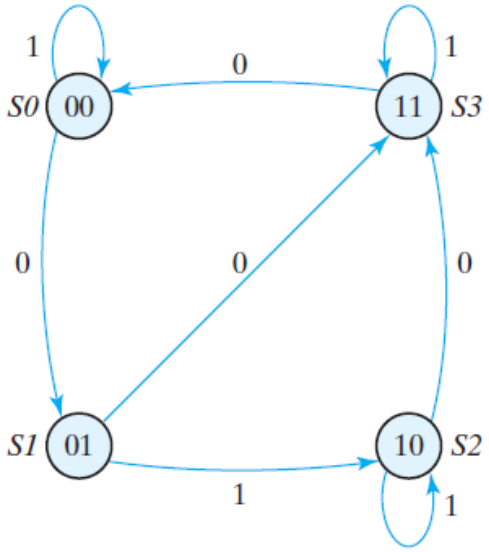
طراحی کامپیوتری سیستم‌های دیجیتال

جلسه هفتم



ماشین متناهی-مدل مور

در این مدل از ماشین متناهی، خروجی فقط تابعی از حالت فعلی است.



Present State		Input	Next State	
A	B	x	A	B
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

دیاگرام حالت

جدول حالت

*نکته: خروجی مدار مستقل از ورودی است.

*در این مثال خروجی مساوی مقدار بردار حالت فعلی می باشد.

مدل سازی رفتاری

ماشین متناهی- مدل مور

```
module Moore_Model        (
  output [1: 0]           y_out,
  input                   x_in, clock, reset
);
  reg [1: 0]              state;
  parameter               S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;

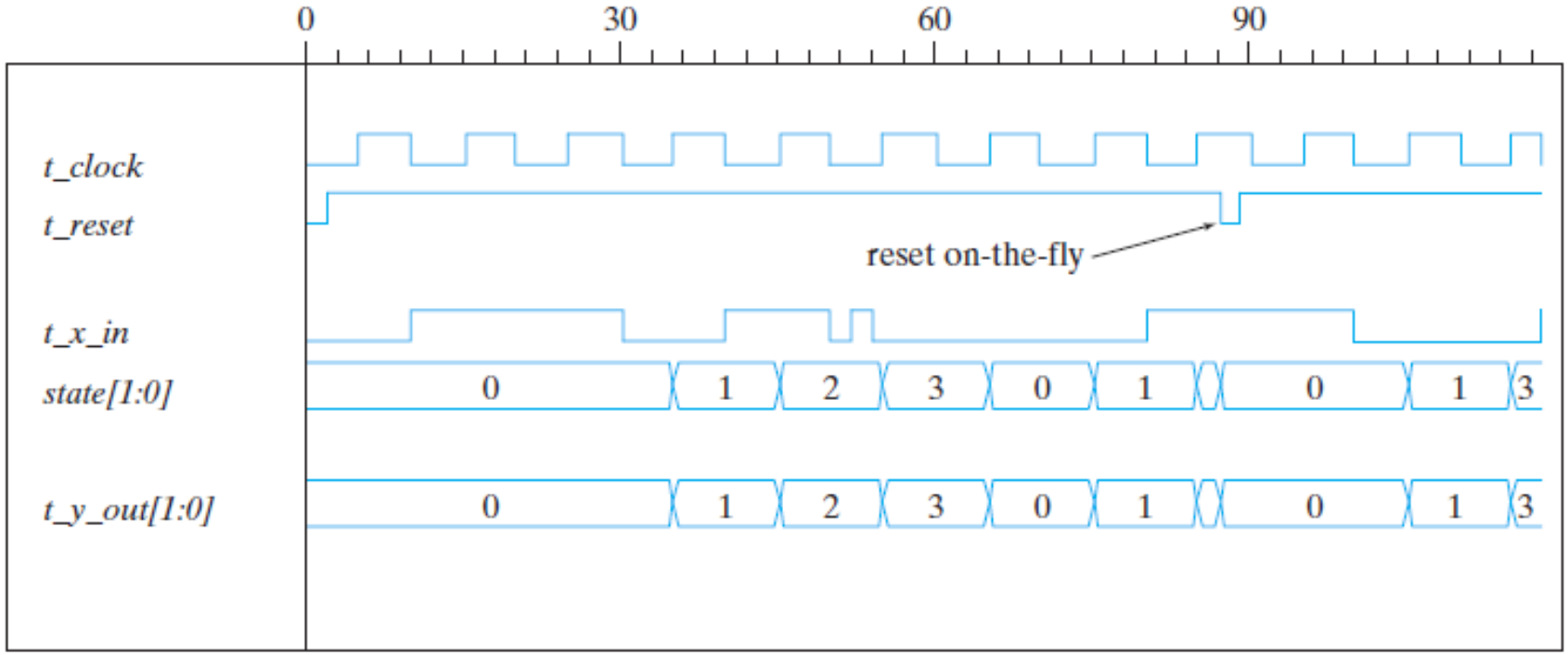
  always @ (posedge clock, negedge reset)
    if (reset == 0) state <= S0;                    // Initialize to state S0
    else case (state)
      S0:     if (~x_in) state <= S1; else state <= S0;
      S1:     if (x_in)  state <= S2; else state <= S3;
      S2:     if (~x_in) state <= S3; else state <= S2;
      S3:     if (~x_in) state <= S0; else state <= S3;
    endcase

  assign y_out = state;       // Output of flip-flops

endmodule
```

مدل مور

موج های ورودی و خروجی

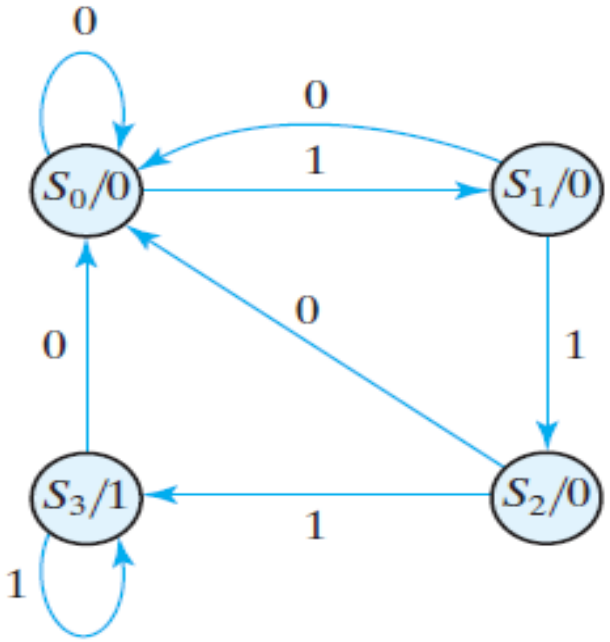


آشکار ساز دنباله

که دنباله ای مشخص از اعداد را در میان یک ورودی سریال (یک رشته متوالی از بیت ها که از طریق یک خط ورودی وارد می شود) تشخیص می دهد.

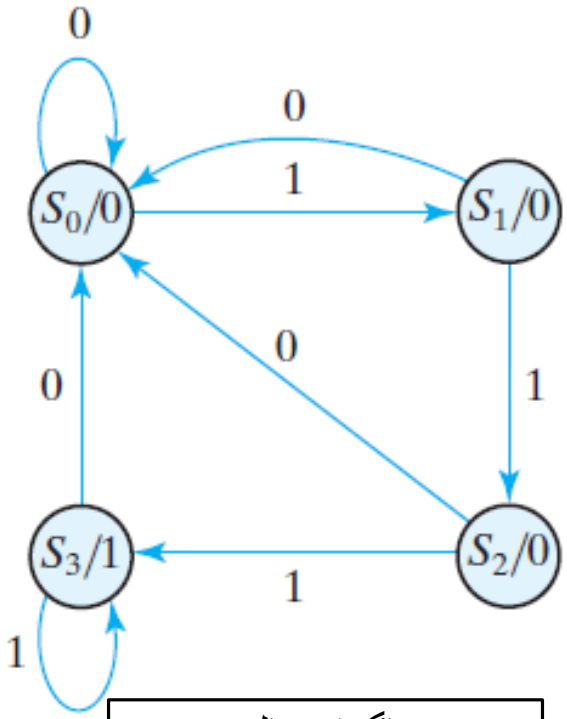
مثال: مداری طراحی کنید که دنباله ای از سه تا 1 متوالی یا بیشتر را در یک جریان بیت سریال (متوالی) تشخیص دهد. برای مدار خواسته شده دیاگرام حالت رسم کنید.

100111110010101011101100000001



آشکار ساز دنباله

مثال: اشکار ساز دنباله ای از سه یک متوالی یا بیشتر



دیاگرام حالت

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
S0	S0	S1	0	0
S1	S0	S2	0	0
S2	S0	S3	0	1
S3	S0	S3	0	1

جدول حالت

*این مدار ترتیبی از نوع مور است زیرا وقتی مدار در حالت S3 است خروجی1 و در غیر این صورت 0 است.

این آشکارساز را با زبان وریلاگ مدل کنید.

```
module Moore_seq_detec(
output reg y_out,
input x_in, clock, reset
);
reg [0:1] state ;
parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;

always @ ( posedge clock, negedge reset)
if (reset == 0) state <= S0; // Initialize to state S0

else case (state)
S0: if (~x_in) state <= S0; else state <= S1;
S1: if (~x_in) state <= S0; else state <= S2;
S2: if (~x_in) state <= S0; else state <= S3;
S3: if (~x_in) state <= S0; else state <= S3;
endcase

always @ ( posedge clock, negedge reset, x_in)
if(state==S3)
y_out=1;
else
y_out=0;
endmodule
```

ثبات Register

به گروهی از فلیپ فلاپ ها که به یک ساعت مشترک اتصال دارند و قادر به ذخیره سازی یک بیت از اطلاعات می باشند ثبات می گویند.

یک ثبات n بیتی متشکل از n فلیپ فلاپ است و قادر به ذخیره سازی n بیت اطلاعات دودویی می باشد.

علاوه بر فلیپ فلاپ ها ثبات ها ممکن است گیت های ترکیبی داشته باشند که وظیفه پردازش داده را بر عهده دارند. در واقع ثبات ها اطلاعات دودویی را نگه می دارند و گیت ها تعیین می کنند که چگونه اطلاعات به ثبات منتقل می شود.

ساده ترین نوع ثبات، ثباتی است که فقط از فلیپ فلاپ (بدون هیچ گیتی) تشکیل شده باشد.

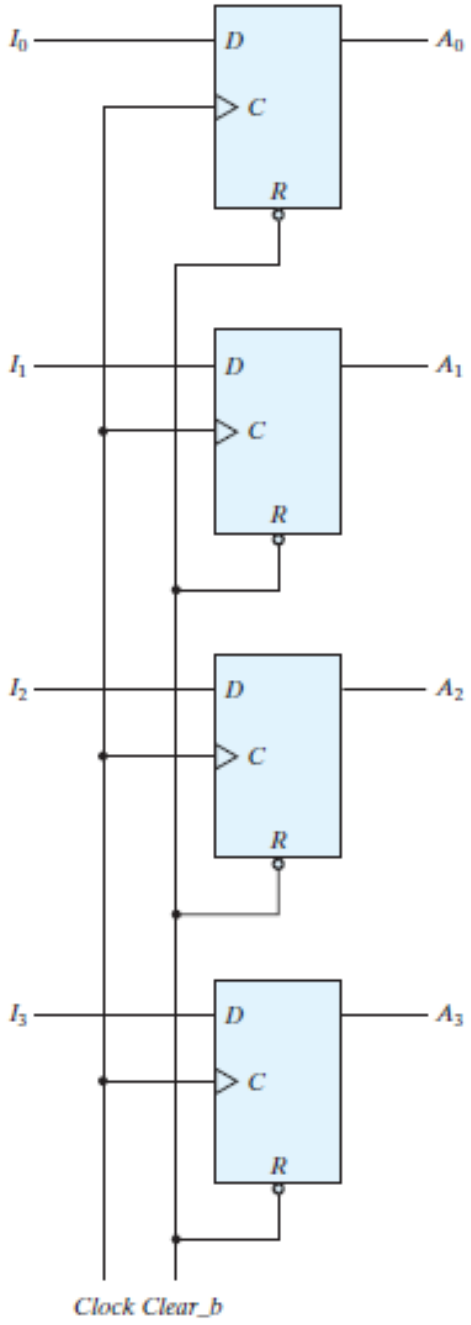
ثبات

نمونه ای از یک ثبات چهاربیتی

که تمام فلیپ فلاپ های آن با لبه مثبت پالس ساعت

تریگر می شود و داده های دودویی موجود در

چهار ورودی به داخل ثبات منتقل می شود.



ثبات

تعریف یک ثبات n بیتی

```
module  regn (D, Clock, Resetn, Q);
    parameter n = 16;
    input  [n–1:0] D;
    input  Clock, Resetn;
    output  reg  [n–1:0] Q;

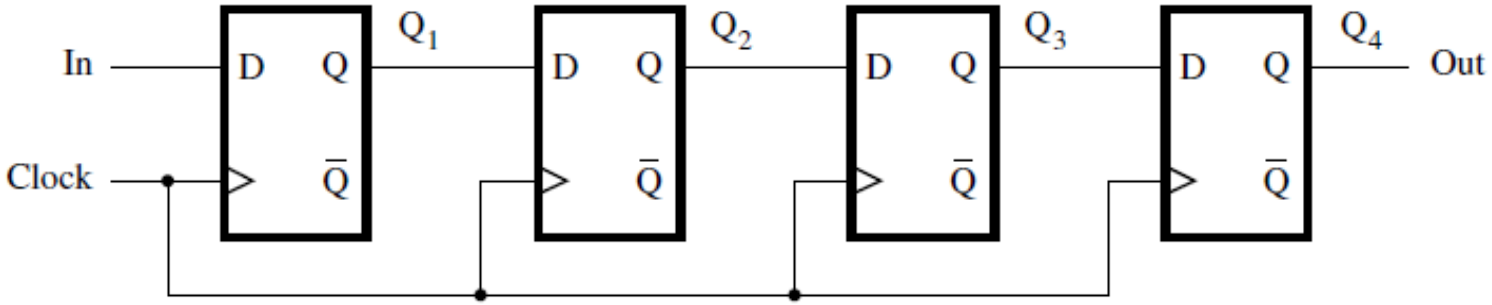
    always @(negedge Resetn, posedge Clock)
        if (!Resetn)
            Q <= 0;
        else
            Q <= D;

endmodule
```

ثبات های شیفت Shift Registers

یک ثبات شیفت قادر به شیفت دهی اطلاعات دودویی موجود در هر سلول به سلول همسایگی اش، در جهت انتخاب شده، می باشد.

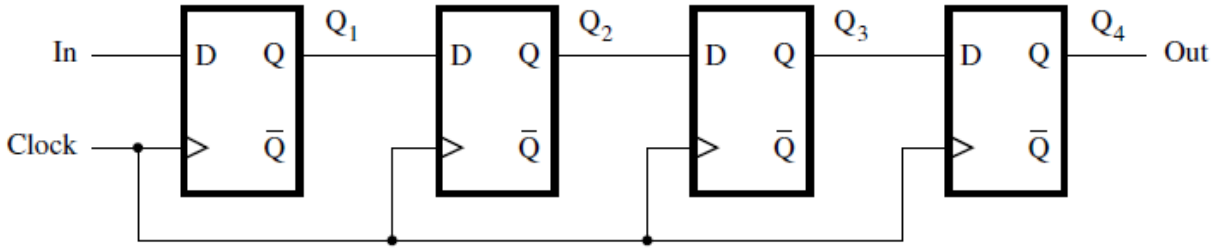
ثبات شیفت متشکل از زنجیره ای از فلیپ فلاپ ها به طور متوالی است به طوری که خروجی هر فلیپ فلاپ، متصل به ورودی فلیپ فلاپ بعدی است و تمام فلیپ فلاپ ها در پالس ساعت مشترک هستند.



ثبات شیفت چهار بیتی تک جهتی (چپ به راست)

عملکرد ثبات شیفت

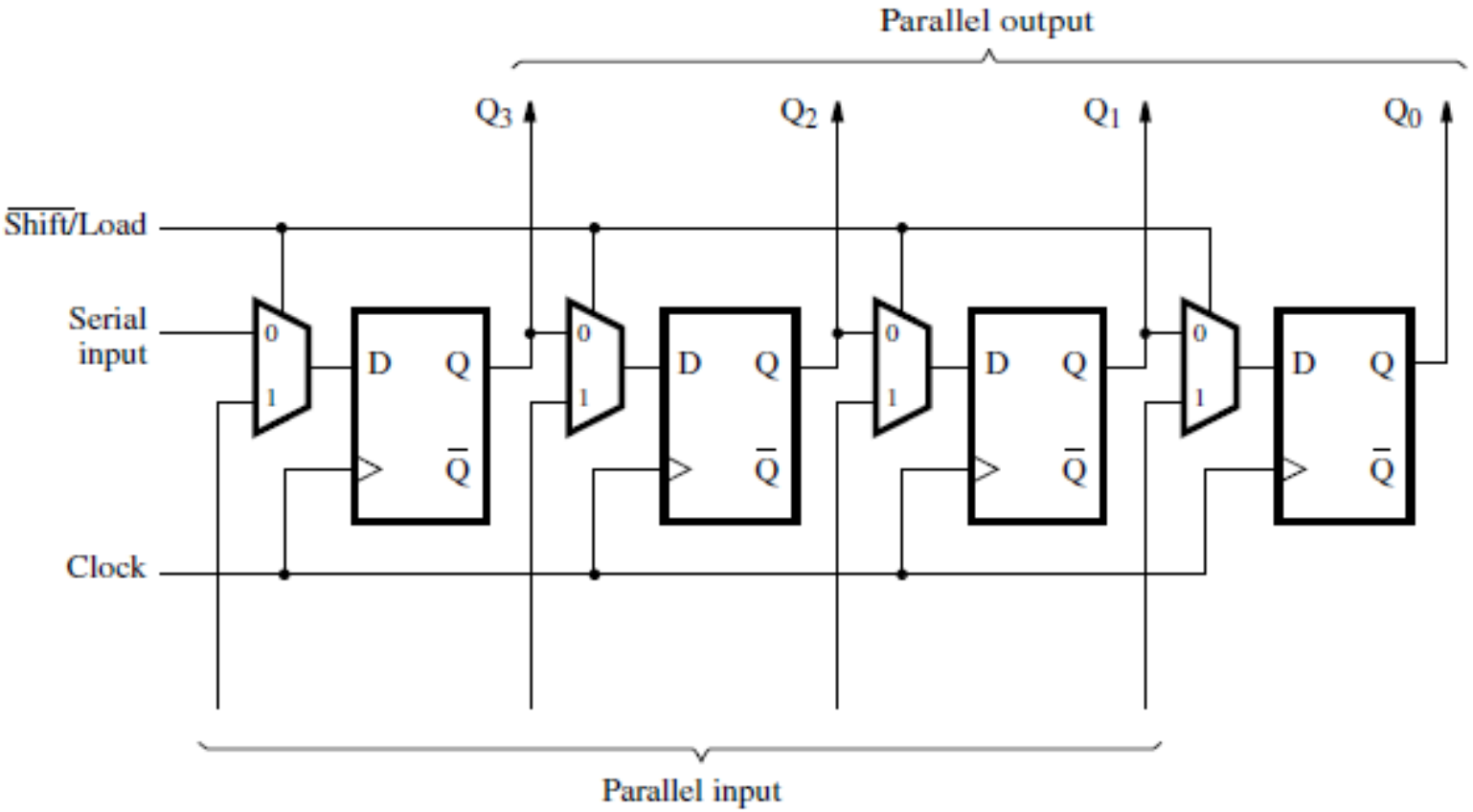
یک ثبات انتقالی ساده



	In	Q_1	Q_2	Q_3	Q_4 = Out
t_0	1	0	0	0	0
t_1	0	1	0	0	0
t_2	1	0	1	0	0
t_3	1	1	0	1	0
t_4	1	1	1	0	1
t_5	0	1	1	1	0
t_6	0	0	1	1	1
t_7	0	0	0	1	1

ثبات انتقالی با دسترسی موازی

ثبات انتقالی چهاربیتی با دسترسی موازی



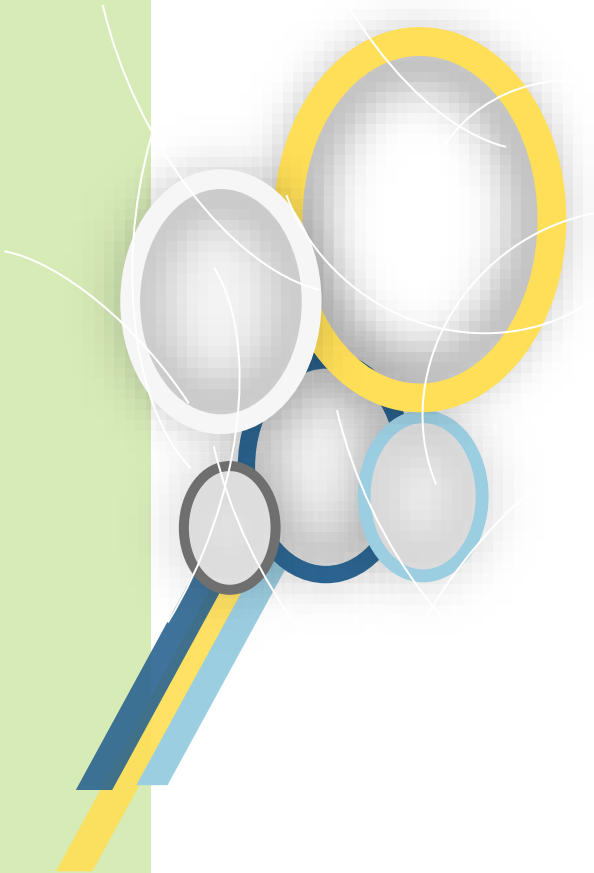
ثبات انتقالی با دسترسی موازی

ثبات انتقالی چهاربیتی

```
module  shift4 (R, L, w, Clock, Q);
    input  [3:0] R;
    input  L, w, Clock;
    output  reg  [3:0] Q;

    always @(posedge Clock)
        if (L)
            Q <= R;
        else
        begin
            Q[0] <= Q[1];
            Q[1] <= Q[2];
            Q[2] <= Q[3];
            Q[3] <= w;
        end

endmodule
```



خسته نباشید