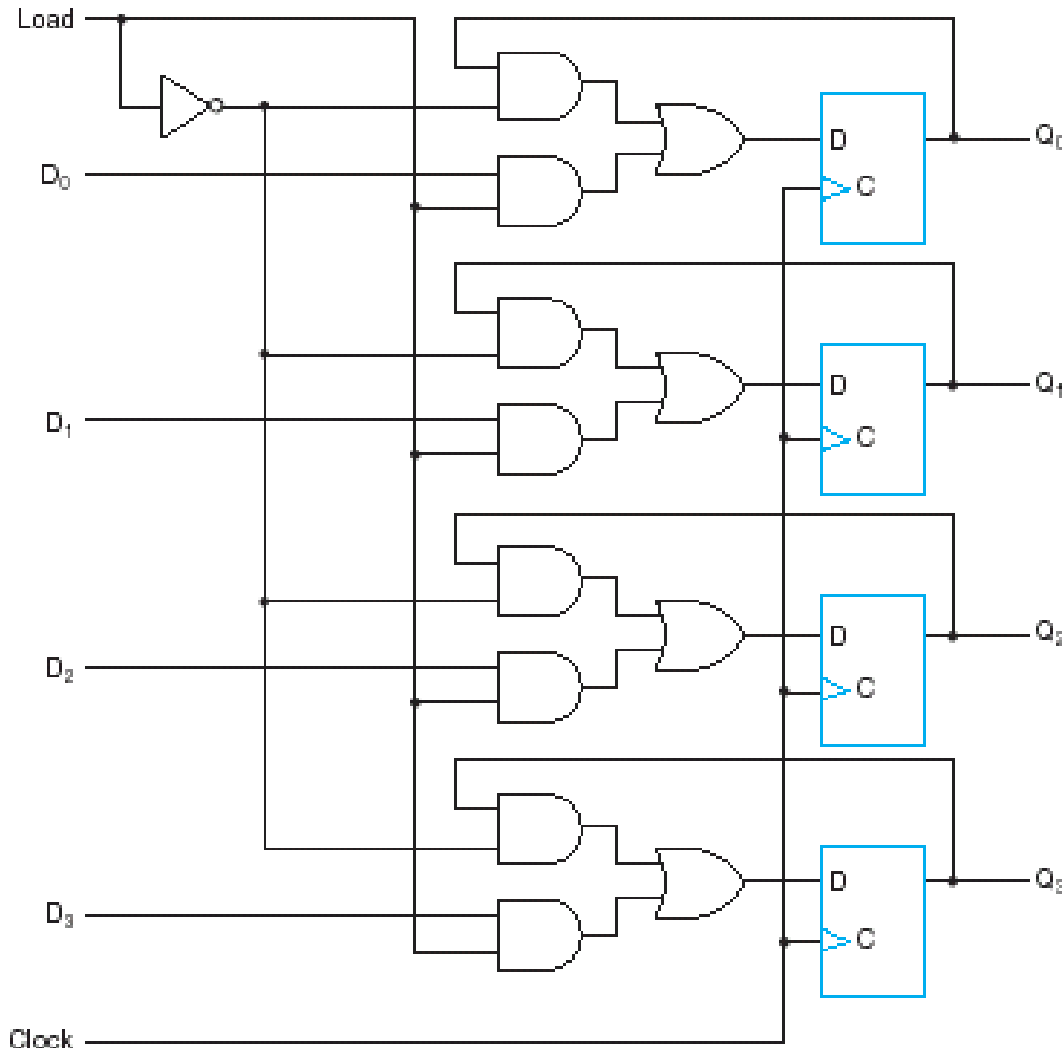
---



If the contents of the register are to be unchanged at a clock input then the clock must be prevented from reaching the clock input of the register

A Load control input is combined with a clock. When the Load input is 1 the C input is Clock and the register is clocked normally. When the Load control is 0 the C input is constant 1 without any transitions so the register is not clocked and the outputs remain unchanged.

# 4-Bit Register with Parallel Load

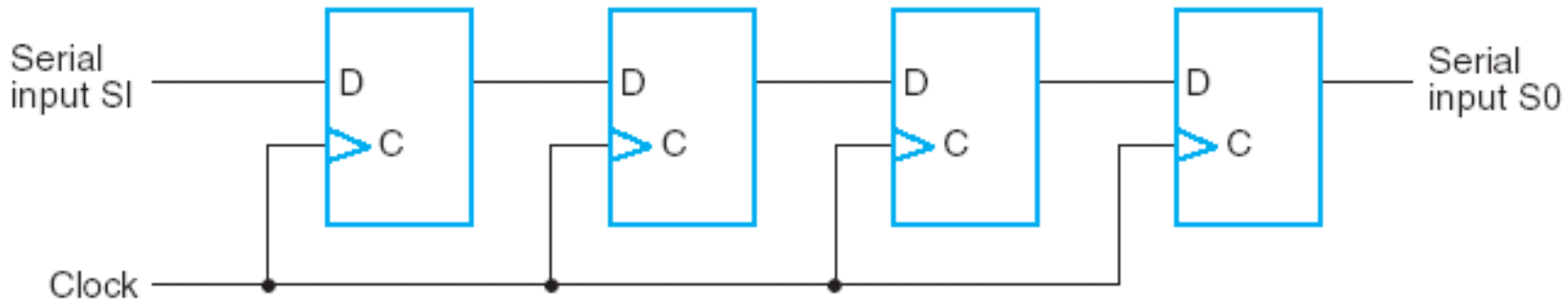


Load input is gated with D inputs rather than with the clock inputs

When the Load is 0 the output of the flip-flop is fed back to the D input (of the flip-flop) so the output remains unchanged

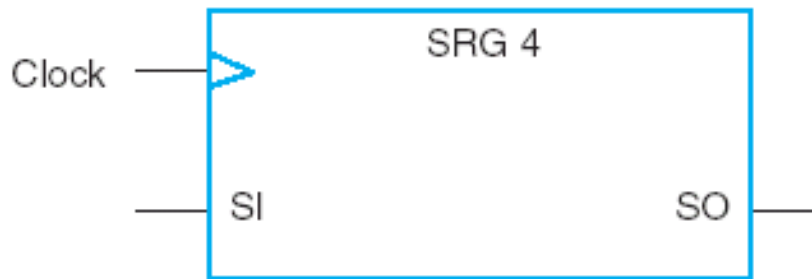
When the Load is 1 the input to the register is determined by the D input to the register

# 4-Bit Shift Register



(a) Logic diagram

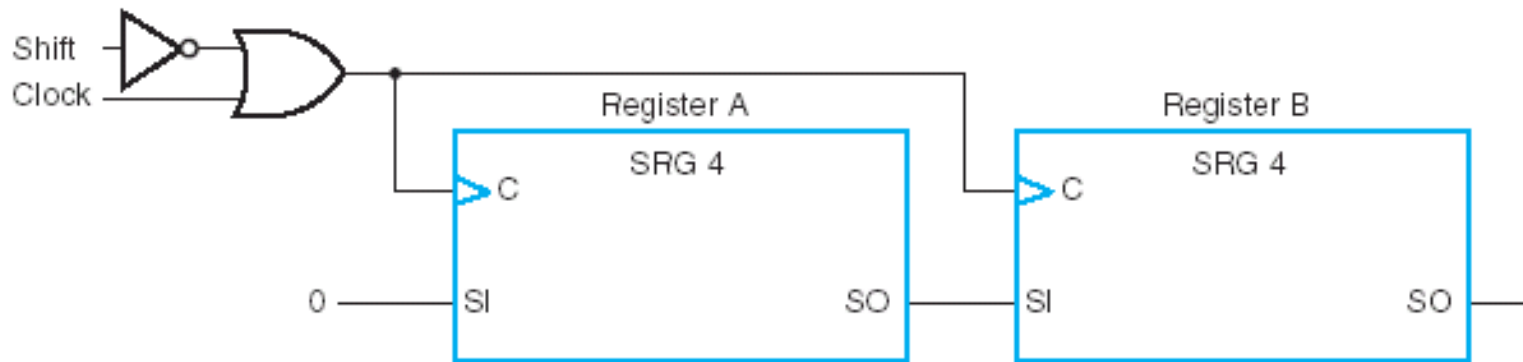
A register that is capable of shifting its stored bits laterally in one or both directions is called a shift register.



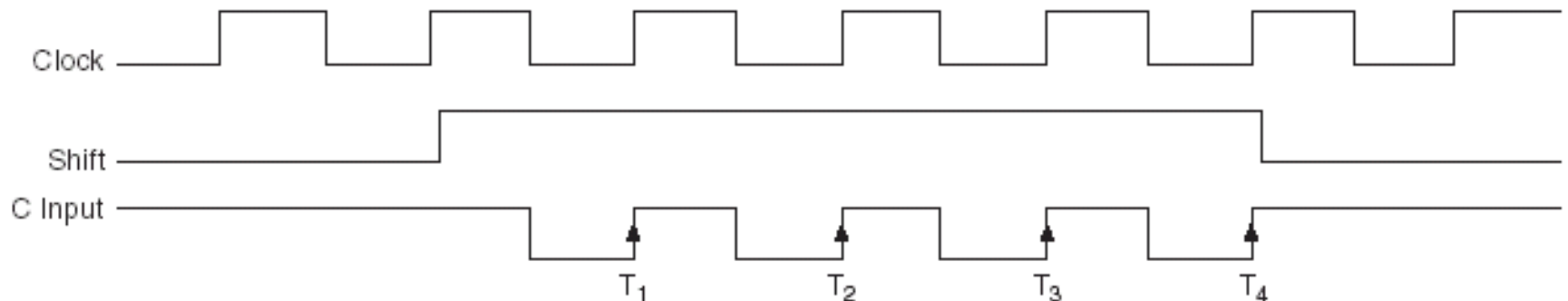
(b) Symbol

The logical configuration of a shift register consists of a chain of flip-flops in cascade with the output of one flip-flop connected to the input of the next flip-flop.

# Serial Transfer



(a) Block diagram



(b) Timing diagram

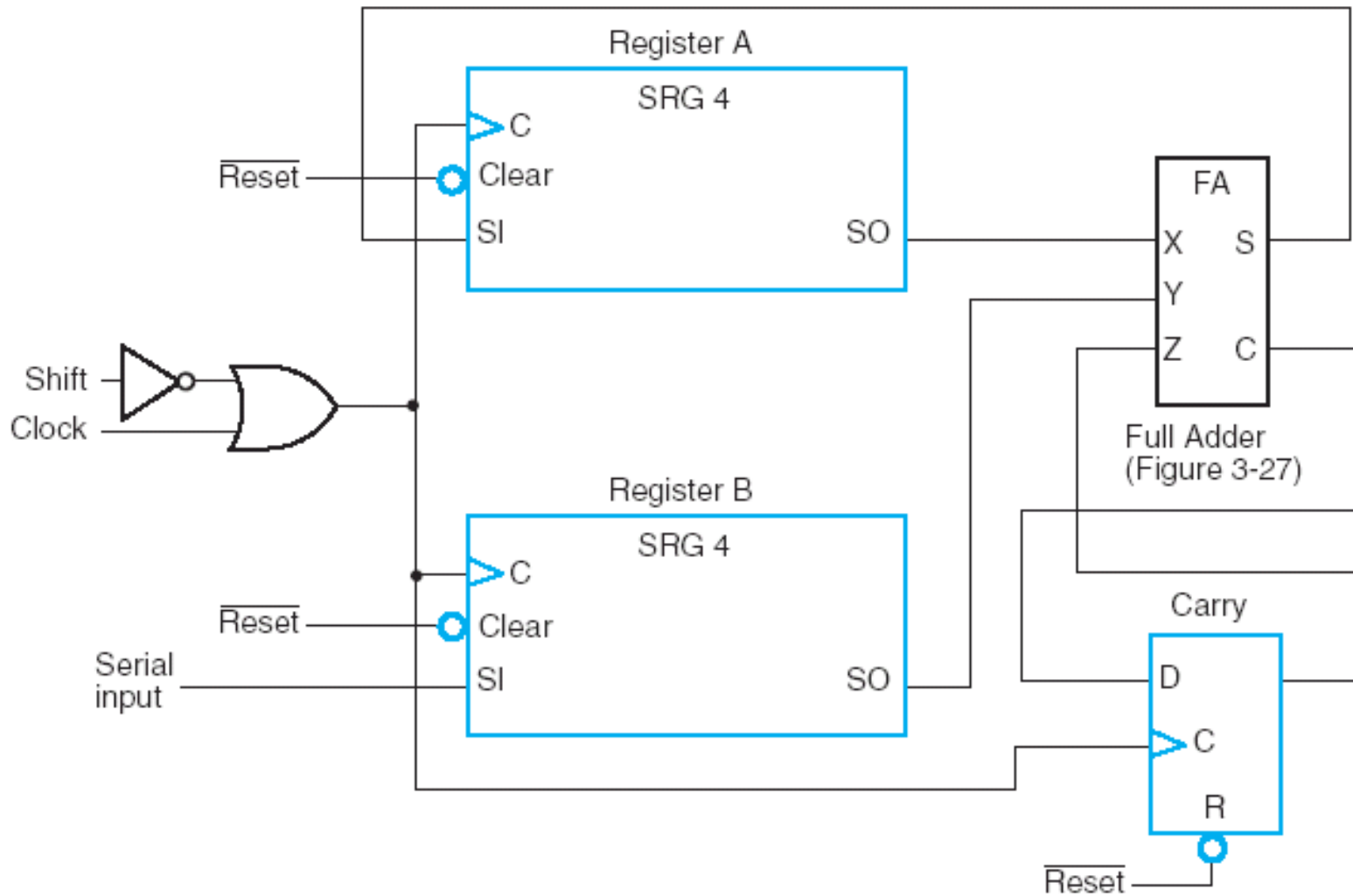
A digital system is said to operate in serial mode when the information in the system is transferred or manipulated one bit at a time.

# Serial Addition

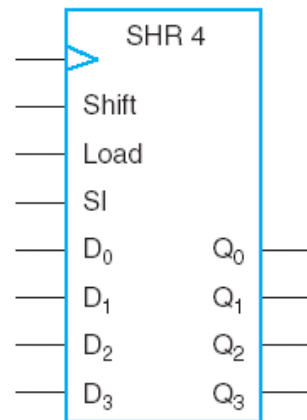
---

- An example of time-space trade-off.
- We can add two 4-bit number using just one full-adder rather than 4
- However, it will take longer to add.
- The circuit shown can be used to add a sequence of 4 bit numbers with the result being accumulated in Shift Register A

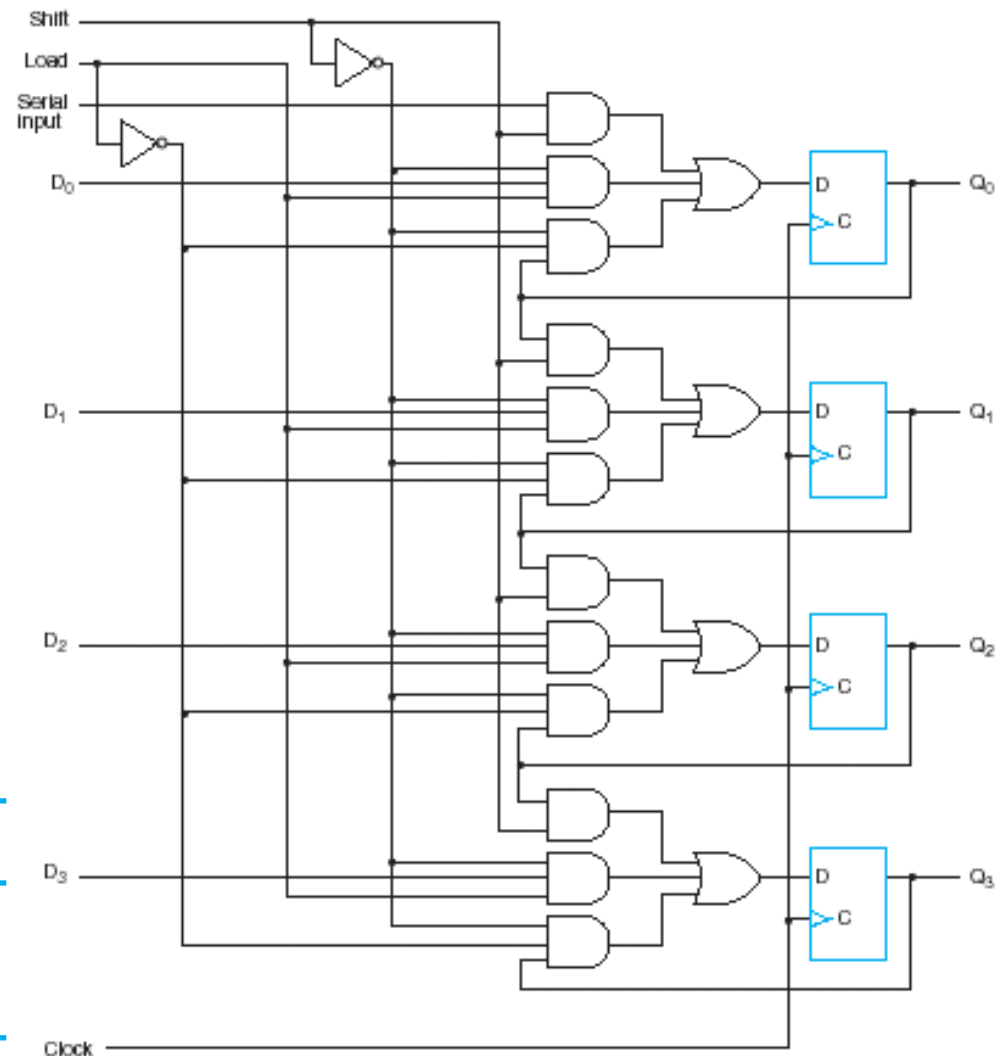
# Serial Addition



# Shift Register with Parallel Load

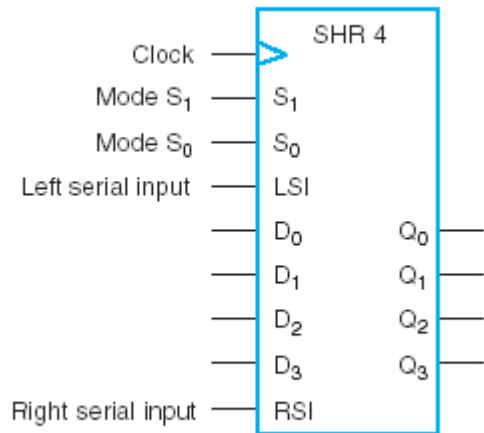


(b) Symbol



Shift	Load	Operation
0	0	No change
0	1	Load parallel data
1	×	Shift down from $Q_0$ to $Q_3$

# Bidirectional Shift Register with Parallel Load

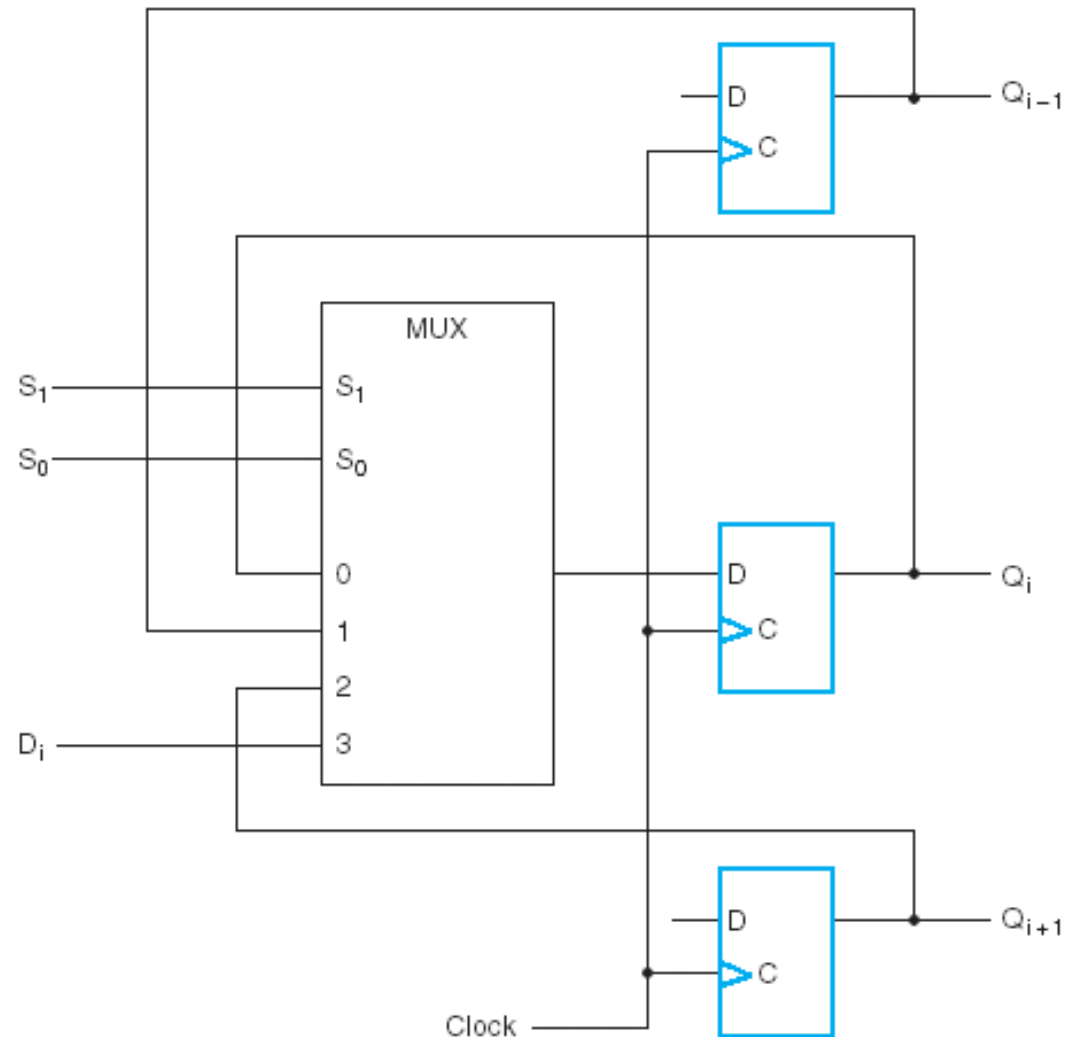


(b) Symbol

**TABLE 5-3**  
Function Table for the Register of Figure 5-7

Mode control		Register Operation
$S_1$	$S_0$	
0	0	No change
0	1	Shift down
1	0	Shift up
1	1	Parallel load

Table 5-3 Function Table for the Register of Figure 5-7



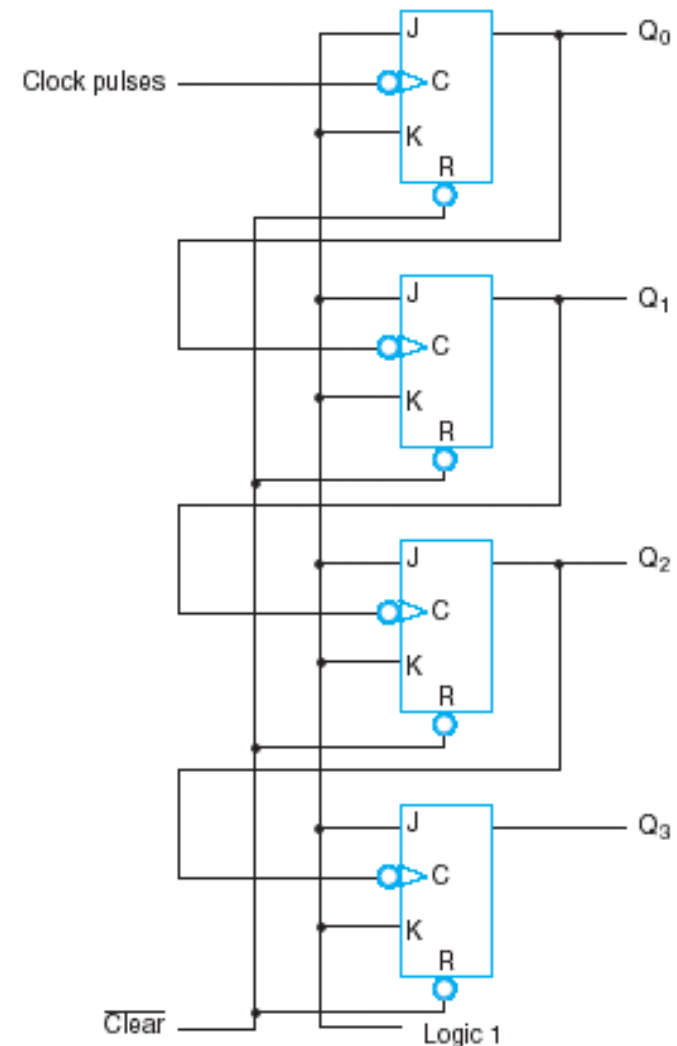
# Types of Counters

---

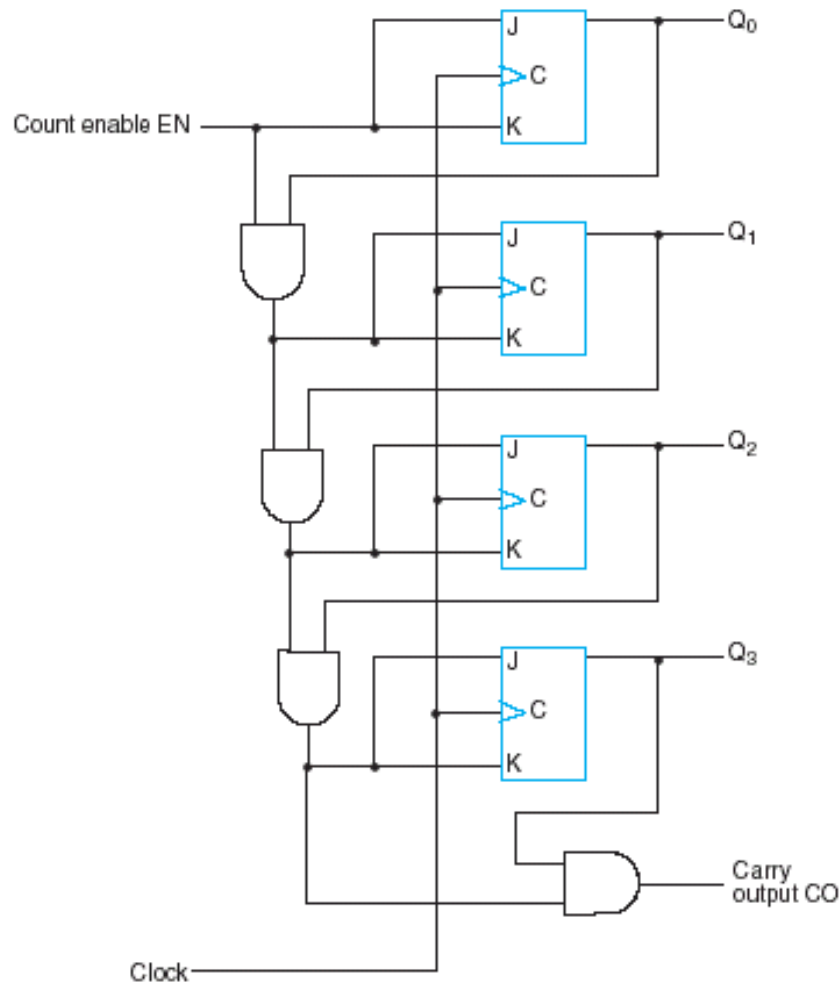
- **Binary Counter** – A counter that follows the binary sequence is called a binary counter.
- **Ripple Counters** – in a ripple counter the flip-flop output transition serves as a source of triggering other flip-flops.
- **Synchronous Counters** – in synchronous counters, the C input of all the flip-flops is triggered by the common clock signal.

# 4-Bit Ripple Counter

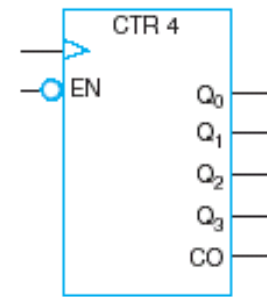
Upward Counting Sequence				Downward Counting Sequence			
Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0
0	0	1	0	1	1	0	1
0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	0	1	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	0	0
1	0	0	0	0	1	1	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0



# 4-Bit Synchronous Binary Counter

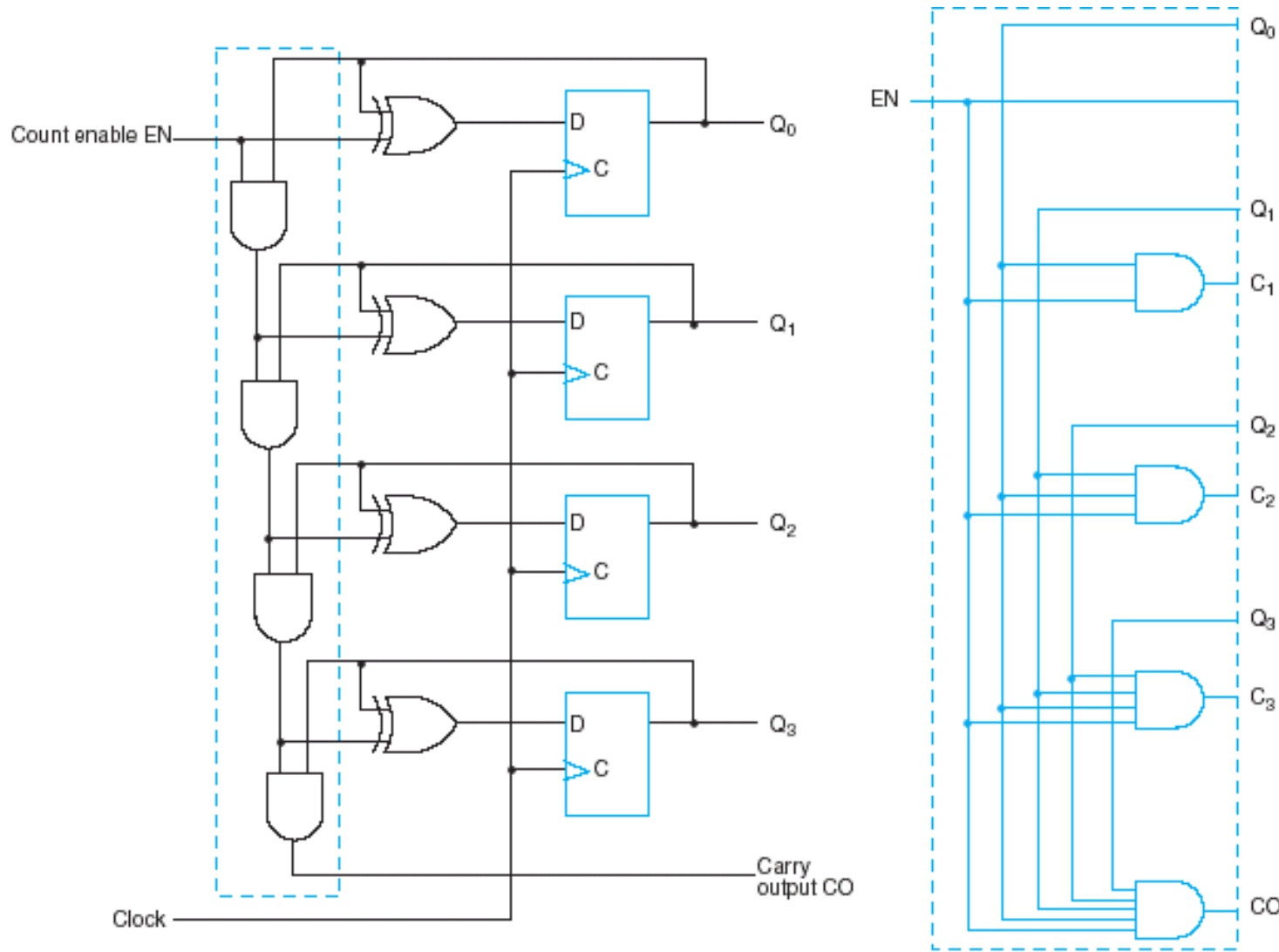


(a) Logic diagram



(b) Symbol

# 4-Bit Binary Counter with D Flip-Flops

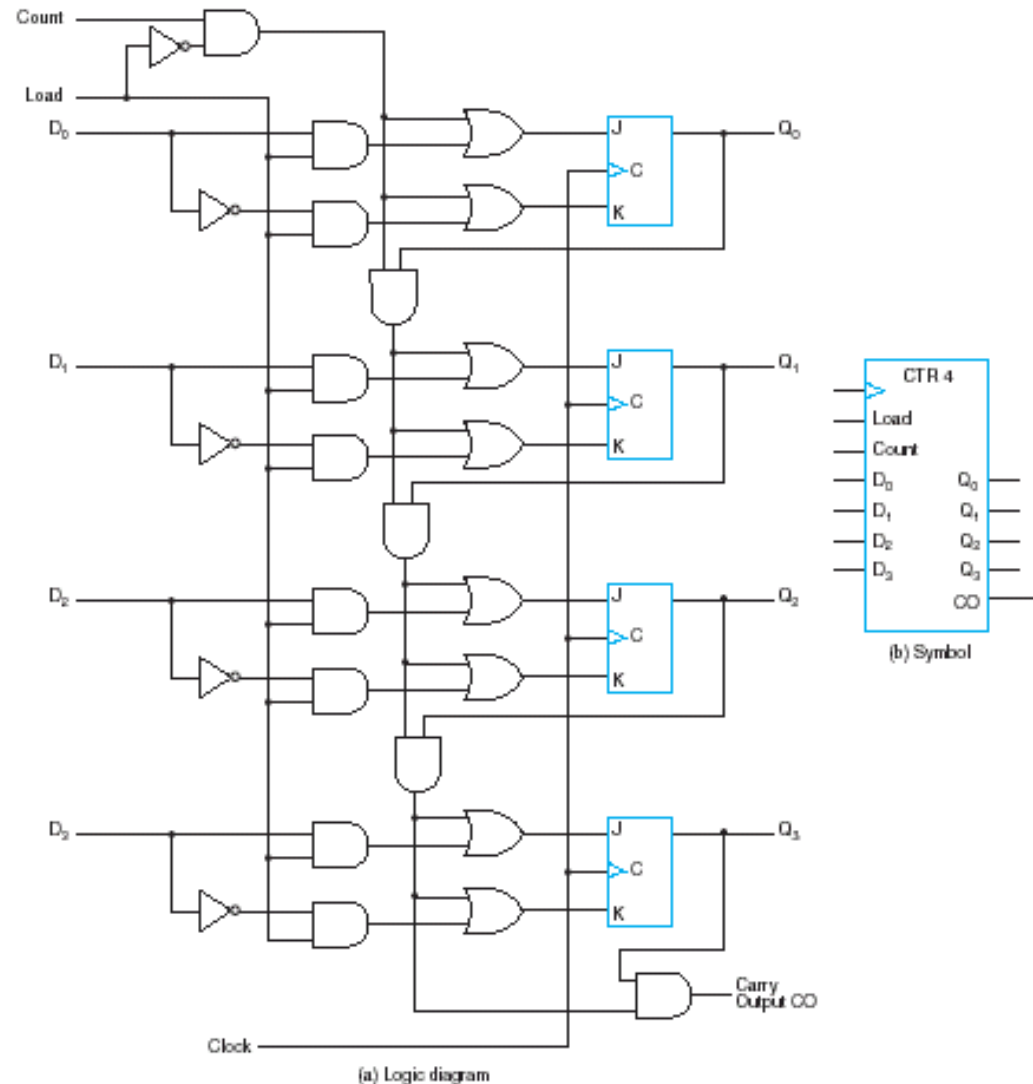


# 4-Bit Binary Counter with Parallel Load

Sometimes it is necessary to load the counter with an initial value at the startup.

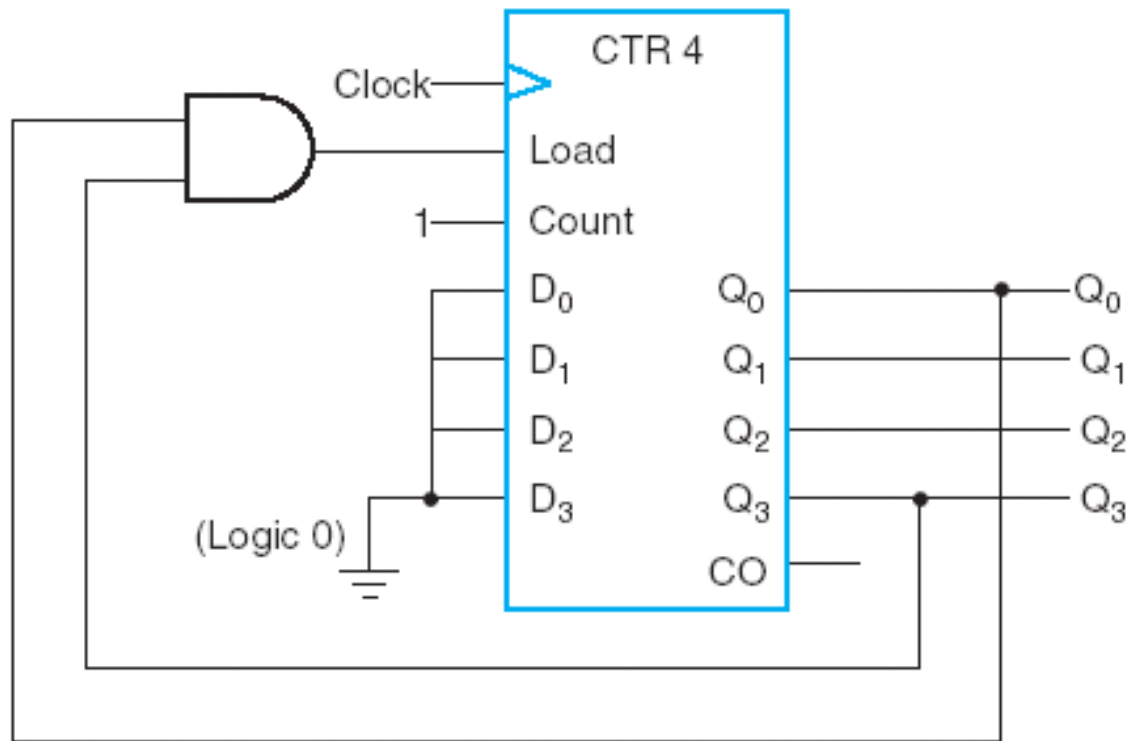
The initial value is other than zero.

The circuit shown can be loaded with an initial value and be used as a counter from then on.



# Other Counters

- **BCD Counter** – can be obtained from a binary counter with parallel load.



# Verilog Code for 4-bit Binary Counter with Reset

---

```
// 4-bit Binary Counter with Reset
// (See Figure 5-10)
module count_4_r_v (CLK, RESET, EN, Q, CO);
input CLK, RESET, EN;
output [3:0] Q;
output CO;
reg [3:0] Q;
assign CO = (count == 4'b1111 && EN == 1'b1) ? 1 : 0;
always@(posedge CLK or posedge RESET)
    begin
        if (RESET)
            Q <= 4'b0000;
        else if (EN)
            Q <= Q + 4'b0001;
    end
endmodule
```

# Verilog Code for 4-bit Shift Register with Reset

---

```
// 4-bit Shift Register with Reset
// (See Figure 5-3)
module srg_4_r_v (CLK, RESET, SI, Q,S0);
input CLK, RESET, SI;
output [3:0] Q;
output S0;
reg [3:0] Q;
assign S0 = Q[3];
always@(posedge CLK or posedge RESET)
    begin
        if (RESET)
            Q <= 4'b0000;
        else
            Q <= {Q[2:0], SI};
    end
endmodule
```