

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

پروژه ۵ درس مدار منطقی
دکتر نوابی

مهدی وجهی

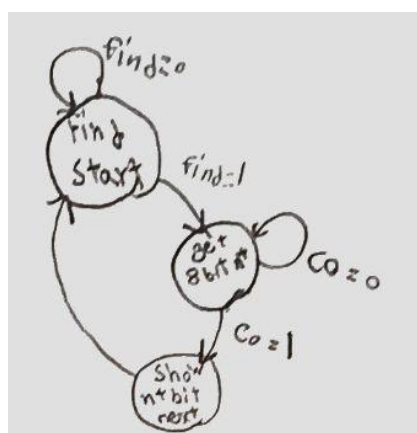
۸۱۰۱۰۱۵۵۸

بخش a

شرح اجمالی

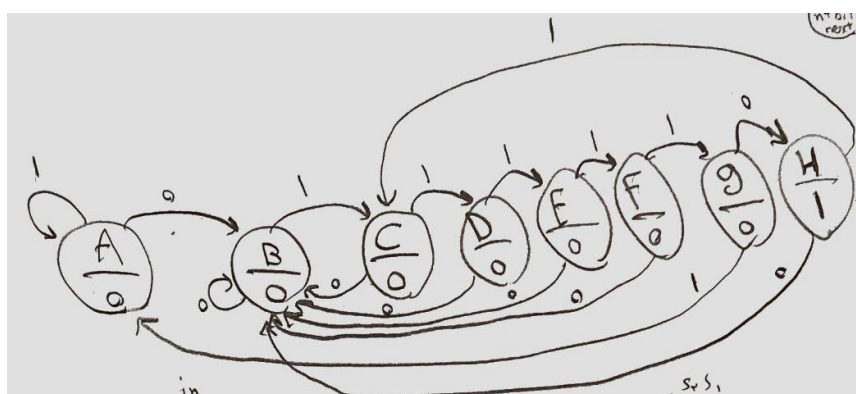
مدار این پروژه ۳ مرحله دارد:

1. پیدا کردن یک توالی (۰۱۱۱۱۱۰) در ورودی
 2. گرفتن یک ورودی ۸ بیت به صورت سریالی و ذخیره آن
 3. نمایش اعداد ورودی بر روی خروجی به تعداد عدد ۸ بیتی مرحله قبل
- که نمودار آن به طور خلاصه به شکل زیر است

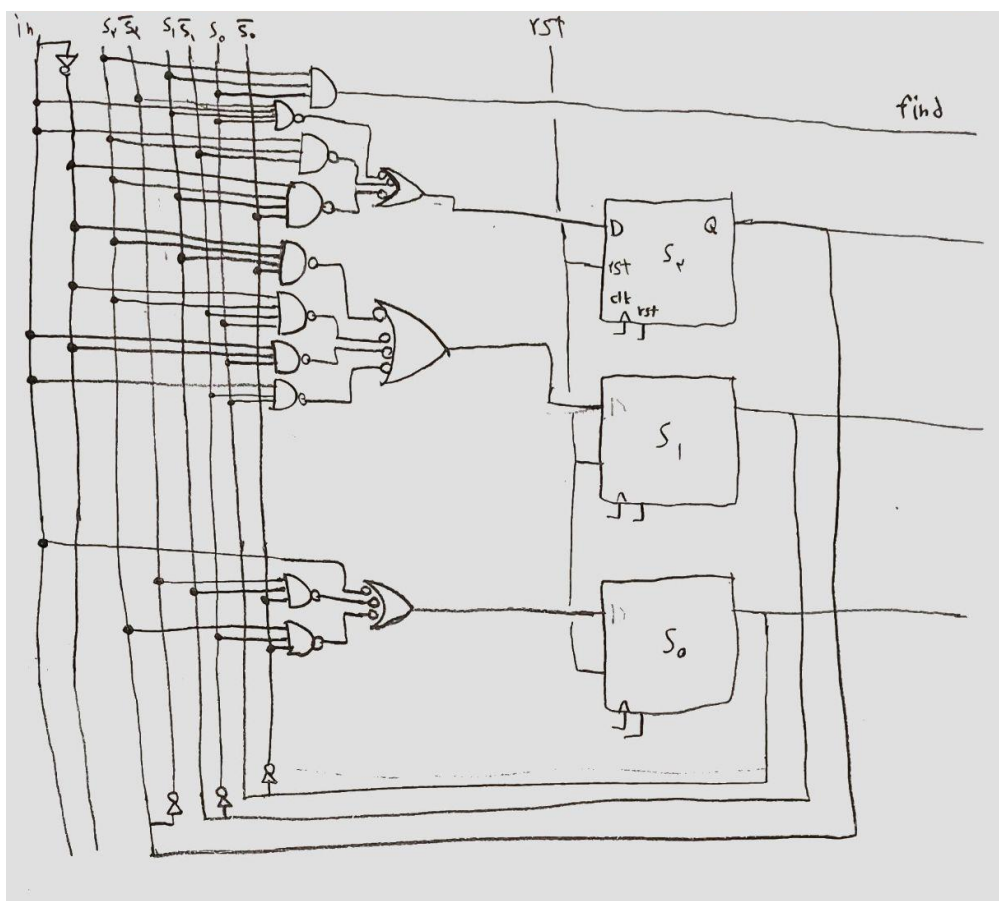
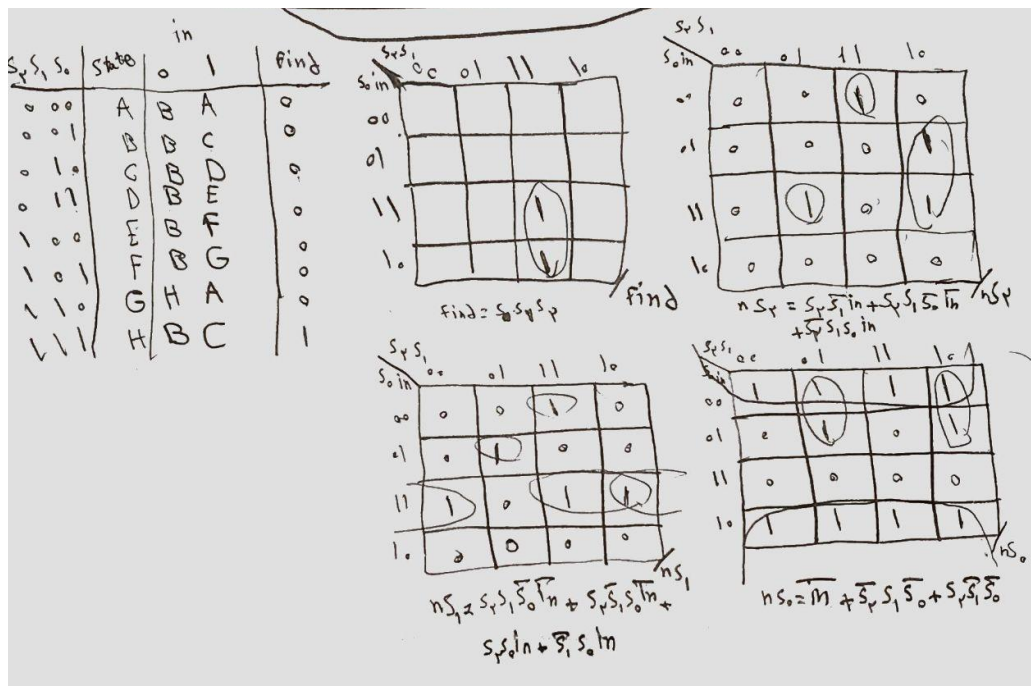


پیاده سازی مدار شناسایی ۰۱۱۱۱۱۰

حال در این بخش باید توالی (۰۱۱۱۱۱۰) را شناسایی کنیم برای این کار یک moore machine ساده طراحی می کنیم که نمودار آن به شکل زیر است:



حال مدار آن را به صورت زیر پیاده می کنیم:



نوشتن کد

حال که مدار طراحی شد کد آن می نویسیم که شامل ۳ بخش است:

- مدار بدون حافظه

- مدار حافظه
- مدار خروجی

```

module qa(input in, rst, clk, output find);
    logic [2:0] ns,ps;
    parameter [2:0] A = 3'b000,
                  B = 3'b001,
                  C = 3'b010,
                  D = 3'b011,
                  E = 3'b100,
                  F = 3'b101,
                  G = 3'b110,
                  H = 3'b111;

    // combinational part
    always @(ps,in) begin
        ns = A;
        case (ps)
            A: ns = in ? A : B;
            B: ns = in ? C : B;
            C: ns = in ? D : B;
            D: ns = in ? E : B;
            E: ns = in ? F : B;
            F: ns = in ? G : B;
            G: ns = in ? A : H;
            H: ns = in ? C : B;
            default: ns = A;
        endcase
    end

    // output
    assign find = (ps == H) ? 1'b1 : 1'b0;

    // sequential part
    always @(posedge clk, posedge rst) begin
        if (rst)
            ps <= A;
        else
            ps <= ns;
        end
    end
endmodule

```

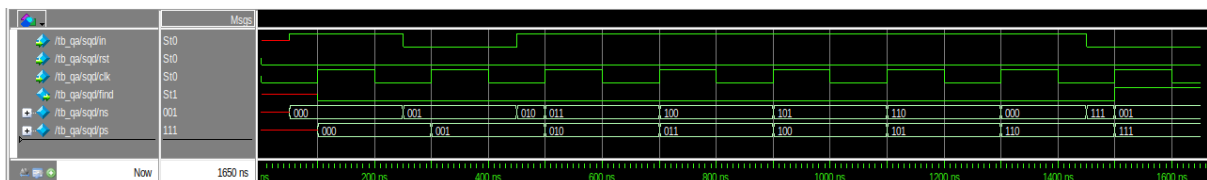
زیر بخش ۱

در این بخش لازم است یک میز آزمایش برای مدار خود طراحی کنیم. این قسمت پیچیدگی خاصی ندارد فقط کافیه یک ورودی غلط بدهیم سپس ورودی درست هم بدهیم و صحت آن را بسنجیم. کد آن به صورت زیر است:

```
`timescale 1ns/1ns
module tb_qa();
    logic in, clk = 0, rst = 0;
    wire out;
    qa sqd(.in(in), .clk(clk), .rst(rst), .find(out));

    always #100 clk = ~clk;
    initial begin
        #50 in = 1;
        #200 in = 0;
        #200 in = 1;
        #200 in = 1;
        #200 in = 1;
        #200 in = 1;
        #200 in = 1;
        #200 in = 0;
        #200 $stop;
    end
endmodule
```

موج خروجی مدار به شکل زیر است:



زیر بخش ۲

synthesising

ابتدا باید کد را synthesis کنیم.

Flow Status	Successful - Sun Dec 31 21:27:49 2023
Quartus Prime Version	23.1std.0 Build 991 11/28/2023 SC Lite Edition
Revision Name	ca5
Top-level Entity Name	qa
Family	Cyclone IV GX
Device	EP4CGX15BF14A7
Timing Models	Final
Total logic elements	7 / 14,400 (< 1 %)
Total registers	7
Total pins	4 / 81 (5 %)
Total virtual pins	0
Total memory bits	0 / 552,960 (0 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)

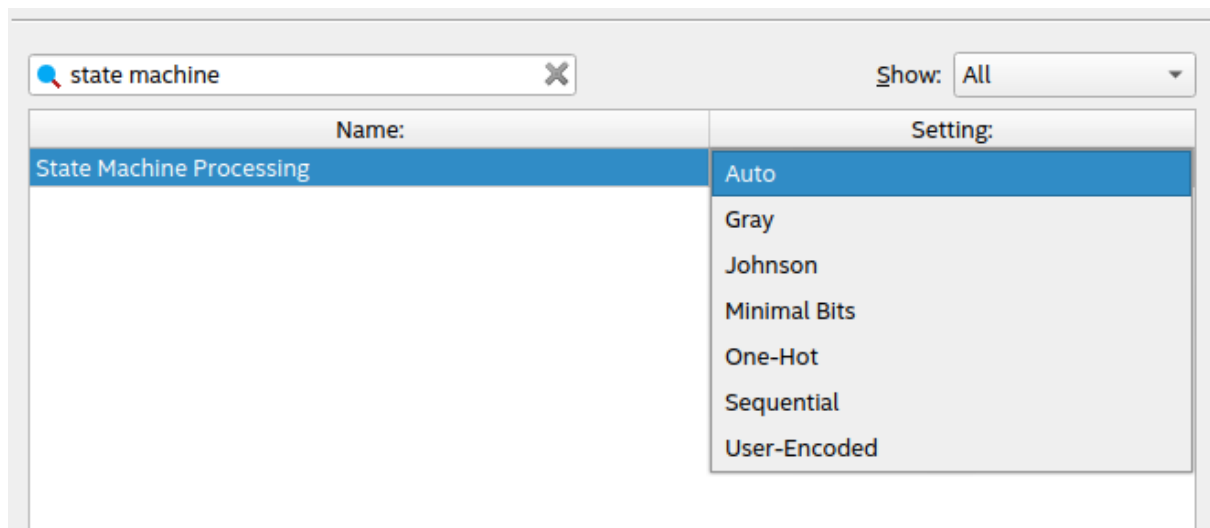
اما مشاهده می شود که ۷ رجیستر استفاده شده این یعنی از روش دیگری استفاده شده که encode آن به شکل زیر است:

	Name	H	G	F	E	D	C	B	A
1	A	0	0	0	0	0	0	0	0
2	B	0	0	0	0	0	0	1	1
3	C	0	0	0	0	0	1	0	1
4	D	0	0	0	0	1	0	0	1
5	E	0	0	0	1	0	0	0	1
6	F	0	0	1	0	0	0	0	1
7	G	0	1	0	0	0	0	0	1
8	H	1	0	0	0	0	0	0	1

لازم از از طریق

Edit settings -> compiler settings -> advanced settings (synthesis) -> state machine processing

این موضوع را اصلاح کنیم:



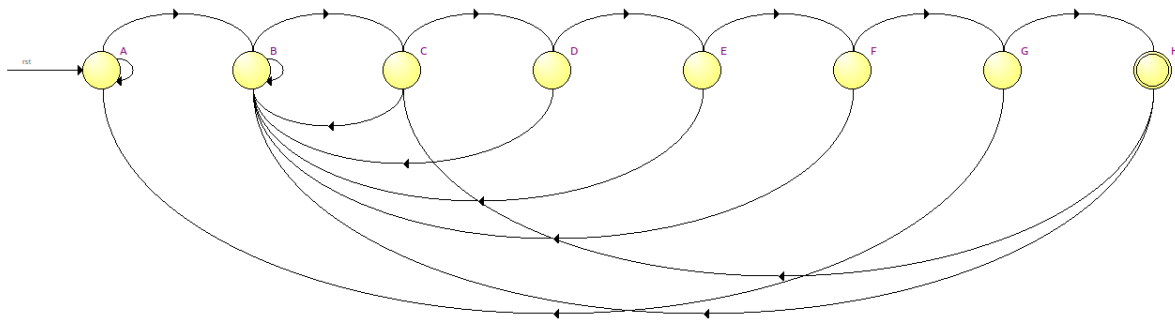
همانطور که مشاهده می شود انواع مختلفی وجود دارد ما بر روی user-encoded می گذاریم که با کد ما همخوانی داشته باشد.

حال مجدد خروجی می گیریم که نتیجه به صورت زیر است:

Flow Status	Successful - Sun Dec 31 21:38:04 2023
Quartus Prime Version	23.1std.0 Build 991 11/28/2023 SC Lite Edition
Revision Name	ca5
Top-level Entity Name	qa
Family	Cyclone IV GX
Device	EP4CGX15BF14A7
Timing Models	Final
Total logic elements	4 / 14,400 (< 1 %)
Total registers	3
Total pins	4 / 81 (5 %)
Total virtual pins	0
Total memory bits	0 / 552,960 (0 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)

بررسی قطعات مدار و پیاده سازی آن

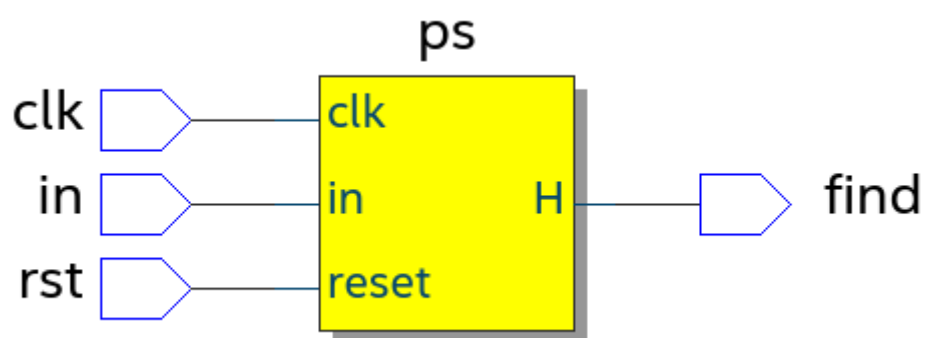
نمودار و همچنین encode آن نیز تطابق دارد.



	Name	ps~6	ps~5	ps~4
1	A	0	0	0
2	B	0	0	1
3	C	0	1	0
4	D	0	1	1
5	E	1	0	0
6	F	1	0	1
7	G	1	1	0
8	H	1	1	1

	Source State	Destination State	Condition
1	A	B	(lin)
2	A	A	(in)
3	B	C	(in)
4	B	B	(lin)
5	C	D	(in)
6	C	B	(lin)
7	D	B	(lin)
8	D	E	(in)
9	E	F	(in)
10	E	B	(lin)
11	F	B	(lin)
12	F	G	(in)
13	G	A	(in)
14	G	H	(lin)
15	H	C	(in)
16	H	B	(lin)

مدار آن نیز به شکل زیر است:



در گزارش می بینیم:

	Statistic	Value
1	Total registers	3
2	Number of registers using Synchronous Clear	0
3	Number of registers using Synchronous Load	0
4	Number of registers using Asynchronous Clear	3
5	Number of registers using Asynchronous Load	0
6	Number of registers using Clock Enable	0
7	Number of registers using Preset	0


که همان ۳ رجیستری است که در شکل دستی هست. همچنین داریم که:


	Resource	Usage
1	I/O pins	4
2		
3	DSP block 9-bit elements	0
4		
5	Maximum fan-out node	ps~4
6	Maximum fan-out	4
7	Total fan-out	29
8	Average fan-out	1.93

که نشان می دهد ما ۴ ورودی داریم که در شکل نیز مشخص بود آنها find,clk,rst,in هستند. همچنین در گزارش کلی می بینیم که ما از 4/14,400 مدار یعنی کمتر از ۱ درصد آن استفاده کردیم. برای درک بهتر این موضوع پیاده سازی آن را روی برد بررسی می کنیم.

بررسی مدار از لحاظ زمانی

ابتدا داده های زمانی خود کوارتز را بررسی می کنیم.

Slow 1200mV 125C Model Fmax Summary				
 <<Filter>>				
	Fmax	Restricted Fmax	Clock Name	Note
1	782.47 MHz	250.0 MHz	clk	limit due to minimum period restriction (max I/O toggle rate)

Slow 1200mV -40C Model Fmax Summary				
 <<Filter>>				
	Fmax	Restricted Fmax	Clock Name	Note
1	933.71 MHz	250.0 MHz	clk	limit due to minimum period restriction (max I/O toggle rate)

در این قسمت گفته شده که مدار در دمای ۱۲۵ درجه می تواند تا 782MHz کار کند اما به دلیل محدودیت سایکلون ۴ این مقدار حداکثر ۲۵۰ است همچنین در دمای ۴۰- درجه مدار می تواند با فرکانس ۹۳۳ کار کند. حال باید در مدل سیم مدار را شبیه سازی کنیم برای این کار فایل ca5/simulation/ca5.svo را به مدل سیم اضافه می کنیم سپس برای شبیه سازی کتابخانه های زیر را انتخاب می کنیم و sdo مربوطه را به آن می دهیم:

