ShanghaiTech University

EE 115B: Digital Circuits

Fall 2022

Lab 3: Universal Shift Register

Total: 100 Points

Assigned: December 6, 2022. Due: December 18, 2022.

In this lab, you will design a 4-bit universal shift register that implements four operation modes: no change (i.e., hold), right shift, left shift, and parallel load. The block diagram of the register is shown as follows:

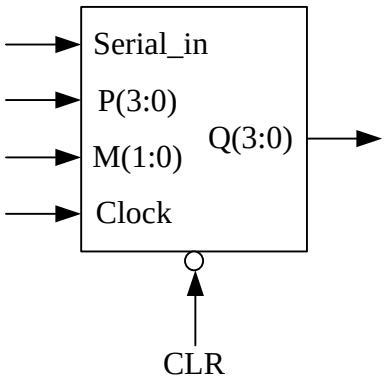

Figure 1: Block diagram of universal shift register.

The signals are defined as follows:

M(1:0)	Mode selection signal
Serial_in	Serial input
P(3:0)	Parallel input
Clock	Clock signal: positive edge active
CLR	Asynchronous clear signal: active low
Q(3:0)	Output

The four operation modes are defined as follows:

M(1)	M(0)	Mode
0	0	Right shift: Serial_in to Q(3), Q(3) to Q(2), Q(2) to Q(1), Q(1) to Q(0)
0	1	Left shift: Serial_in to Q(0), Q(0) to Q(1), Q(1) to Q(2), Q(2) to Q(3)
1	0	Parallel load: P(3) to Q(3), P(2) to Q(2), P(1) to Q(1), P(0) to Q(0)
1	1	No change

This register can be constructed using four D flip-flops and four 4-to-1 multiplexers. The schematic is shown as follows:

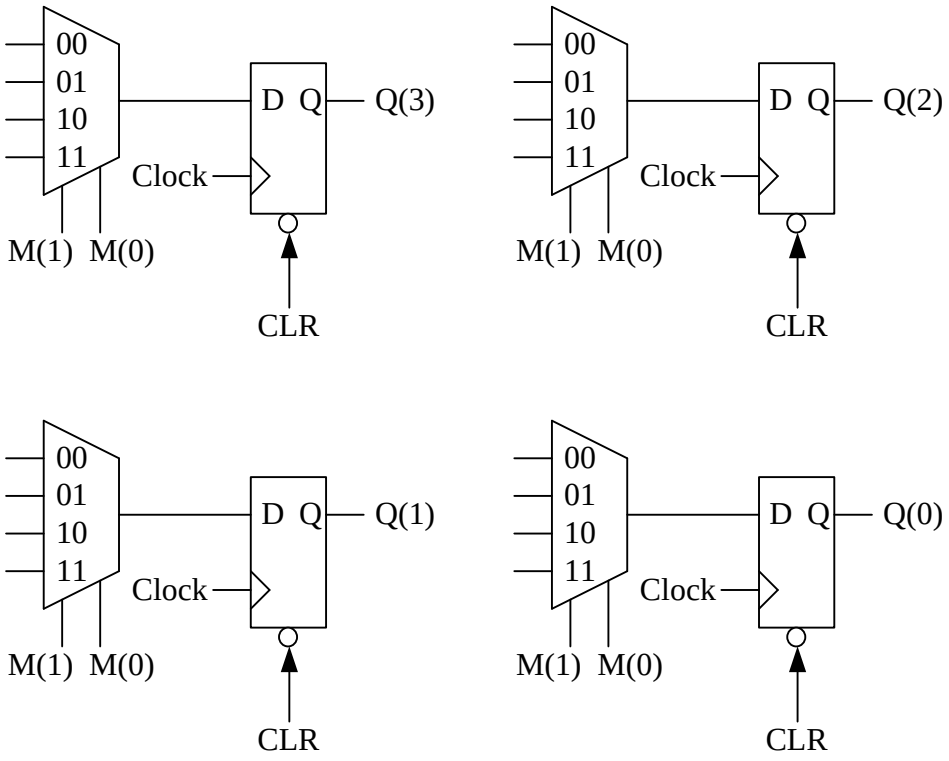

Figure 2: Schematic of universal shift register.

To implement this register, you need to complete the following steps:

(1) Complete the schematic shown in Figure 2 by labeling the **actual inputs** for each multiplexer,
(2) Write the **behavioral** code for two components: D flip-flop and 4-to-1 multiplexer,
(3) Write the **structural** code for the register,
(4) Test the register by simulating it using a testbench. A sample testbench is provided in this document, which tests the asynchronous clear signal (i.e., CLR) and two modes: parallel load and right shift. You need to modify it to test the other two modes: no change and left shift.

Please use the template to prepare your report. You need to include the following items:

(1) Schematic showing the appropriate signals at the multiplexer inputs,
(2) Code for the components: D flip-flop and 4-to-1 multiplexer,
(3) Code for the register,
(4) Code for the testbench,
(5) Simulation timing diagram verifying the four operation modes.

Sample testbench code

```
ENTITY shiftreg_test IS
END shiftreg_test;

ARCHITECTURE behavior OF shiftreg_test IS
        -- Component Declaration for the Unit Under Test (UUT)
        COMPONENT shiftreg
        PORT(
                P : IN std_logic_vector(3 downto 0);
                Q : OUT std_logic_vector(3 downto 0);
                clock : IN std_logic;
                clr : IN std_logic;
                serial_in : IN std_logic;
                m : IN std_logic_vector(1 downto 0)
        );
END COMPONENT;

        --Inputs
        signal P : std_logic_vector(3 downto 0) := (others => '0');
        signal clock : std_logic := '0';
        signal clr : std_logic := '0';
        signal serial_in : std_logic := '0';
        signal m : std_logic_vector(1 downto 0) := (others => '0');

        --Outputs
        signal Q : std_logic_vector(3 downto 0);
```

```
-- Clock period definitions
constant clock_period : time := 20ns;

BEGIN
	-- Instantiate the Unit Under Test (UUT)
	uut: shiftreg PORT MAP (
		P => P,
		Q => Q,
		clock => clock,
		clr => clr,
		serial_in => serial_in,
		m => m
	);

-- Clock process definitions
clock_process :process
begin
	clock <= '0';
	wait for clock_period/2;
	clock <= '1';
	wait for clock_period/2;
end process;

-- Stimulus process
stim_proc: process
begin
	-- test CLR
	clr<='0';
```

```
        wait for 30ns;

        -- test parallel load
        clr<='1';
        m<="11";
        P<="0110";
        wait for 40ns;

        -- test right shift
        m<="01";
        serial_in<='1';
        wait for 80ns; -- shift 4 bits

        wait;
end process;

END;
```