Computer Architecture Lab Session 3

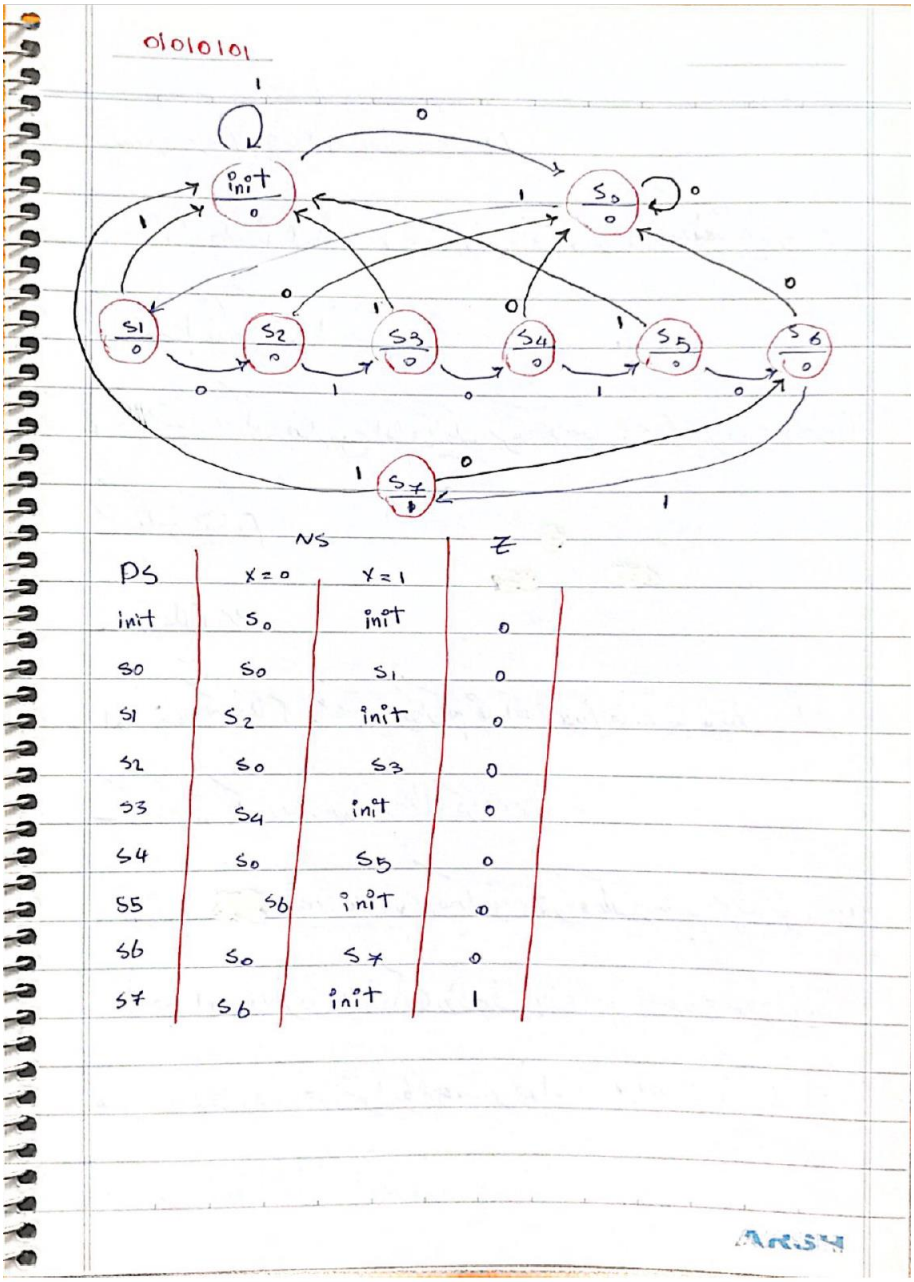
01010101 Sequence Detector

Bardia Ardakanian 9831072

Ali Asad 9831004

طراحی یک Sequence Detector با ماشین مور

برای طراحی یک ماشین مور stateها را می‌نویسیم و خروجی‌های هر استیت را هم مشخص می‌کنیم، در نهایت با توجه به ورودی ارتباط بین stateها را مشخص می‌کنیم.



پیاده سازی یک Sequence Detector با زبان vhdl

برای نوشتن یک Sequence Detector ، ابتدا موجودیت Detector را تعریف میکنیم و سپس پورت های ورودی خروجی را تعریف م یکنیم، در نهایت در ساختمان رفتاری Detector را تعریف م یکنیم.

```
-- VHDL project: VHDL code for Sequence Detector using Moore FSM
-- The sequence being detected is "01010101"
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY VHDL_MOORE_FSM_Sequence_Detector IS
    PORT (
        clock : IN STD_LOGIC; --- clock signal
        reset : IN STD_LOGIC; -- reset input
        sequence_in : IN STD_LOGIC; -- binary sequence input
        detector_out : OUT STD_LOGIC -- output of the VHDL sequence
detector
    );
END VHDL_MOORE_FSM_Sequence_Detector;

ARCHITECTURE Behavioral OF VHDL_MOORE_FSM_Sequence_Detector IS
    TYPE MOORE_FSM IS (Init, S0, S1, S2, S3, S4, S5, S6, S7);
    SIGNAL current_state, next_state : MOORE_FSM;

BEGIN
    -- Sequential memory of the VHDL MOORE FSM Sequence Detector
    PROCESS (clock, reset)
    BEGIN
        IF (reset = '1') THEN
            current_state <= Init;
        ELSIF (rising_edge(clock)) THEN
            current_state <= next_state;
        END IF;
    END PROCESS;

    -- Next state logic of the VHDL MOORE FSM Sequence Detector
    -- Combinational logic
    PROCESS (current_state, sequence_in)
    BEGIN
        CASE(current_state) IS
            WHEN Init =>
                IF (sequence_in = '0') THEN
                    -- "0"
                    next_state <= S0;
```

```
            END IF;
        WHEN S0 =>
            IF (sequence_in = '1') THEN
                -- "01"
                next_state <= S1;
            END IF;
        WHEN S1 =>
            IF (sequence_in = '0') THEN
                -- "010"
                next_state <= S2;
            ELSE
                next_state <= Init;
            END IF;
        WHEN S2 =>
            IF (sequence_in = '1') THEN
                -- "0101"
                next_state <= S3;
            ELSE
                -- "0"
                next_state <= S0;
            END IF;
        WHEN S3 =>
            IF (sequence_in = '0') THEN
                -- "01010"
                next_state <= S4;
            ELSE
                next_state <= Init;
            END IF;
        WHEN S4 =>
            IF (sequence_in = '1') THEN
                -- "010101"
                next_state <= S5;
            ELSE
                -- "0"
                next_state <= S0;
            END IF;
        WHEN S5 =>
            IF (sequence_in = '0') THEN
                -- "0101010"
                next_state <= S6;
            ELSE
                next_state <= Init;
            END IF;
        WHEN S6 =>
            IF (sequence_in = '1') THEN
                -- "01010101"
                next_state <= S7;
            ELSE
                next_state <= S0;
```

```
                END IF;
            WHEN S7 =>
                IF (sequence_in = '0') THEN
                    -- "0101010"
                    next_state <= S6;
                ELSE
                    next_state <= Init;
                END IF;
        END CASE;
    END PROCESS;

    -- Output logic of the VHDL MOORE FSM Sequence Detector
    PROCESS (current_state)
    BEGIN
        CASE current_state IS
            WHEN S7 =>
                detector_out <= '1';
            WHEN OTHERS =>
                detector_out <= '0';
        END CASE;
    END PROCESS;
END Behavioral;
```

پیاده سازی test bench

```
-- VHDL project: VHDL code for Sequence Detector using Moore FSM
-- VHDL testbench for Moore FSM Sequence Detector
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb_VHDL_Moore_FSM_Sequence_Detector IS
END tb_VHDL_Moore_FSM_Sequence_Detector;

ARCHITECTURE behavior OF tb_VHDL_Moore_FSM_Sequence_Detector IS

    -- Component Declaration for the Moore FSM Sequence Detector in VHDL

    COMPONENT VHDL_MOORE_FSM_Sequence_Detector
        PORT (
            clock : IN STD_LOGIC;
            reset : IN STD_LOGIC;
            sequence_in : IN STD_LOGIC;
            detector_out : OUT STD_LOGIC
        );
    END COMPONENT;
    --Inputs
    SIGNAL clock : STD_LOGIC := '0';
    SIGNAL reset : STD_LOGIC := '0';
    SIGNAL sequence_in : STD_LOGIC := '0';

    --Outputs
    SIGNAL detector_out : STD_LOGIC;

    -- Clock period definitions
    CONSTANT clock_period : TIME := 10 ns;

BEGIN

    -- Instantiate the Moore FSM Sequence Detector in VHDL
    uut : VHDL_MOORE_FSM_Sequence_Detector PORT MAP(
        clock => clock,
        reset => reset,
        sequence_in => sequence_in,
        detector_out => detector_out
    );

    -- 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
        -- hold reset state for 100 ns.
        sequence_in <= '0';
        reset <= '1';
        -- Wait 100 ns for global reset to finish
        WAIT FOR 30 ns;
        reset <= '0';
        WAIT FOR 40 ns;
        sequence_in <= '0';
        WAIT FOR 10 ns;
        sequence_in <= '1';
        WAIT FOR 10 ns;
        sequence_in <= '0';
        WAIT FOR 20 ns;
        sequence_in <= '1';
        WAIT FOR 20 ns;
        sequence_in <= '0';
        WAIT FOR 20 ns;
        sequence_in <= '1';
        WAIT FOR 20 ns;
        sequence_in <= '0';
        WAIT FOR 20 ns;
        sequence_in <= '1';
        WAIT FOR 20 ns;
        sequence_in <= '0';
        WAIT FOR 20 ns;
        sequence_in <= '1';
        -- insert stimulus here
        WAIT;
    END PROCESS;

END;
```

Simulation

تست بنچ نوشته شده را simulate می‌کنیم تا موج های خروجی نمایش داده شود.