Design of Registers

October 19, 2020

Introduction

- A register consists of a group of flip-flops with a common clock input, used for storing binary information (data).
- Depending on the configuration, there can be several different variations.
  - Parallel-in-parallel-out (PIPO)
  - Serial-in-serial-out (SISO)
  - Parallel-in-serial-out (PISO)
  - Serial-in-parallel-out (SIPO)
- If a register supports either serial-in or serial-out or both modes, then the register is called a shift register.

Basic PIPO Register

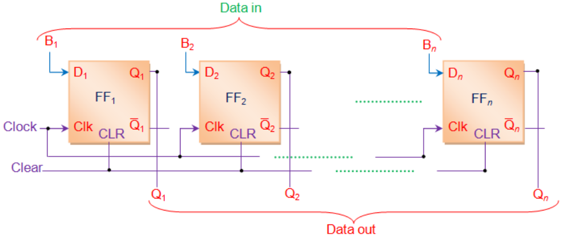


Figure 1 *n*-bit Parallel-In Parallel-Out Register

Figure: n-bit parallel-in-parallel-out register.

- Parallel inputs are fed to the D-inputs of the flip-flops.
- When clear = 1, the outputs of all flip-flops are at zero.
- Clock input is fed in parallel to all flip-flops.
- When the active clock edge comes (positive or negative edge), the available data $(B_1, B_2, \ldots B_n)$ in the inputs is stored in the register and also available at its outputs.

Addition of LOAD Signal with Basic PIPO

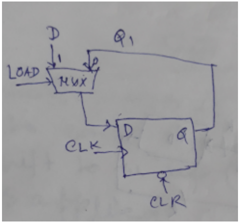
- In practice, the clock is coming continuously and there is a separate signal LOAD that specifies when the register is to be loaded with new data.
- There are two possible solutions:

  (a) Use a gated clock: Not a good solution, as gating the clock with another signal can cause timing problem. Load has to come before clock is arrived.

  (b) Separate out clock and LOAD using multiplexer circuit.

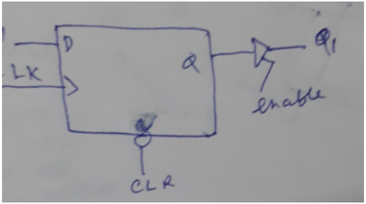
  - Better and recommended solution.

  - Each flip-flop in the register gets replaced by



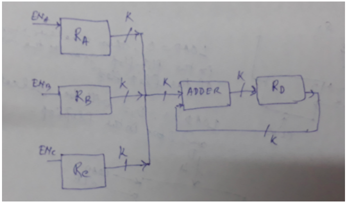
Addition of Enable input with the PIPO register

- Many registers have an enable input that can be used to force the output lines into high impedance state, if required.
- We need tri-state buffer with every flip-flop output. When enable $= 1$, $Q_1 = Q$ and enable $= 0$, the output is at high impedance state.



- Why enable input is required?
  To resolve bus conflicts in bus-based architectures.

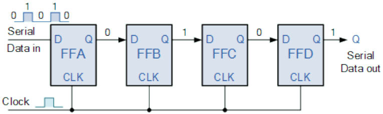
An Example Scenario



- R_A, R_B, R_C, and R_D are k-bit registers.
- Exactly one of the three enable inputs must be activated at any given time.
- Clock is fed in parallel to all the registers.
- If $EN_A = 1$, then $R_D = R_D + R_A$
- If $EN_B = 1$, then $R_D = R_D + R_B$
- If $EN_C = 1$, then $R_D = R_D + R_C$

Shift Register

- A shift register is register in which the binary data can be stored and the data can be shifted to the left (or right) when a shift signal is applied.
- It can be constructed by connecting D, SR or JK filp-flops in cascade.
- A 4-bit shift register is shown below.



- The first FF receives data from external world. The output of the 1st FF is connected to the input of the 2nd FF and so on. We get the serial output from the 4th

Shift Register

- A sample timing diagram is shown below assuming that the initial state is $Q_1 Q_2 Q_3 Q_4 = 0102$

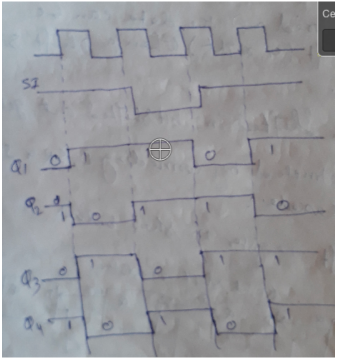


Figure: Timing Diagram

Different Types of Flip-flop

Depending upon the ways the various storage devices are connected, there can different types of shift-register:

(a) Ring counter

(b) Twisted ring or Johnson counter

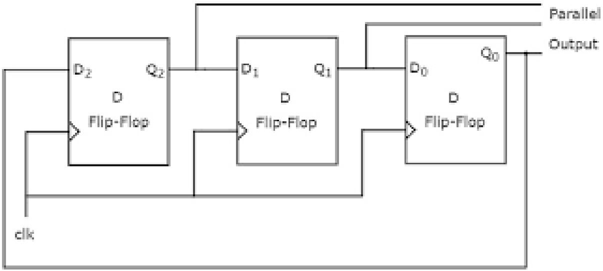
(c) Bidirectional shift register

(d) Universal shift register

(e) Liner feedback shift register

Ring Counter

- This is obtained from SISO shift register by connecting the Q output of the last FF to the D input of the first FF.
- Typically a ring counter is initially with a single 1 and all FFs remain 0's
- This can generate multi-phase clock.
- For a k-bit ring counter, the contents of the register gets repeated after k clocks



Timing Diagram of a 3-bit Ring Counter

A 3 bit ring counter assumes $Q_1 Q_2 Q_3 = 100$ the initial state.

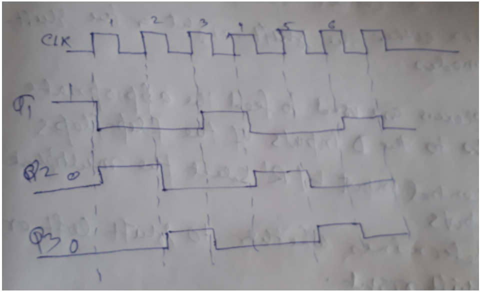
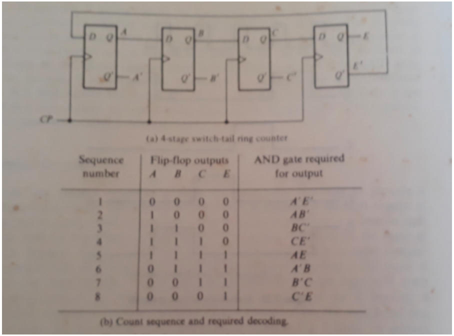


Figure: Timing diagram of a 3 bit ring counter.

Johnson Counter

1. This is obtained from a SISO shift register by connecting the $\overline{Q}$ output of the last flip-flop to the D input of the first flip-flop.
2. The Johnson counter can be initialized to the all 0 state.
3. For a k-bit Johnson counter, contents of the register gets repeated after 2k clocks.

Johnson Counter


D
Q
A
Q'
A'
D
Q
B
Q'
B'
D
Q
C
Q'
C'
D
Q
E
Q'
E'
CP

(a) 4-stage switch-tail ring counter

Sequence number	Flip-flop outputs				AND gate required for output
	A	B	C	E	
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	$C'E$

(b) Count sequence and required decoding.

Bidirectional Shift register

- Base on the control signal, the shift register works in either the shift-right mode or shift-left mode.
- Multiplexers are used to feed the appropriate signals to the D inputs of the flip-flops.
- A control signal $L/\overline{R}$ select the multiplexer inputs. Determine whether to shift left or shift right.
  $L/\overline{R} = 1 \Rightarrow$ left shift
  $L/\overline{R} = 0 \Rightarrow$ right shift
- $L/\overline{R} = 0$, then 0 input of the MUX will be selected. SI will come to D_1, Q_1 will come to D_2 and so on.
- $L/\overline{R} = 1$, SI will come to D_4, Q_4 will come to D_3 and so on.

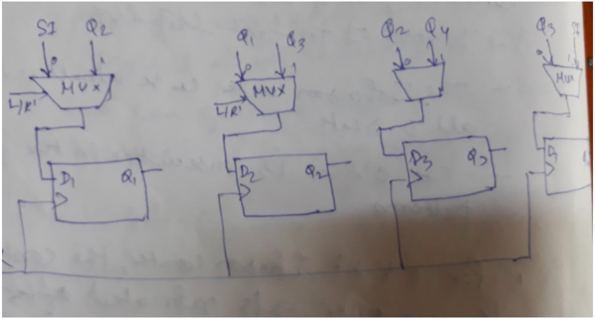


Figure: Bidirectional Shift Register.

Universal Shift Register

A Universal shift register is a bidirectional shift register, whose input can be either in serial or in parallel and whose output also be either in serial or in parallel.

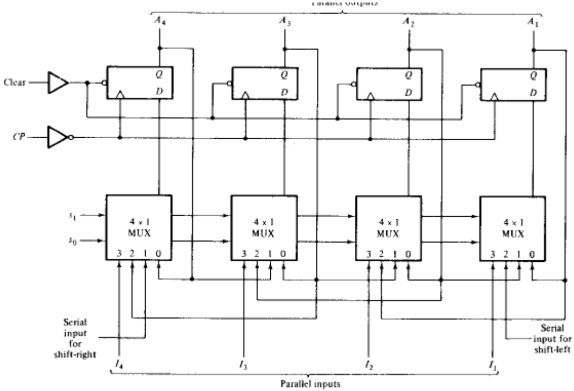


Figure: Universal Shift Register.

Table: Function Table for the Universal Register.

Mode control		Operation
s_1	s_0	
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel loading

Linear feedback shift register (LFSR)

- It is basically a SISO right shift register where the serial input is generated as a linear combination of some of the FF outputs.

Table: Transition sequences of the LFSR.

CLK→	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Q_1	0	1	1	1	1	0	1	0	1	1	0	0	1	0
Q_2	0	0	1	1	1	1	0	1	0	1	1	0	0	1
Q_3	0	0	0	1	1	1	1	0	1	0	1	1	0	0
Q_4	1	0	0	0	1	1	1	1	0	1	0	1	1	0

- The circuit is used to generate random patterns.
- The LFSR is typically initialized to 1 0 0 0.

Excitation and Output Table

- The circuit diagram of LFSR is given below:

bit states except for (0000).

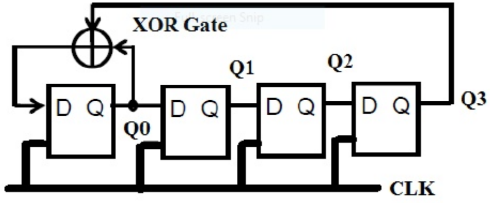


Figure: Linear Feedback Shift Register.