Universal Shift Register

Sparsh Mittal

Universal shift register

A register capable of shifting in one direction only is a *unidirectional* shift register.

One that can shift in both directions is a *bidirectional* shift register.

If the register has both shifts and parallel-load capabilities, it is referred to as a *universal shift register*

Universal shift register

The circuit consists of four *D* flip-flops and four multiplexers.

The four multiplexers have two common selection inputs $s1$ and $s0$.

Input 0 in each multiplexer is selected when $s1s0 = 00$, input 1 is selected when $s1s0 = 01$, and so on.

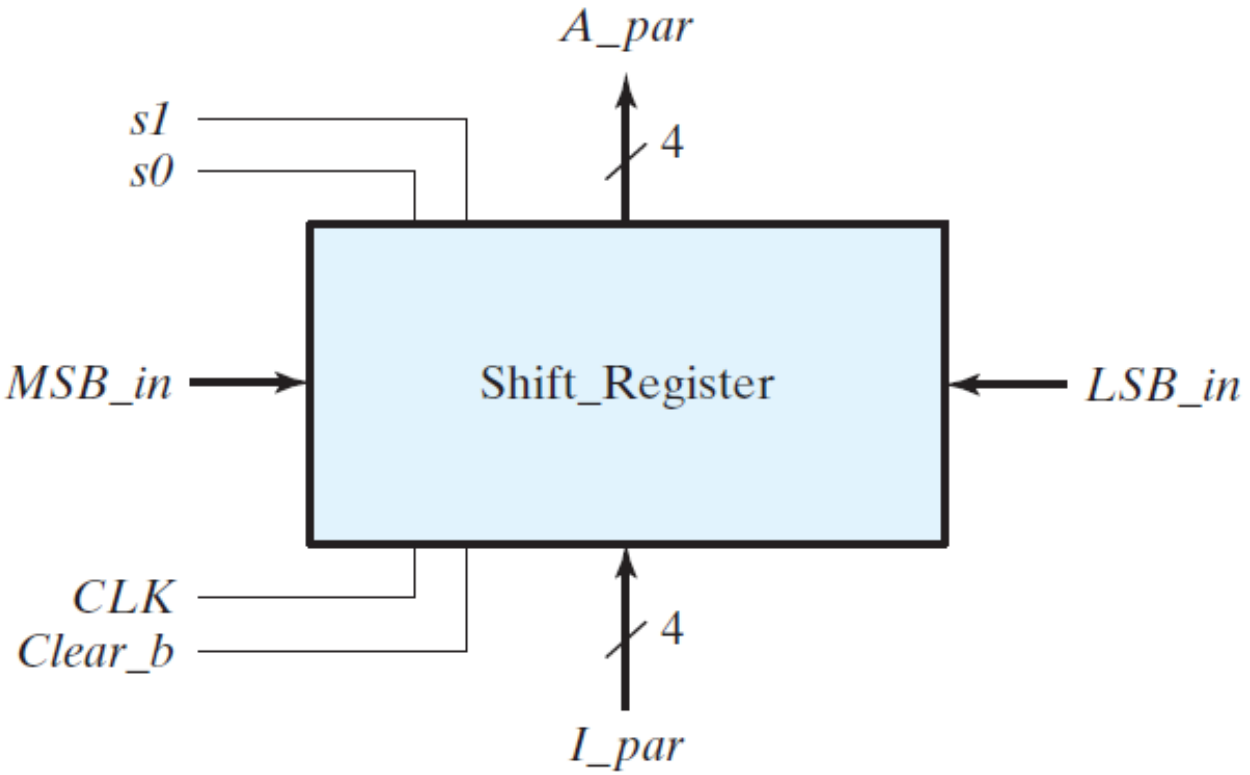
The selection inputs control the mode of operation of the register according to the function entries in this table.

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

Overall design

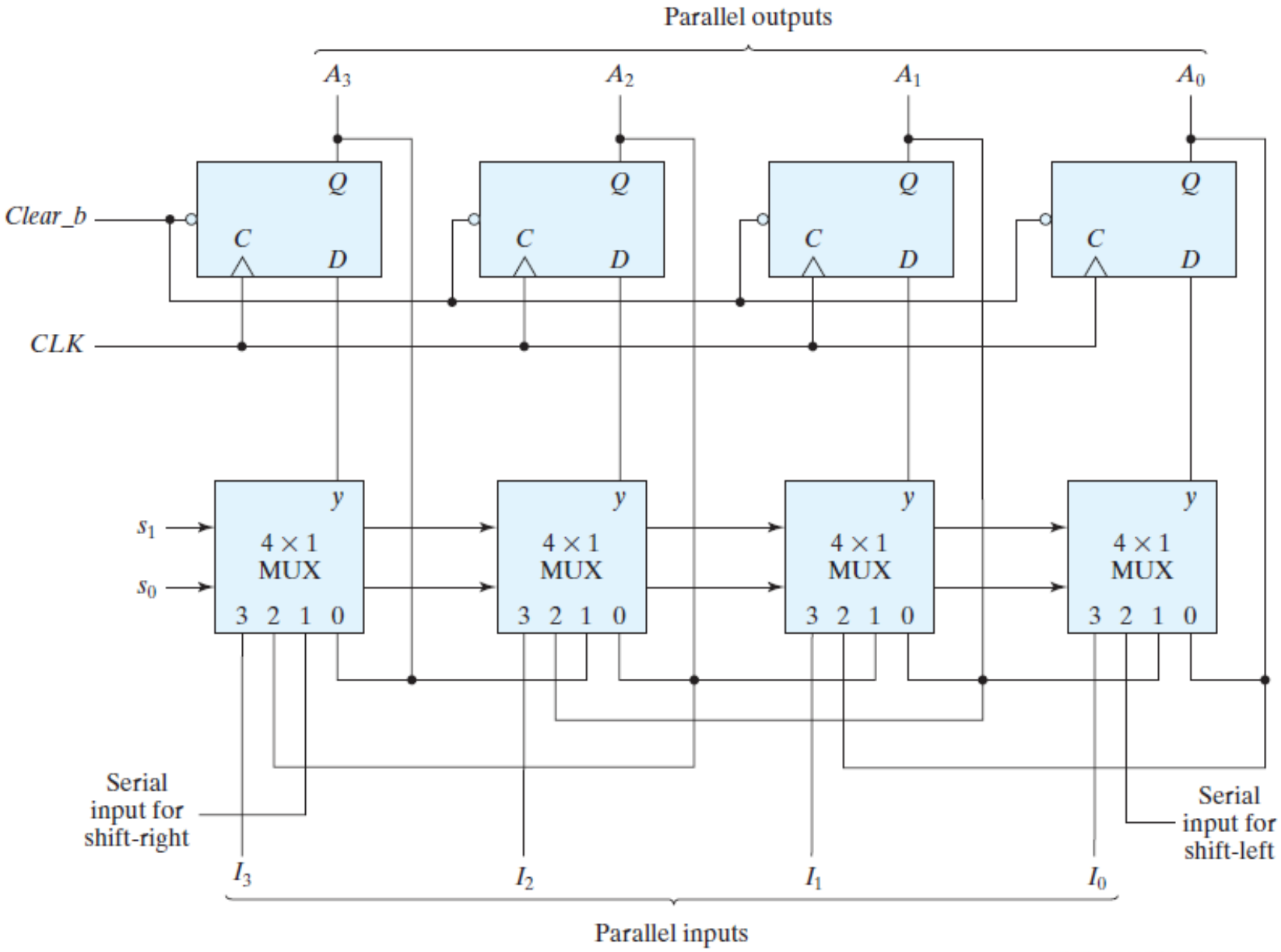
The four multiplexers have two common selection inputs $s1$ and $s0$.

Input 0 in each multiplexer is selected when $s1s0 = 00$, input 1 is selected when $s1s0 = 01$, and so on.



Low-level design

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



When $s1s0 = 00$

When $s1s0 = 00$, present value of the register is applied to the D inputs of the flip-flops.

This condition forms a path from the output of each flip-flop into the input of the same flip-flop, so that the output recirculates to the input in this mode of operation.

The next clock edge transfers into each flip-flop the binary value it held previously, and no change of state occurs.

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

When $s1s0 = 01$ and $=10$ and $=11$

When *s*1*s*0 = 01, terminal 1 of the multiplexer inputs has a path to the *D* inputs of the flip-flops. This causes a shift-right operation, with the serial input transferred into flip-flop *A*3.

When *s*1*s*0 = 10, a shift-left operation results, with the other serial input going into flip-flop *A*0.

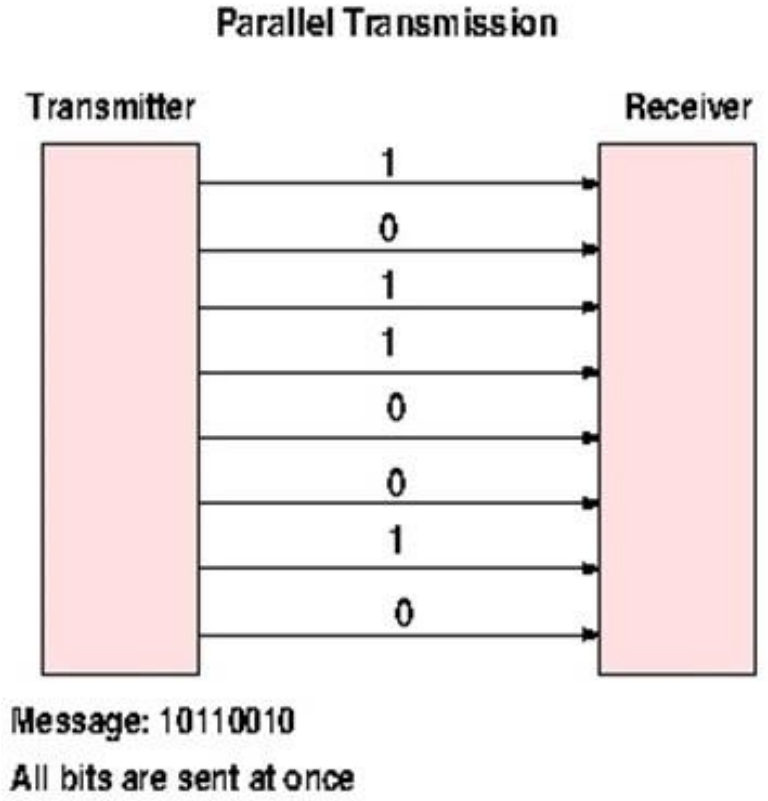
When *s*1*s*0 = 11, the binary information on the parallel input lines is transferred into the register simultaneously during the next clock edge.

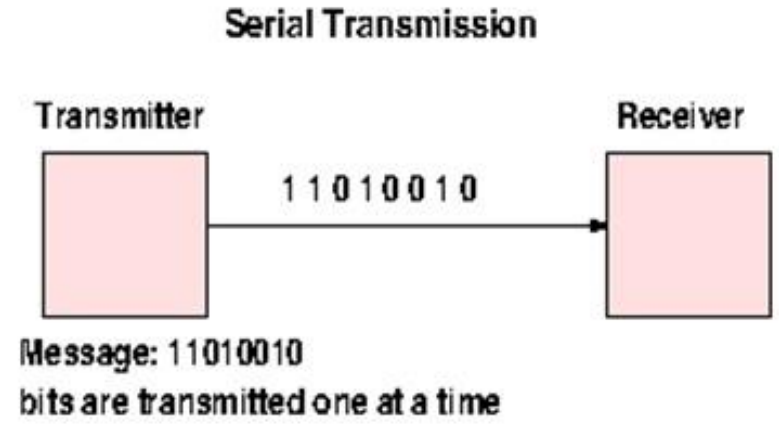
Note that data enters *MSB_in* for a shift-right operation and enters *LSB_in* for a shift-left operation.

Parallel vs Serial Transfer

To transmit an *n* -bit quantity between two far-away points, using n parallel lines is costly.

It is more economical to use a single line and transmit the information serially.





Usecase

Shift registers are used to interface digital systems situated remotely.

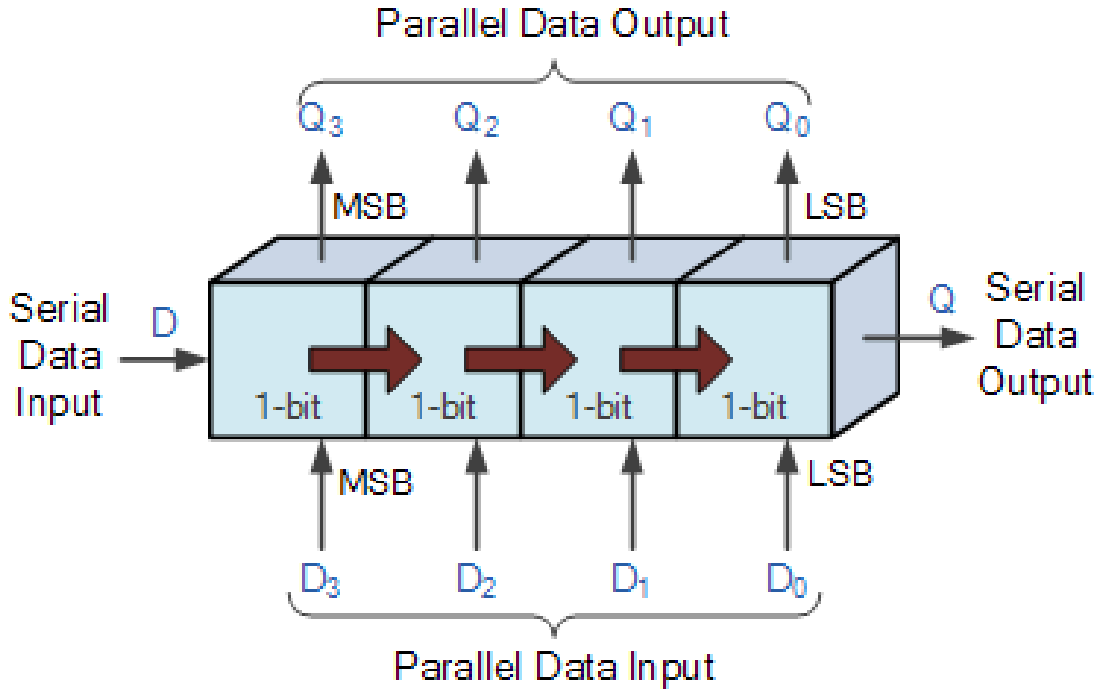
The transmitter accepts *n* -bit data in parallel into a shift register and then transmits the data serially along the common line.

The receiver accepts the data serially into a shift register.

When all *n* bits are received, they can be taken from the outputs of the register in parallel.

Thus, the transmitter performs a parallel-to-serial conversion of data and the receiver does a serial-to-parallel conversion.

Shift register for conversion between serial and parallel



References

https://www.globalspec.com/learnmore/semiconductors/logic/digital_parallel_serial_converters

https://www.electronics-tutorials.ws/sequential/seq_5.html