DLD CLASS TASK

22I-1552 Arham Ahmed

22I-1668 Muhammad Azan

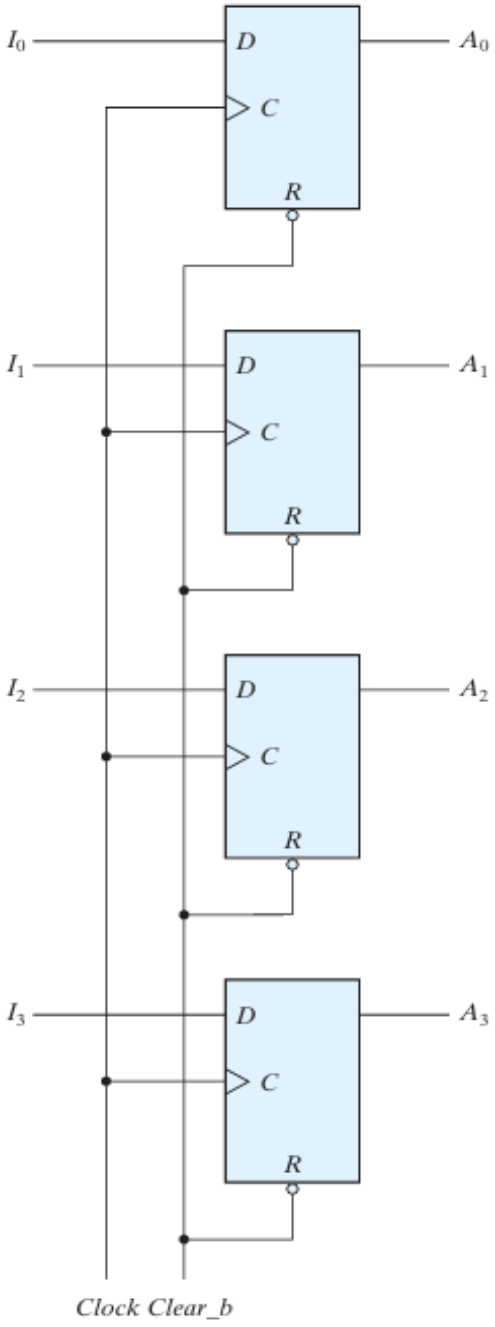


FIGURE 6.1
Four-bit register

A register that is built using four D-type flip-flops to create a four-bit data storage register is depicted in Figure 6.1. Every flip-flop is activated on the positive edge of every pulse by the common clock input, and the binary data that is available. The register receives the four inputs transferred there. (I3, I2, I1, and I0)'s value is instantly the value of (A3, A2, A1, and A0) after the clock edge is determined before the clock edge. The binary data kept in the register can be obtained at any moment by sampling any one of the four outputs. All four flip-flops have an active-low R (reset) input that receives the input Clear_b. When the input drops to zero, each flip-flop is asynchronously reset. Clear_b is a useful input to clear the register to all 0s before it is timed. During regular clocked operation, the R inputs must be kept at logic 1 (i.e., de-asserted). It is important to note that the register can be transferred to an all-0 state using either Clear, Clear_b, reset, or reset_b, depending on the flip-flop.

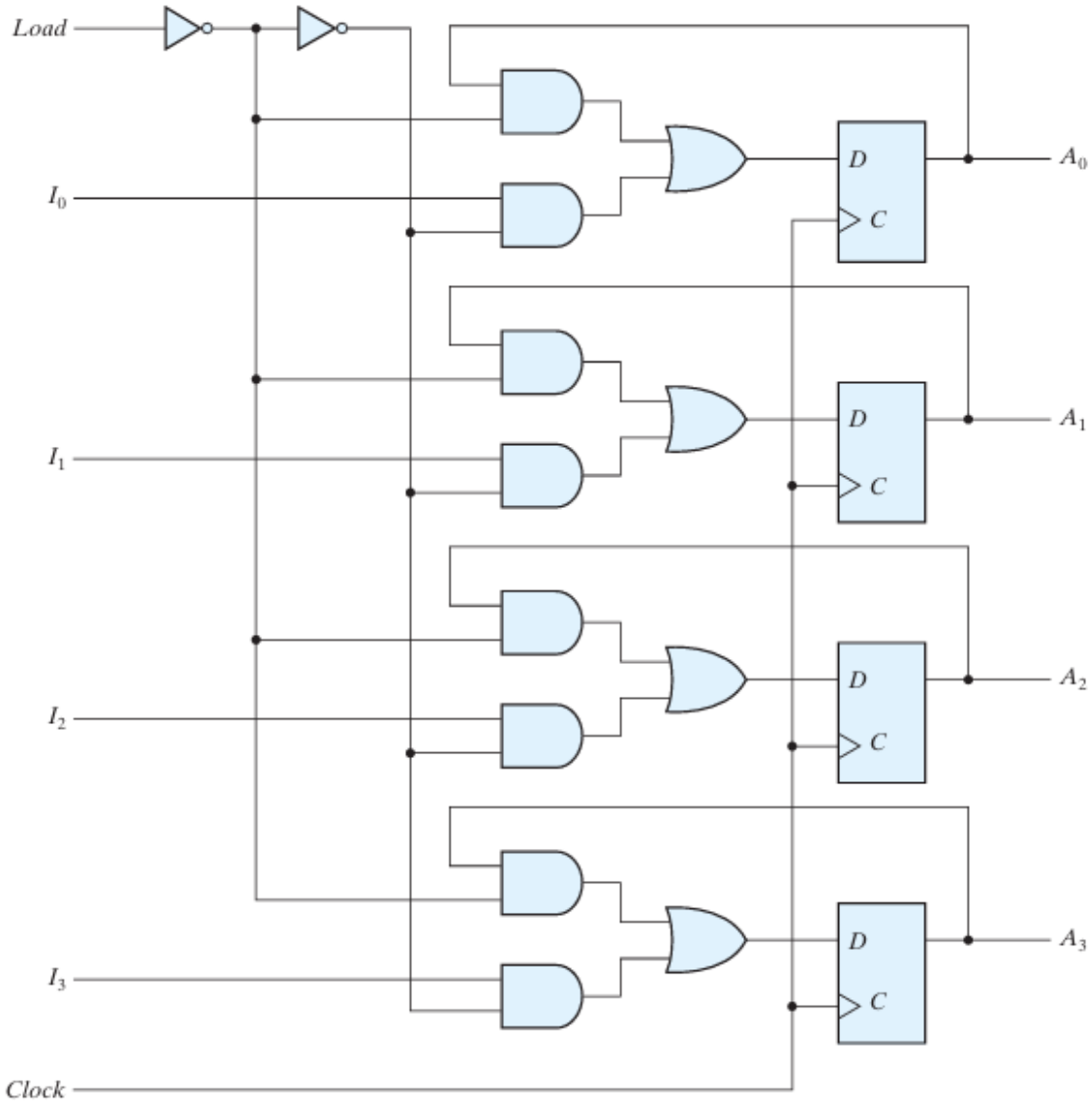


FIGURE 6.2
Four-bit register with parallel load

A four-bit data-storage register with gates directing the load control input. Figure 6.2 illustrates how the additional gates build a two-channel mux whose output drives the input to the register with either the data bus or the D inputs of the flip-flops, or the register's output. The action is determined by the load input to the register, captured each time a clock pulse. When the load input is 1, the data at the four external inputs are transferred into the register with the next positive edge of the clock.

The flip-flops' outputs are connected to their corresponding inputs when the load input is 0. The feedback link between the input and output is required in a D flip-flop since it lacks a "no change" condition. The D input dictates the register's subsequent state with each clock edge. The output must be made to circulate to the input at each clock pulse (i.e., the D input must match the output's present value to leave the output unchanged). The C inputs get uninterrupted clock pulse application. The next pulse's ability to accept new data or preserve the information already in the register is determined by the load input. Information is transferred from the data inputs, or the output of the register is done simultaneously with all four bits in response to a clock edge.

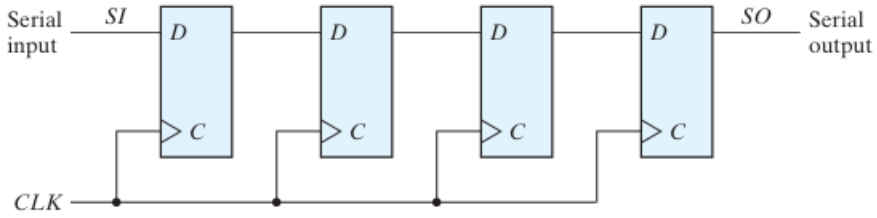
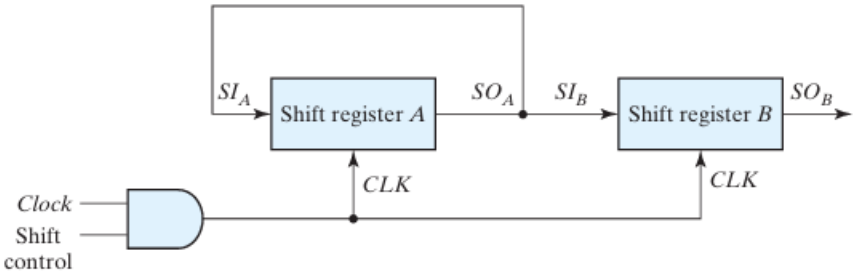
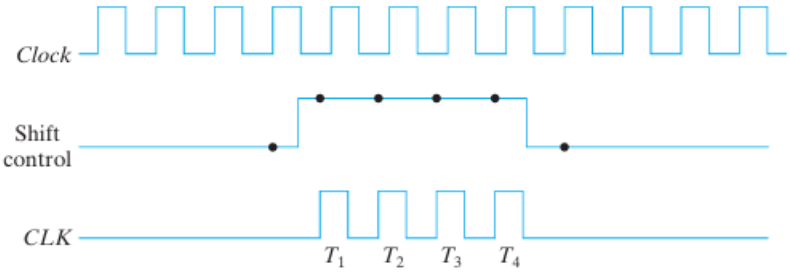


FIGURE 6.3
Four-bit shift register

As illustrated in Fig. 6.3, the most basic shift register is one that just makes use of flip-flops. A flip-flop's output is linked to the D input of the flip-flop on its right. It is a left-to-right shift register that is unidirectional. The register's contents are moved one bit to the right with each clock pulse. A shift to the left is not supported by this arrangement. What is put into the leftmost flip-flop during the shift is determined by the serial input. The output of the rightmost flip-flop is where the serial output is obtained from. Controlling the shift to happen just with specific pulses and not others is sometimes required. Similar to the data register covered in the part before, the clock's signal can be inhibited by gating the clock signal to prevent the register from shifting.



(a) Block diagram



(b) Timing diagram

FIGURE 6.4
Serial transfer from register *A* to register *B*

When data is transferred and altered bit by bit, a digital system's data path is said to be operating in serial mode. By moving the bits from the source register to the destination register, information is transferred one bit at a time. In contrast to this kind of transfer, parallel transfer transfers each bit in the register one at a time. Shift registers are used in the serial transfer of data from register A to register B, as the block diagram of Fig. 6.4 (a) illustrates. Register A's serial output (SO) and register B's serial input (SI) are linked. To guard against the data being lost from the source register, the information in register A is made to circulate by connecting the serial output to its serial input. The initial content of register B is shifted out through its serial output and is lost unless it is transferred to a third shift register. The shift control input determines when and how many times the registers are shifted. For illustration here, this is done with an AND gate that allows clock pulses to pass into the CLK terminals only when the shift control is active.

Table 6.1
Serial-Transfer Example

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

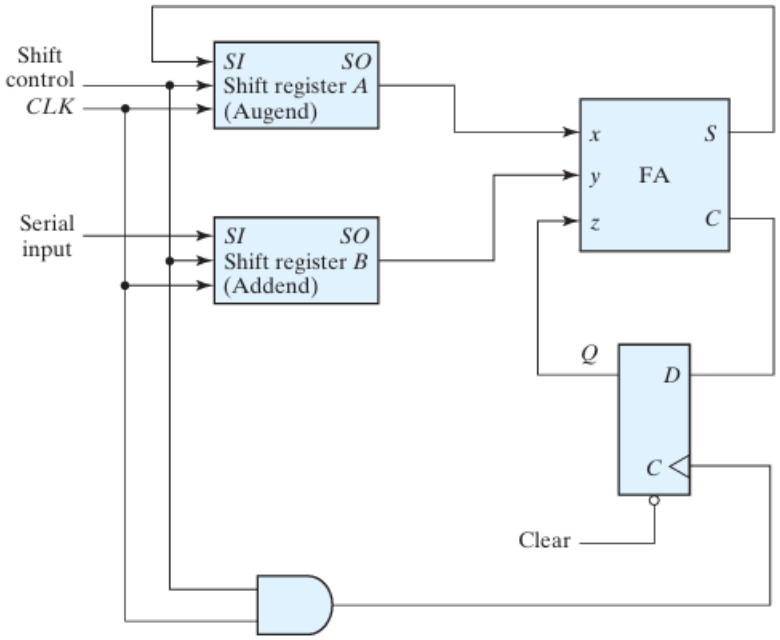


FIGURE 6.5
Serial adder

Two shift registers hold the two binary numbers that need to be added serially. As illustrated in Fig. 6.5, the circuit starts with the least significant pair of bits and adds one pair at a time using a single full-adder (FA) circuit. The complete adder's output is transferred to a D flip-flop, whose carry input is subsequently utilized for the subsequent pair of significant bits. It is possible to transfer the sum bit from the full adder's S output into a third shift register. One register can be used to store both the augend and the sum bits by shifting the sum into A while shifting the bits of A out. The serial input of register B can be used to transfer a new binary number while the addend bits are shifted out during the addition.

Table 6.2
State Table for Serial Adder

Present State	Inputs		Next State	Output	Flip-Flop Inputs	
Q	x	y	Q	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

$$J_Q = xy$$
$$K_Q = x'y' = (x + y)'$$
$$S = x \oplus y \oplus Q$$

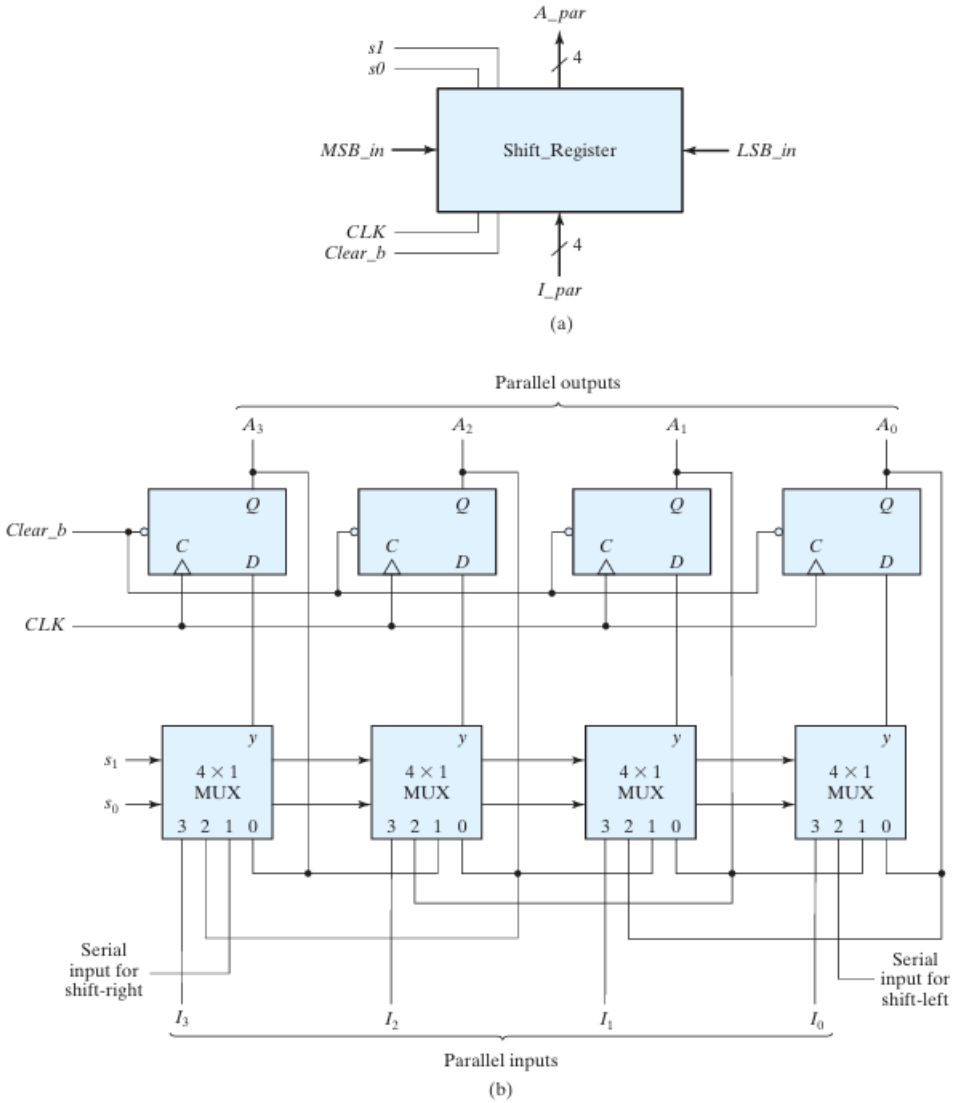


FIGURE 6.7
Four-bit universal shift register

An obvious way to reset the register to 0. a clock input to keep everything in sync. A shift-right control to activate the shift-right function and the shift-right-related serial input and output lines. A shift-left control to activate the shift-left function and the shift-left-related serial input and output lines. To facilitate a parallel transfer, a parallel-load control and the n input lines linked to it are required. n output lines running in parallel. a control state wherein, in response to the clock, the data in the register remains unchanged. With at least one shift operation, other shift registers might only have a few of the features.

Table 6.3
Function Table for the Register of Fig. 6.7

Mode Control		
s_1	s_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load