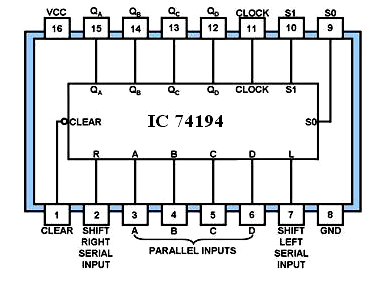
**Experiment No. 9**

**Sequence Detector and Shift Register**

**Hardware runs**

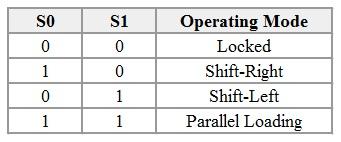
**Components Required**



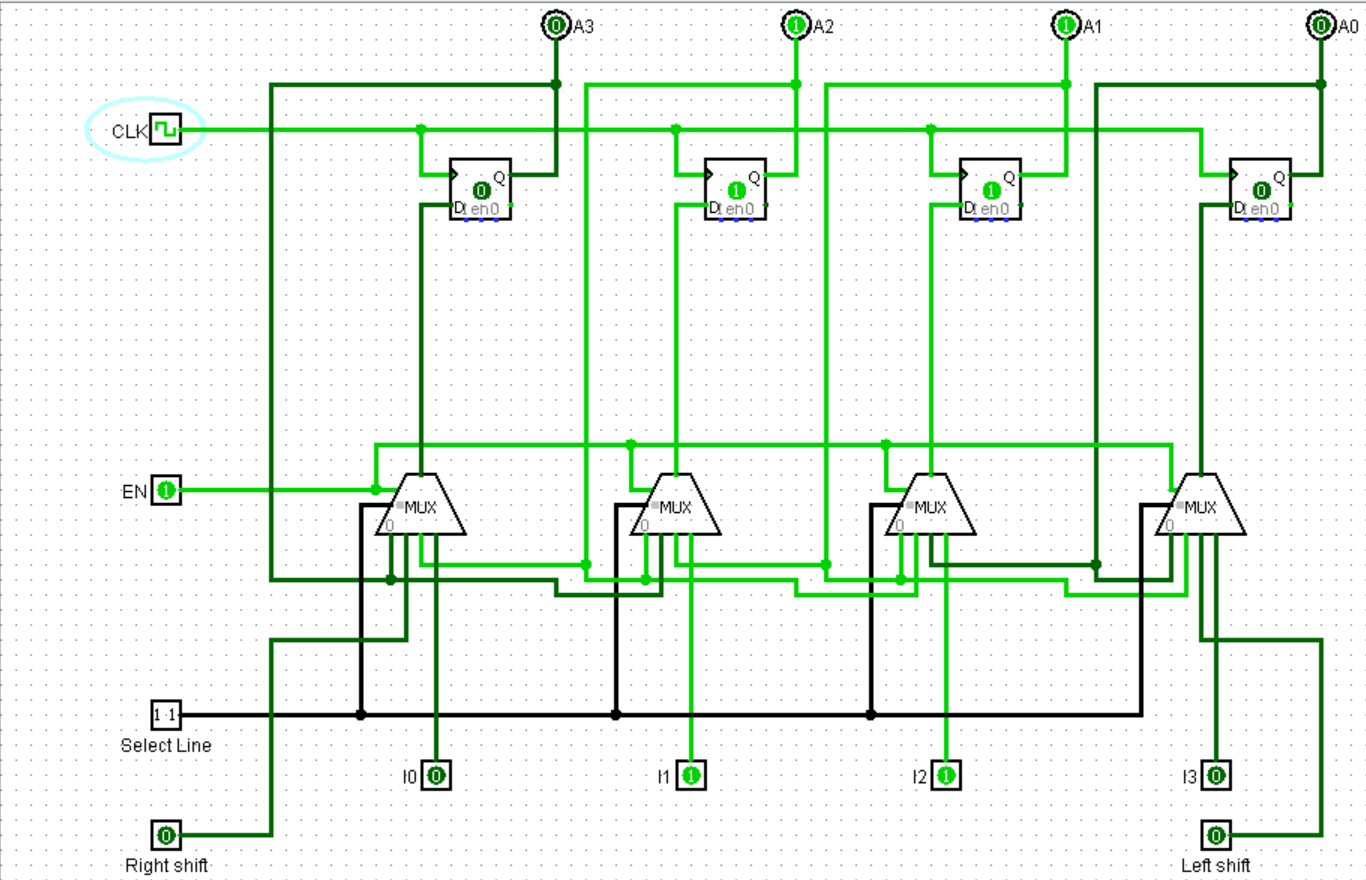
**IC 74194 Universal shift register**

**Shift registers**

A universal register is a general-purpose register capable of performing three operations: shift-right, shift-left, and parallel load.



**Run 1: Parallel-in parallel-out (20 mins)**

****

**Truth Table**

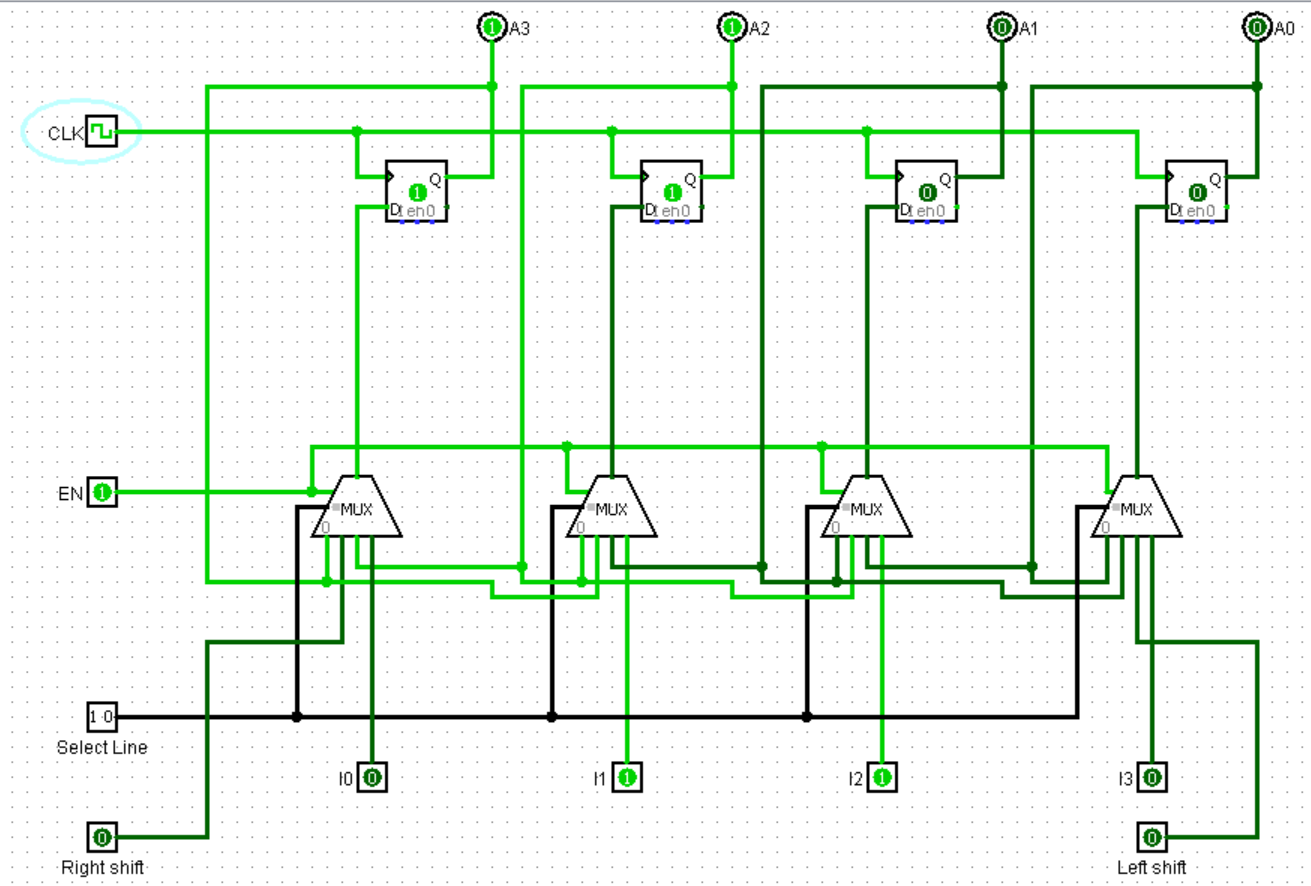
**S0=1, S1=1,**

|  |  |  |
| --- | --- | --- |
| **Input** | **Clk** | **Output** |
| **A B C D** | **QAQBQCQD** |
| 0000 | 1 | 0000 |
| 1010 | 1 | 1010 |
| 1111 | 1 | 1111 |

**Run 2: Serial-in Parallel-out (Left Shift) (15 mins)**

**First load the ‘0000’ to the O/P, with help of Parallel Load.**

**The following circuit is a universal Shift Register. It can shift in parallel and serial in all combinations i.e. parallel shift, left shift and right shift.**

****

**Truth Table**

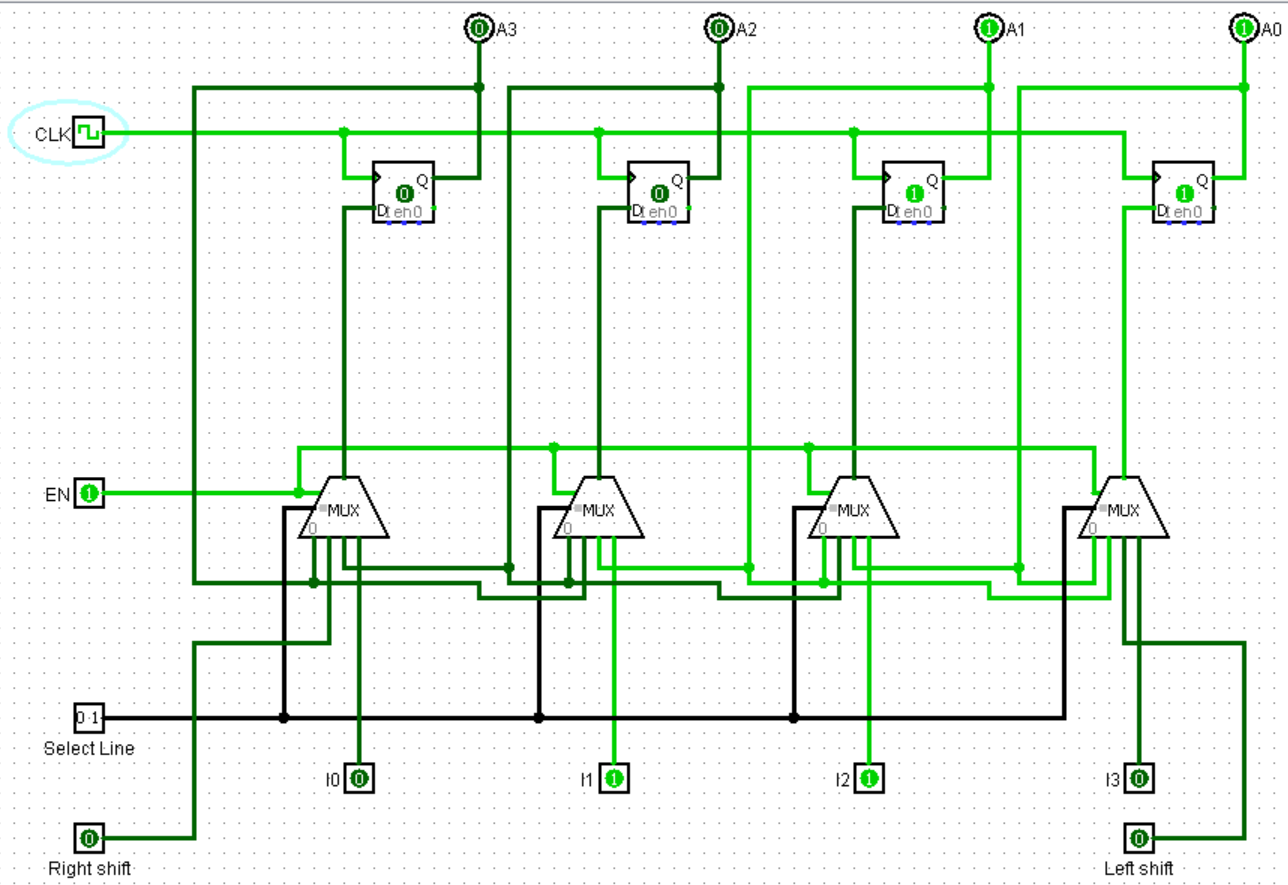
**S0=1, S1=0,**

|  |  |  |
| --- | --- | --- |
| **Shift Right (I/P)**  **Pin No.-2** | **No. of pulses(n)** | **QAQBQCQD** |
| 1 | 1 | 0001 |
| 1 | 2 | 0011 |
| 1 | 3 | 0111 |
| 1 | 4 | 1111 |
| 0 | 5 | 1110 |
| 0 | 6 | 1100 |
| 0 | 7 | 1000 |
| 0 | 8 | 0000 |
| 1 | 9 | 0001 |

**Run 3: Serial-in Parallel-out (Right Shift) (10**

**mins)**

**First load the ‘0000’ to the O/P, with help of Parallel Load.**

****

**Truth Table**

**S0=0, S1=1,**

|  |  |  |
| --- | --- | --- |
| **Shift Left (I/P)**  **Pin No.-7** | **No. of pulses(n)** | **QAQBQCQD** |
| 1 | 1 | 1000 |
| 1 | 2 | 1100 |
| 1 | 3 | 1110 |
| 1 | 4 | 1111 |
| 0 | 5 | 0111 |
| 0 | 6 | 0011 |
| 0 | 7 | 0001 |
| 0 | 8 | 0000 |
| 1 | 9 | 1000 |

**Software runs**

**Run 4: (25 mins)**

**Sequence detector for ‘110’:** Write the Verilog code and testbench for the sequence detector as Moore machine.

<https://www.edaplayground.com/x/PMaF>

**0**

**1**



**S0/0**

**S1/0**



**0**



**1**

**0**

**1**

**S2/0**

**S3/1**



**0**

**1**

*module detector\_110 (input x, clk, rst, output reg z);*

*parameter [1:0] S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;*

*reg [1:0] present\_state;*

*always @ (posedge clk, posedge rst) begin*

*z <= (present\_state == S3) ? 1 : 0;*

*if (rst) begin*

*z <= 1'b0;*

*present\_state <= S0;*

*end*

*else begin*

*case (present\_state)*

*S0: begin*

*if (x) present\_state <= #1 S1;*

*else present\_state <= #1 S0;*

*end*

*S1: begin*

*if (x) present\_state <= #1 S2;*

*else present\_state <= #1 S0;*

*end*

*S2: begin*

*if (x) present\_state <= #1 S2;*

*else present\_state <= #1 S3;*

*end*

*S3: begin*

*if (x) present\_state <= #1 S1;*

*else present\_state <= #1 S0;*

*end*

*endcase*

*end // end of else begin*

*z<= (present\_state == S3) ? 1 : 0;*

*end // end of always begin*

*endmodule*

*module t\_exp9a;*

*wire z;*

*reg x,clk, rst;*

*initial #200 $finish;*

*initial begin clk = 0; forever #5 clk = ~clk; end*

*initial begin*

*x = 0; rst = 1;*

*#10 x=1;*

*#10 x= 0; rst =0;*

*#10 x= 1;*

*#10 x= 1;*

*#10 x= 0;*

*#10 x= 1;*

*#10 x= 0;*

*#10 x= 1;*

*#10 x= 0;*

*#10 x= 1;*

*#10 x= 1;*

*#10 x= 0;*

*#10 x= 0;*

*#10 x= 0;*

*#10 x= 1;*

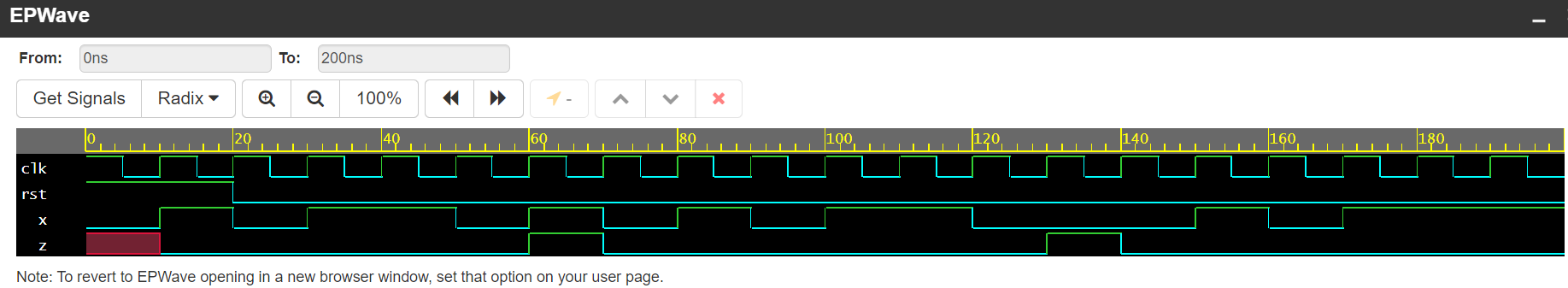
*#10 x= 0;*

*#10 x= 1;*

*end*

*module detector\_110 u(x,clk,rst,z);*

*endmodule*



**Run 5: (25 mins)**

4-bit universal shift register: Write Verilog code and testbench for 4-bit universal shift register. Code is given below for your reference.

<https://www.edaplayground.com/x/uVhT>

*module uni\_shift\_reg (input shift\_L, shift\_R, clk, input [3:0] parallel\_in, input [1:0] mode, output reg [3:0]*

*parallel\_out);*

*always @ (negedge clk) begin*

*case (mode)*

*0: parallel\_out <= parallel\_out; // for mode 0 data retain as it is*

*1: parallel\_out <= {parallel\_out [2:0], shift\_L}; // for mode 1 shift left*

*2: parallel\_out <= {shift\_R, parallel\_out [3:1] }; // for mode 2 shift right*

*3: parallel\_out <= parallel\_in; // for mode 3 parallel data in*

*default: parallel\_out <= 4'bx;*

*endcase*

*end*

*endmodule*

// Test Bench

*module tb;*

*reg shift\_L, shift\_R, clk;*

*reg [1:0] mode;*

*reg [3:0] parallel\_in;*

*wire [3:0] parallel\_out;*

*initial*

*begin*

*clk = 1'b0;*

*repeat (300) # 5 clk = ~clk; // clock is of time period 10 and 50% duty cycle.*

*end*

*always begin*

*#00 mode = 2'b11; parallel\_in=4'b0000; shift\_L=1; shift\_R=1;*

*#40 mode =2'b01; shift\_L=1;shift\_R=0;parallel\_in =4'b0000;*

*#40 mode =2'b11; parallel\_in=4'b0000;*

*#40 mode =2'b10; shift\_R=1;shift\_L=0;*

*#40 mode =2'b10; shift\_R=0;shift\_L=1;*

*#40 mode =2'b11; parallel\_in =4'b1010;*

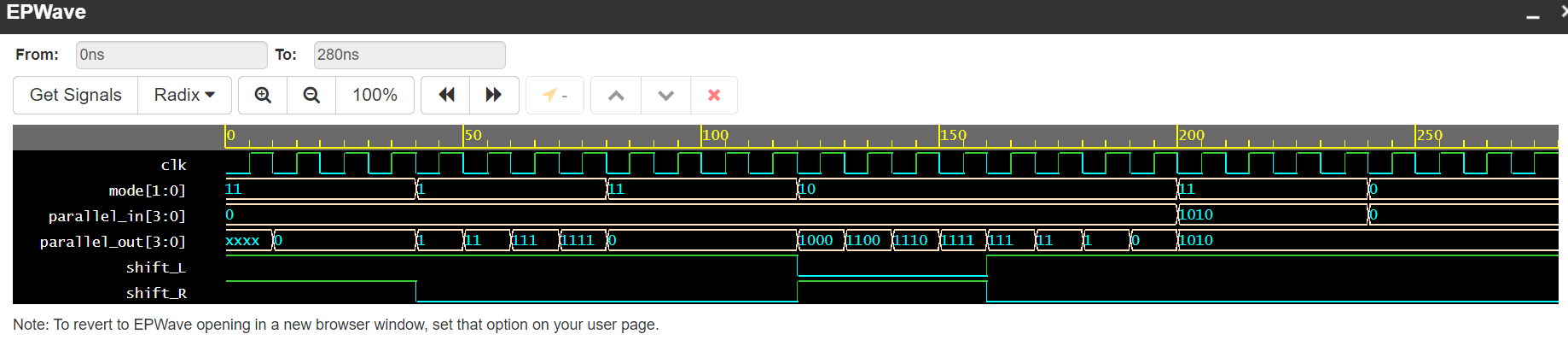
*#40 mode =2'b00; parallel\_in =4'b0000;*

*#40 $stop;*

*end*

*uni\_shift\_reg u(shift\_L, shift\_R, clk, parallel\_in,mode, parallel\_out);*

*endmodule*

**