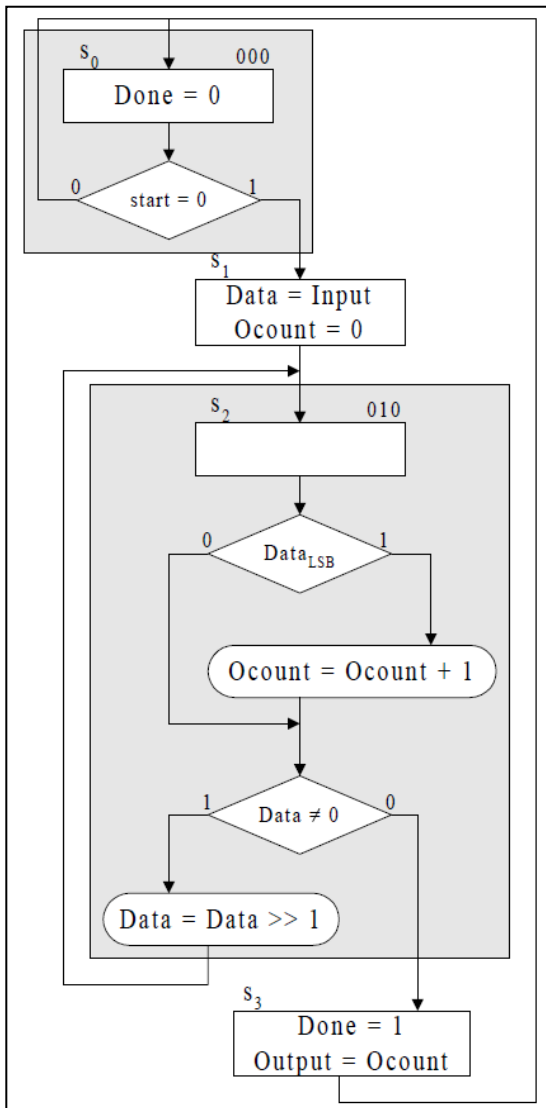




Az ASM Mealy egy állapotán belül leírtak egymásutániséga nem végrehajtási sorrendet jelent. Értelmezése az s2 állapotban pl. a következőt adja:

Ha következő órajel aktív éle alatt ($Data_{LSB} == 1$) akkor végre kell hajtani az $Ocount = Ocount + 1$ értékadást.

Ha következő órajel aktív éle alatt ($Data$

$!= 0$) nem igaz, akkor végre kell hajtani a $Data = Data \gg 1$ értékadást. Tehát ha a feltételek olyanok (itt $Data != 0$ ÉS $Data_{LSB} == 1$), akkor ezek az értékadások egyszerre történnek meg az órajel következő aktív élére.

A feladat Verilog leírása pl. a következő lehet:

```
module ASM_Mealy(  
    input clk,  
    input rst,  
    input start,  
    input [7:0] Input,  
    output reg [3:0] Output,  
    output Done  
);  
  
parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10, s3 = 2'b11;  
reg [1:0] s;  
reg ld;  
reg [1:0] jmp_addr;  
reg [7:0] Data;  
wire DataLSB;  
assign DataLSB = Data[0];  
  
reg [7:0] Ocount; // számláló  
  
//állapotregiszter tölthető számlálóval  
always@(posedge clk)  
begin  
    if(rst) s <= s0;  
    else if(ld) s <= jmp_addr; //ugrási cím  
        else s <= s + 1;  
end
```

```

//tölthető számláló vezérlése
always@(*)
case(s)
s0: if(!start)
    begin
        jmp_addr <= s0;
        ld <= 1;           // ugrás
    end
    else
        begin
            ld <= 0;       // lépés
            jmp_addr <= 2'bxx;
        end
s1: begin
    ld <= 0;             // lépés
    jmp_addr <= 2'bxx;
end
s2: if(Data!= 8'b0)
    begin
        jmp_addr <= s2;
        ld <= 1;       // ugrás
    end
    else
        begin
            ld <= 0;       // lépés
            jmp_addr <= 2'bxx;
        end
s3: begin
    jmp_addr <= s0;
    ld <= 1;       // ugrás
end
default:
    begin
        jmp_addr <= s0;
        ld <= 1;
    end
endcase

```

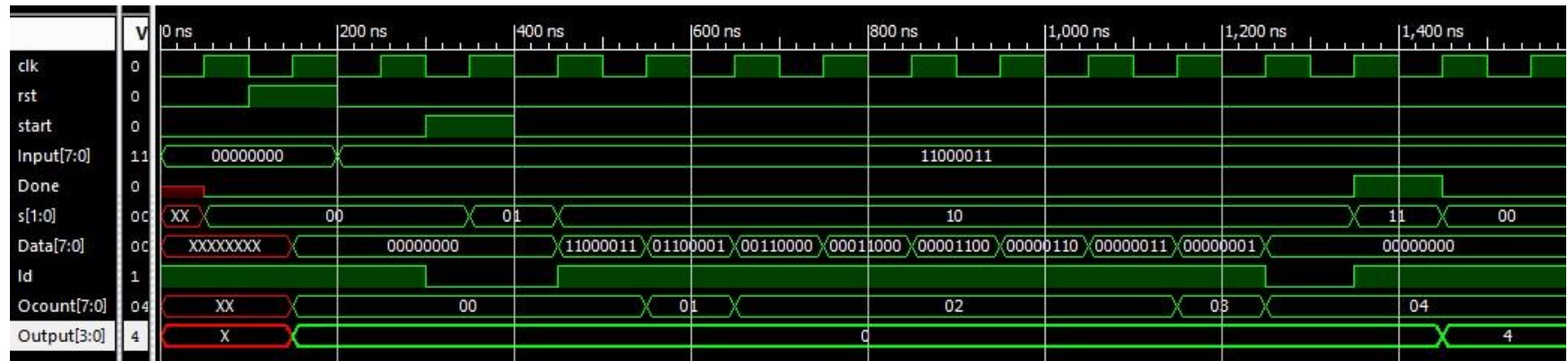
```
//shiftregiszter
always@(posedge clk)
begin
if(rst) Data <= 8'h00;
else if(s == s1) Data <= Input;
    else if((s == s2)& (Data != 8'h00))
        Data <= {1'b0,Data[7:1]};
end
```

```
// bitszámláló
always@(posedge clk)
begin
if(rst | (s == s1))
    Ocount <= 8'h00;
else if((s == s2)&DataLSB)
    Ocount <= Ocount + 1;
end
```

```
// kimeneti regiszter
always@(posedge clk)
begin
if(rst) Output <= 8'h00;
else if(s == s3)
    Output <= Ocount;
end
```

```
assign Done = (s == s3);
```

```
endmodule
```



Mikroprogramozott vezérlő

