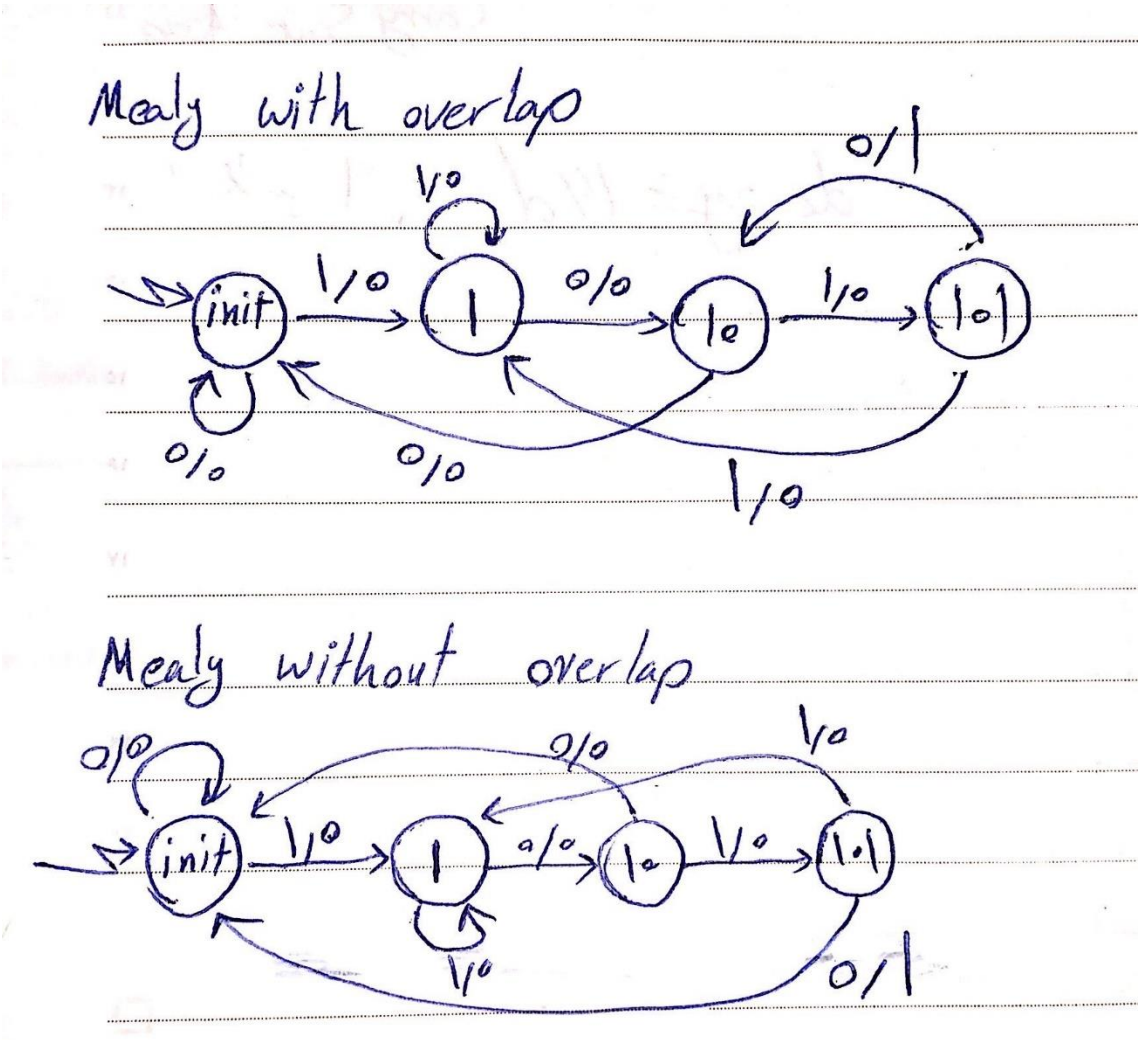
به نام خدا

گزارش کار آزمایش چهارم

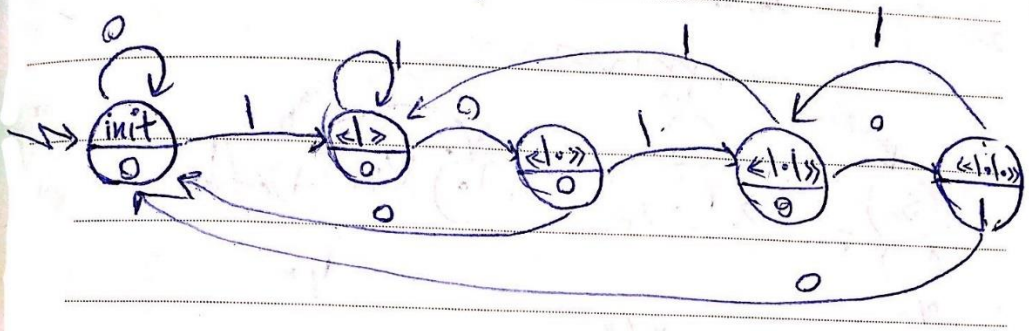
علی نوروزبیگی – فرهاد امان

هدف آزمایش: پیاده سازی Sequence Detector برای دو دنباله 1110 و 1010 با استفاده از ماشین مور و میلی و برای دوحالت بدون overlap و با overlap

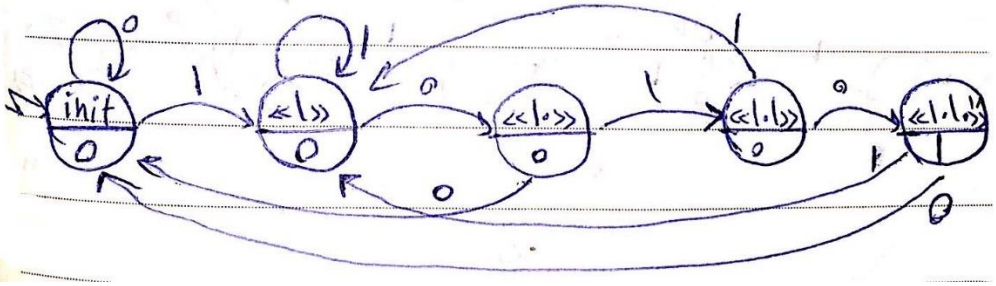
برای پیاده سازی یک Sequence Detector نیاز به پیاده سازی ماشین حالت داریم برای این کار ابتدا دیاگرام های حالت مورد نیاز را رسم می کنیم.



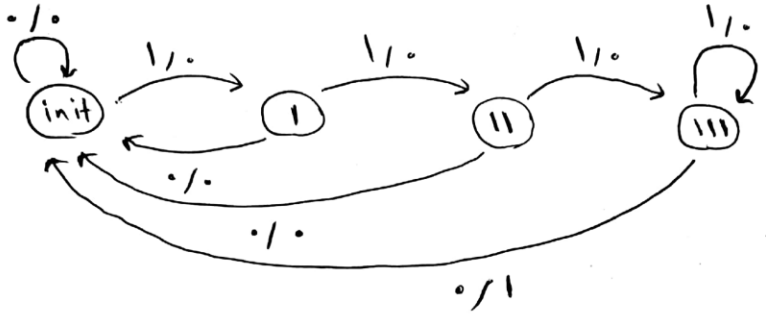
Moore with overlap



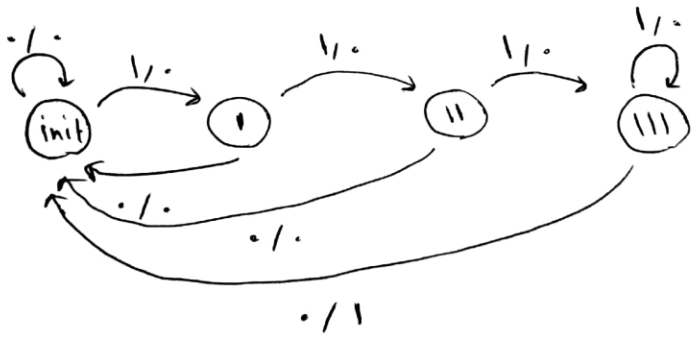
Moore without overlap



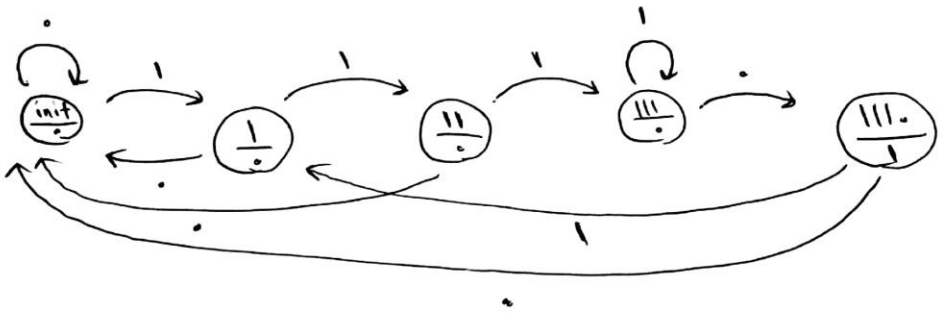
Mealy with overlap



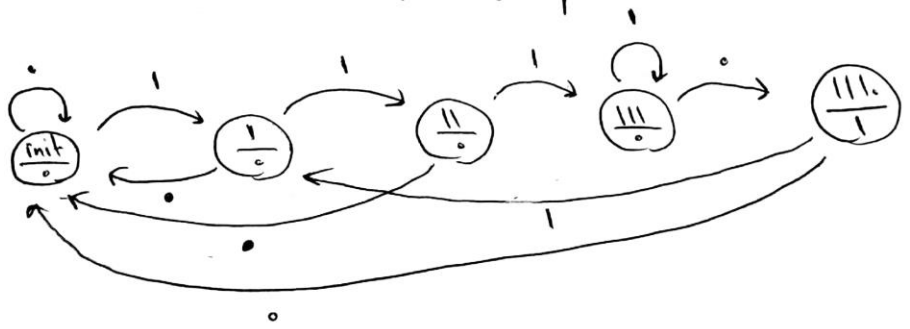
Mealy Without overlap



Moore With Overlap



Moore Without Overlap



همانطور که مشاهده می شود در ماشین های مور خروجی تنها به state فعلی وابسته است و به ورودی بستگی ندارد اما در ماشین های میلی خروجی هم به حالت فعلی هم به ورودی بستگی دارد.

در دنباله 1110 امکان overlap شدن وجود ندارد و ماشین مربوط به حالت with overlap و without overlap هیچ تفاوتی ندارد.

```
entity seq_1010_0 is
	port( clk, reset: in std_logic;
			input : in std_logic;
			output : out std_logic
			);
end seq_1010_0;
```

در بخش اول موجودیت ماشین تعیین می شود. چون مدار ترتیبی است کلاک یکی از ورودی های مدار است. و همچنین مدار دارای یک ریست آسنکرون می باشد.

```
process (clk, reset)
	begin
		if(reset = '1') then
			current_state <= init;
		elsif (rising_edge(clk)) then
			current_state <= next_state;
		end if;
	end process;
```

بخش architecture ماشین دارای دو process می باشد. Process اول به clk و reset حساس است علت وجود reset در کنار clk نشان می دهد که ریست از نوع آسنکرون می باشد و مستقل از کلاک عمل می کند.

```
process (current_state, input)
        begin
                case current_state is
                        when init =>
                                if input='0' then
                                        next_state <= init;
                                        output <= '0';
                                else
                                        next_state <= got_1;
                                        output <= '0';
                                end if;
                        when got_1 =>
                                if input='1' then
                                        next_state <= got_1;
                                        output <= '0';
                                else
                                        next_state <= got_10;
                                        output <= '0';
                                end if;
                        when got_10 =>
                                if input='1' then
                                        next_state <= got_101;
                                        output <= '0';
                                else
```

```
                next_state <= init;
                output <= '0';
            end if;
        when got_101 =>
            if input='1' then
                next_state <= got_1;
                output <= '0';
            else
                next_state <= got_10;
                output <= '1';
            end if;
    end case;
end process;
```

در process دوم روند کلی ماشین تشریح میشود. در یک ساختار switch case تمام ورودی ها و نتیجه آن ها بررسی می شود. ماشینی که در بالا کد آن آمده است یک ماشین میلی می باشد به همین دلیل ورودی در خروجی حالت ها تاثیر گذار است. اما در ماشین های مور ورودی در خروجی هر حالت تاثیرگذار نیست.

```
type states is (init, got_1, got_10, got_101);
signal current_state, next_state : states;
```

همچنین قبل از شروع process ها نیاز به تعریف state ها و signal های مورد نیاز داریم.