ESEMPIO di SINTESI di una UNITA' con Registro MJR

```
//-------------------------------------------------------------
module esempio_uso_MJR(x1,x0,out, clock,reset_);
 input  clock, reset_;
 input  x1,x0;
 output out;
 reg OUT; assign out=OUT ;
 reg [1:0] STAR,MJR;
 parameter S0=0, S1=1, S2=2, Sp=3;
  always @(reset_==0) begin OUT<=0; STAR<=S0; end
  always @(posedge clock) if (reset_==1) #3
   casex(STAR)
    S0: begin OUT<=1; MJR<=((x1|x0)==0)?S0:(x1==1)?S1:S2; STAR<=Sp; end
    S1: begin OUT<=0; STAR<=S2; end
    S2: begin OUT<=x1&x0; STAR<=S0; end
    Sp: begin STAR<=MJR; end
   endcase
endmodule
//-------------------------------------------------------------
```

Primo passo di sintesi

```
//-------------------------------------------------------------
module esempio_uso_MJR(x1,x0,out, clock,reset_);
 input  clock, reset_;
 input  x1,x0;
 output out;
 reg OUT; assign out=OUT ;
 reg [1:0] STAR,MJR;
 parameter S0=0, S1=1, S2=2, Sp=3;

// Registro operativo OUT
  always @(reset_==0) begin OUT<=0; end
  always @(posedge clock) if (reset_==1) #3
   casex(STAR)
    S0: OUT<=1;
    S1: OUT<=0;
    S2: OUT<=x1&x0;
    Sp: OUT<=OUT;
   endcase

// Registro operativo MJR
 always @(posedge clock) #3
   casex(STAR)
    S0:       MJR<=((x1|x0)==0)?S0:(x1==1)?S1:S2;
    S1,S2,Sp: MJR<=MJR;
   endcase

// Registro di stato STAR
  always @(reset_==0) begin STAR<=S0; end
  always @(posedge clock) if (reset_==1) #3
   casex(STAR)
    S0: begin STAR<=Sp; end
    S1: begin STAR<=S2; end
    S2: begin STAR<=S0; end
    Sp: begin STAR<=MJR; end
   endcase

endmodule
//-------------------------------------------------------------
```

Secondo passo di sintesi

```
//-------------------------------------------------------------
module esempio_uso_MJR(x1,x0,out, clock,reset_);
 input  clock, reset_;
 input  x1,x0;
 output out;
 reg OUT; assign out=OUT ;
 reg [1:0] STAR,MJR; parameter S0=0, S1=1, S2=2, Sp=3;

// Registro operativo OUT
 wire b1,b0;
 assign {b1,b0}=(STAR==S0)?00:
                (STAR==S1)?01:
                (STAR==S2)?10:
                /*default*/11;
  always @(reset_==0) begin OUT<=0; end
  always @(posedge clock) if (reset_==1) #3
   casex({b1,b0})
    00: OUT<=1;
    01: OUT<=0;
    10: OUT<=x1&x0;
    11: OUT<=OUT;
   endcase

// Registro operativo MJR
  wire b2;
 assign b2=(STAR==S0)?0:1;
  always @(posedge clock) if (reset_==1) #3
   casex(b2)
    0: MJR<=((x1|x0)==0)?'B00:(x1==1)?'B01:'B10;// ovvero MJR<={!x1&x0,x1};
    1: MJR<=MJR;
   endcase

// Registro di stato STAR
  always @(reset_==0) begin STAR<=S0; end
  always @(posedge clock) if (reset_==1) #3
   casex(STAR)
    S0: begin STAR<=Sp; end
    S1: begin STAR<=S2; end
    S2: begin STAR<=S0; end
    Sp: begin STAR<=MJR; end
   endcase

endmodule
//-------------------------------------------------------------
```

Ultimo passo di sintesi

```
//-------------------------------------------------------------
module esempio_uso_MJR(x1,x0,out,clock,reset_);
 input clock,reset_;
 input x1,x0;
 output out;
 wire b1,b0,b2;
 wire [1:0] mjr;
 Parte_Operativa PO(x1,x0,out,b2,b1,b0,clock,mjr,reset_);
 Parte_Controllo PC(b2,b1,b0,mjr,clock,reset_);
endmodule

module Parte_Operativa(x1,x0,out,b2,b1,b0,clock,mjr,reset_);
 input clock,reset_;
 input x1,x0;
 output out;
 input b2,b1,b0;
 output[1:0] mjr;
 reg  OUT;   assign out=OUT;
 reg [1:0] MJR; assign mjr=MJR;
  always @(reset_==0) begin OUT<=0; end
  always @(posedge clock) if (reset_==1) #3
   casex({b1,b0})
    'B00: OUT<=1;
    'B01: OUT<=0;
    'B10: OUT<=x1&x0;
    'B11: OUT<=OUT;
   endcase
  always @(posedge clock) if (reset_==1) #3
   casex(b2)
    0: MJR<={!x1&x0,x1};
    1: MJR<=MJR;
   endcase
endmodule


module Parte_Controllo(b2,b1,b0,mjr,clock,reset_);
 input clock,reset_;
 input [1:0] mjr;
 output b2,b1,b0;
 reg [1:0] STAR; parameter S0=0,S1=1,S2=2,Sp=3;
 assign {b2,b1,b0}=(STAR==S0)?'B000:
                   (STAR==S1)?'B101:
                   (STAR==S2)?'B110:
                   (STAR==Sp)?'B111:
                   /*default*/'BXXX;
  always @(reset_==0) begin STAR<=S0; end
  always @(posedge clock) if (reset_==1) #3
  casex(STAR)
   S0: STAR<=Sp;
   S1: STAR<=S2;
   S2: STAR<=S0;
   Sp: STAR<=mjr;
  endcase
endmodule
```

Nella parte Controllo non vi sono micro salti a due alternative per cui mancano le variabili di condizionamento. Il modello è pertanto quello riportato nella figura seguente dove la tabella di verità della ROM della Parte Controllo è:

	microindirizzo		microistruzione				
	y1	y0	b2	b1	b0	next_μaddress	μtype
Linea corrispondente a S0	0	0	0	0	0	1 1	0
Linea corrispondente a S1	0	1	1	0	1	1 0	0
Linea corrispondente a S2	1	0	1	1	0	0 0	0
Linea corrispondente a Sp	1	1	1	1	1	- -	1