

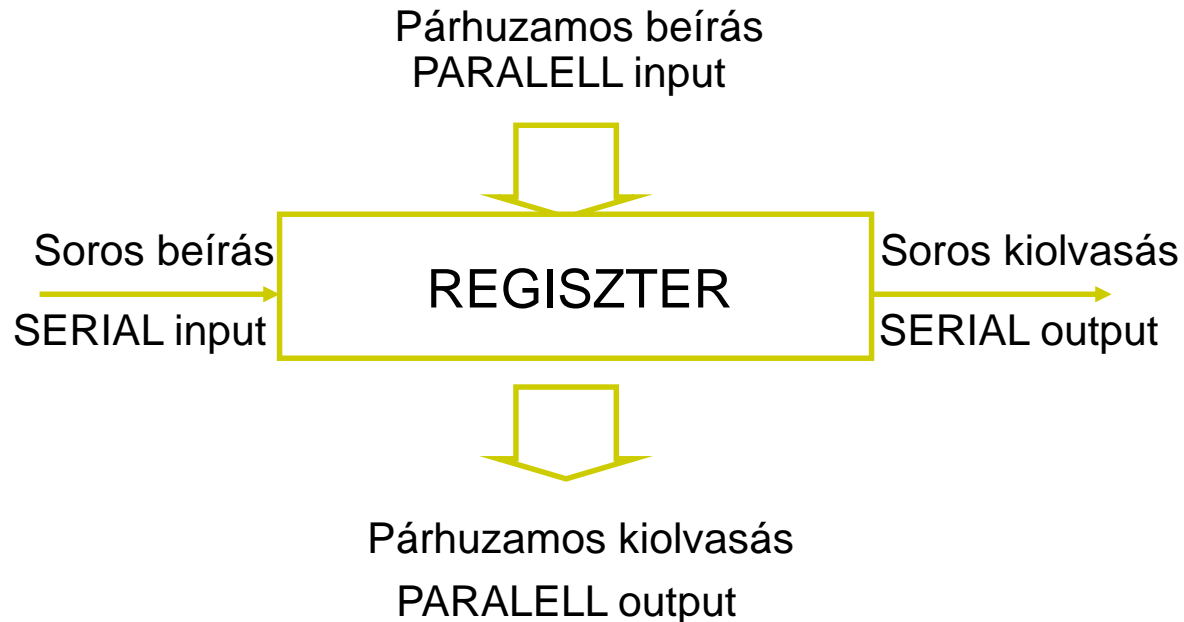
DIGITÁLIS TECHNIKA

9

Dr. Oniga István

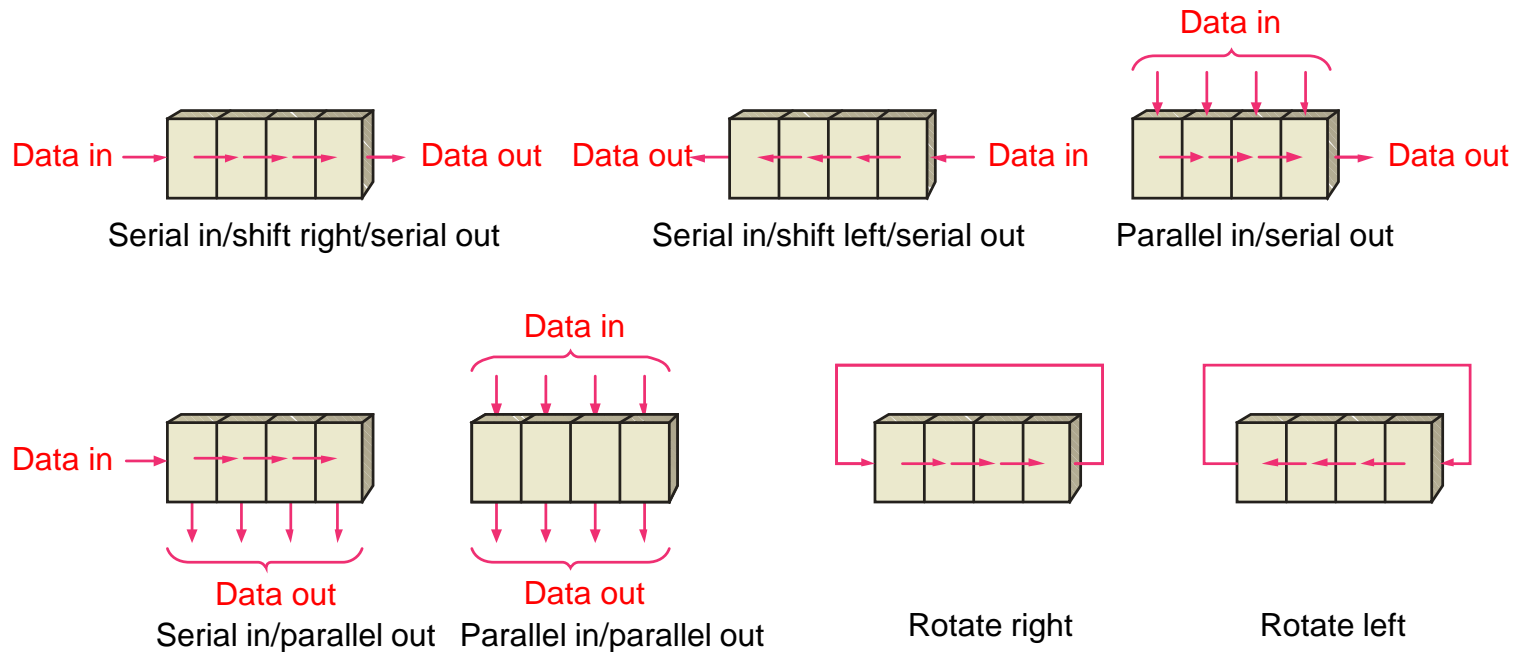
Regiszterek

- A regiszterek több bites tárolók hálózata
- S-R, J-K, D, vagy kapuzott D tárolókból készülnek
- Fontosabb alkalmazások: adatok tárolása és adatmozgatás
- Funkcióik:
 - adatok beírása
 - ♦ soros
 - ♦ párhuzamos
 - adatok tárolása
 - adatok kiolvasása
 - ♦ soros
 - ♦ párhuzamos



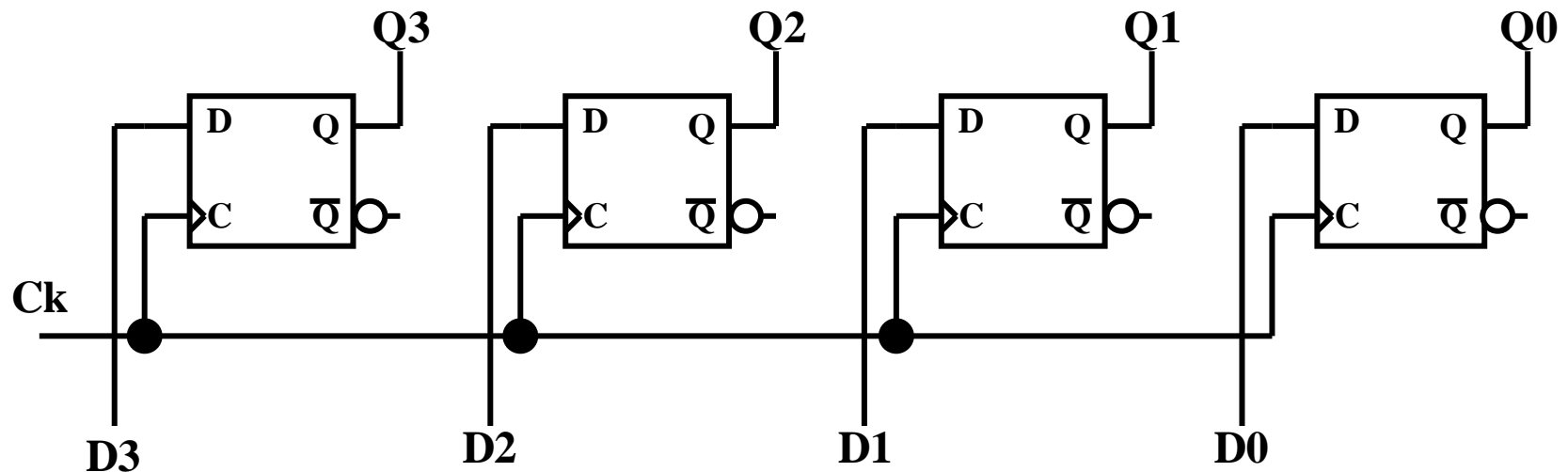
Regiszterek típusai

- A regiszterek főbb típusai:



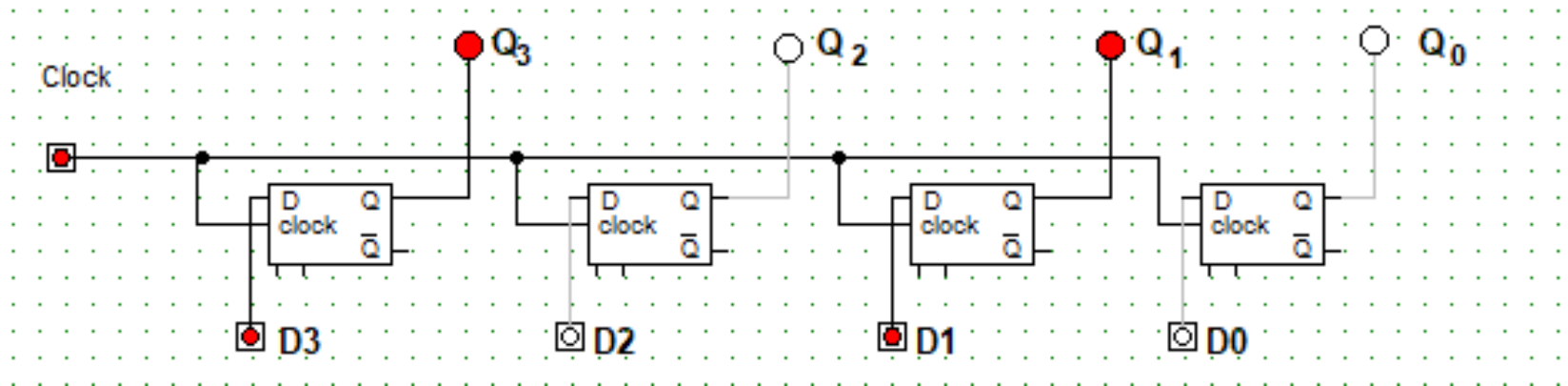
Puffer regiszterek

- Átmeneti tárolók
- Párhuzamos beírás
- Párhuzamos kiolvasás
- Kapuzott vagy élvezérelt tárolókat használunk
- Latch - kapuzott D tárolókkal



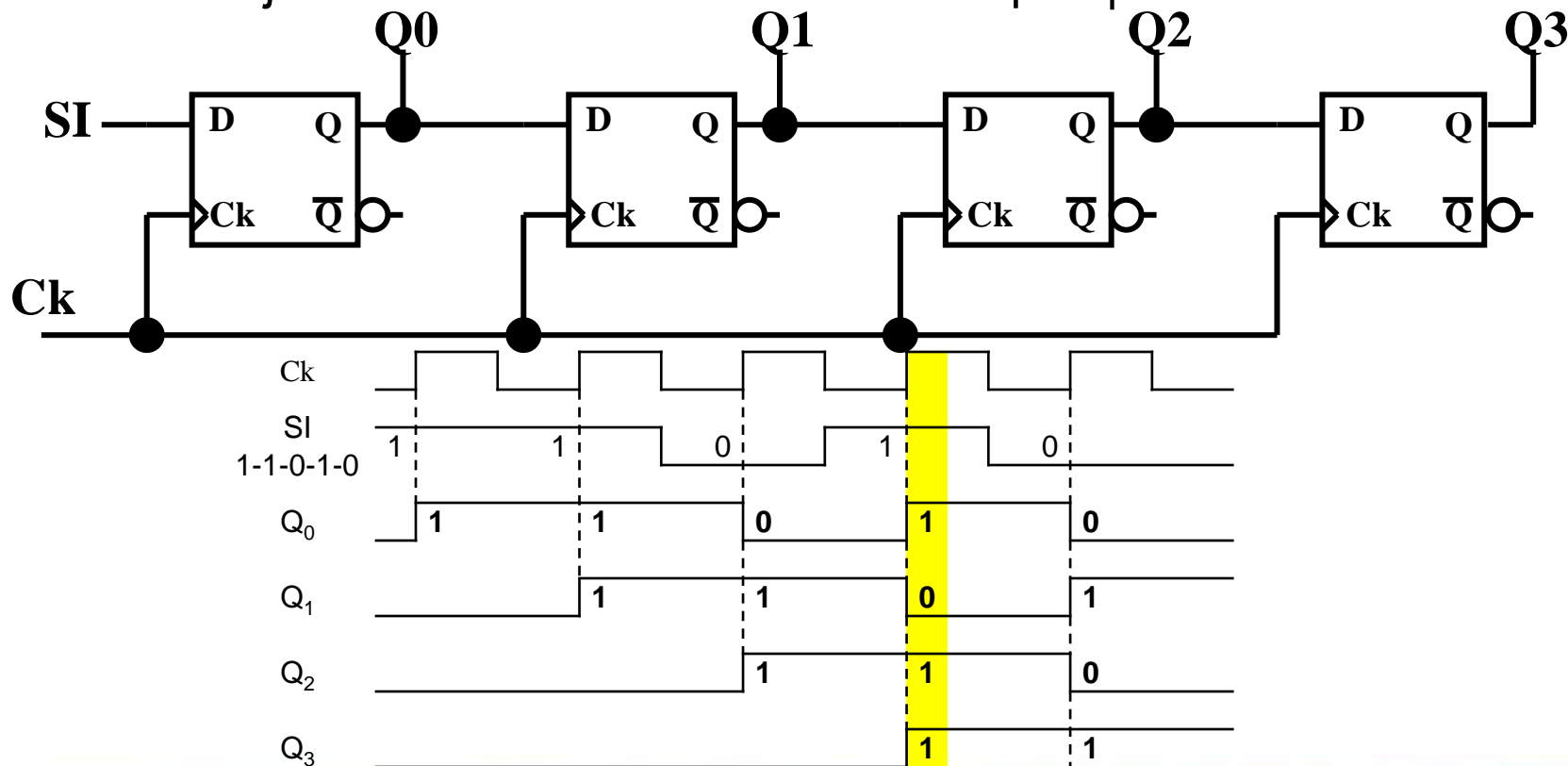
Puffer regiszterek

```
module p_register (input Ck, input [3:0] D, output reg [3:0] Q);  
  
    always @(posedge Ck)  
        begin  
            Q <= D;  
        end  
endmodule
```

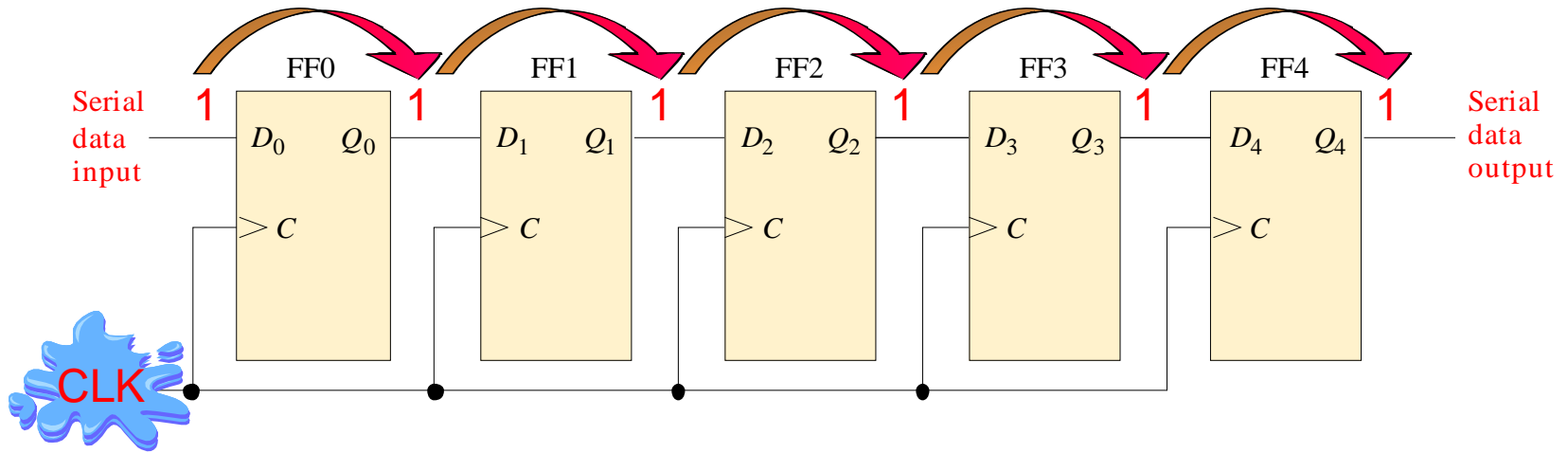


Shift regiszterek

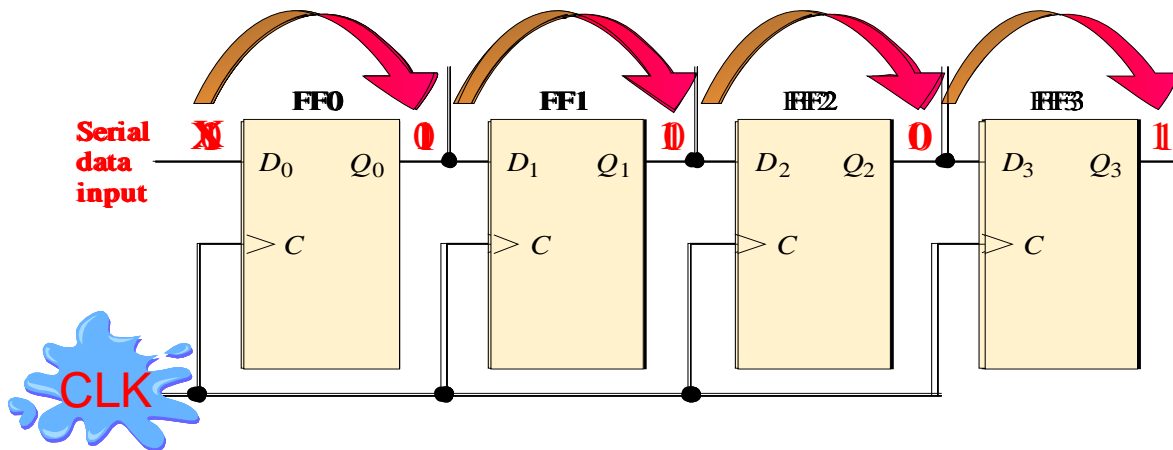
- Shift regiszterek vagy léptető regiszterek
- Soros be- és/vagy kimenet
- Minden órajel a bemeneti bitet a következő flip-flop-nak küldi



Shift regiszterek

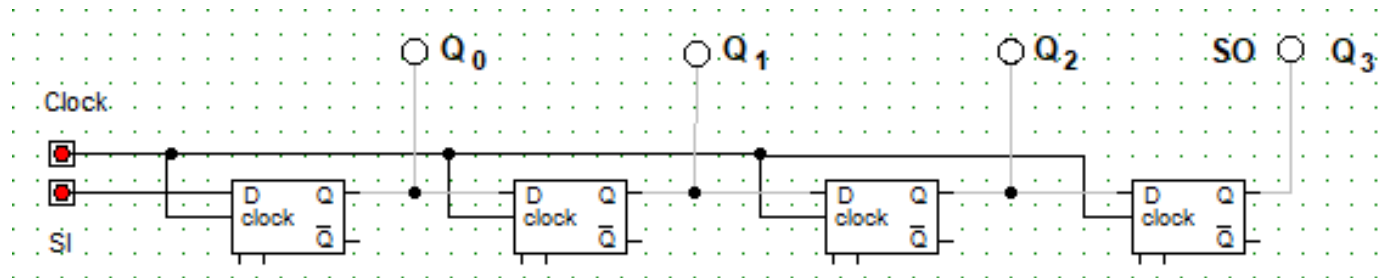


1011



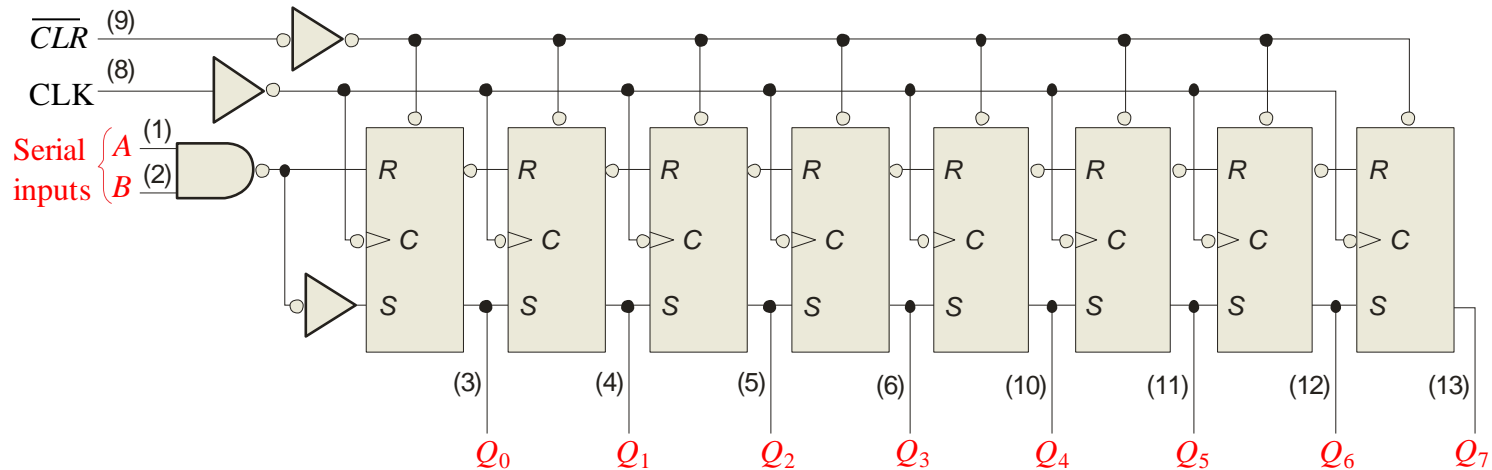
Shift regiszterek

```
module v_shift_registers_1 (input C, SI, output SO);  
  
  reg [3:0] tmp;  
  
  always @(posedge C)  
    begin  
      tmp <= tmp << 1;  
      tmp[0] <= SI;  
    end  
  
  assign SO = tmp[3];  
  
endmodule
```



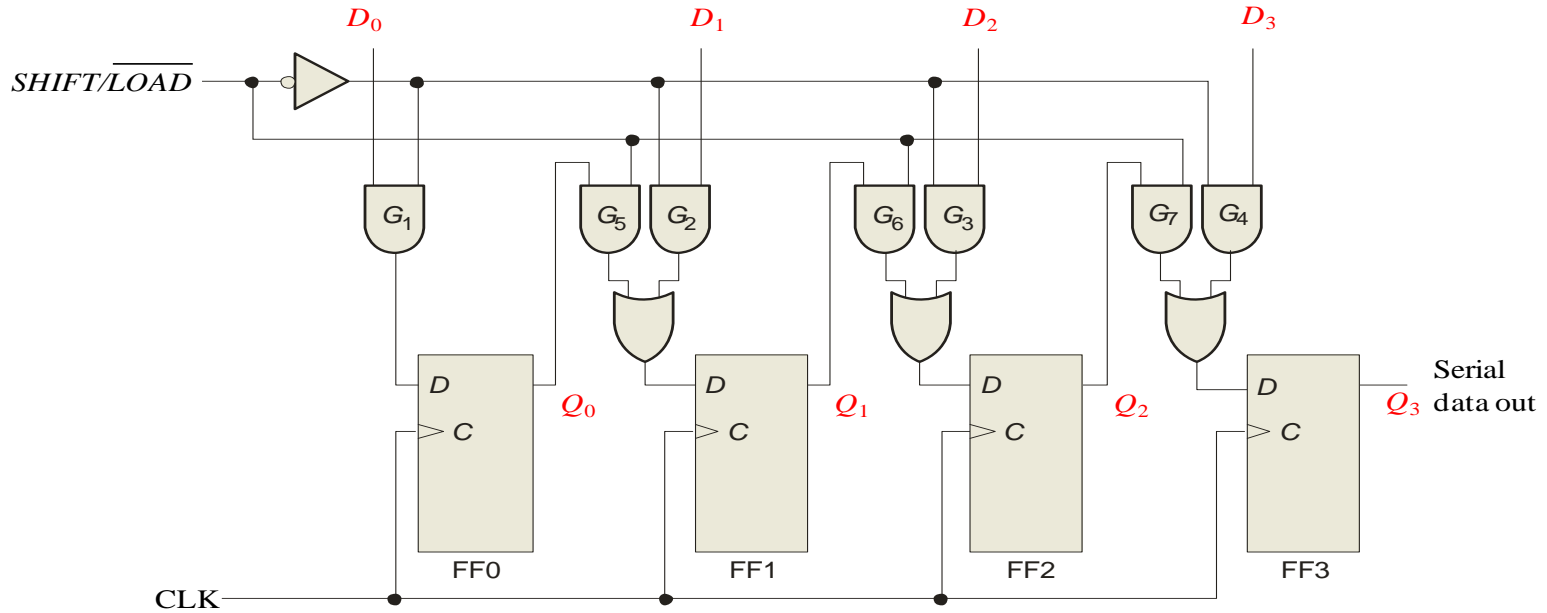
8 bites S-P shift regiszter

- Soros/párhuzamos átalakítás
- 74HC164A – CMOS 8 bites S-P shift regiszter
- ÉS-ezet bemenetek: A és B,
 - egyik az adat bemenet
 - a másik lehet
 - engedélyező bemenet
 - V_{CC}
- alacsony szinttel vezérelt aszinkron törlés

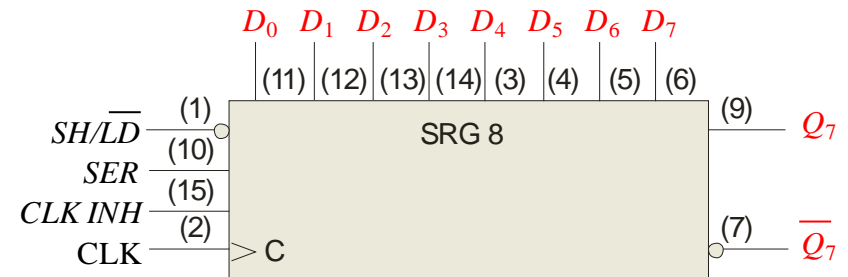


P-S shift regiszter

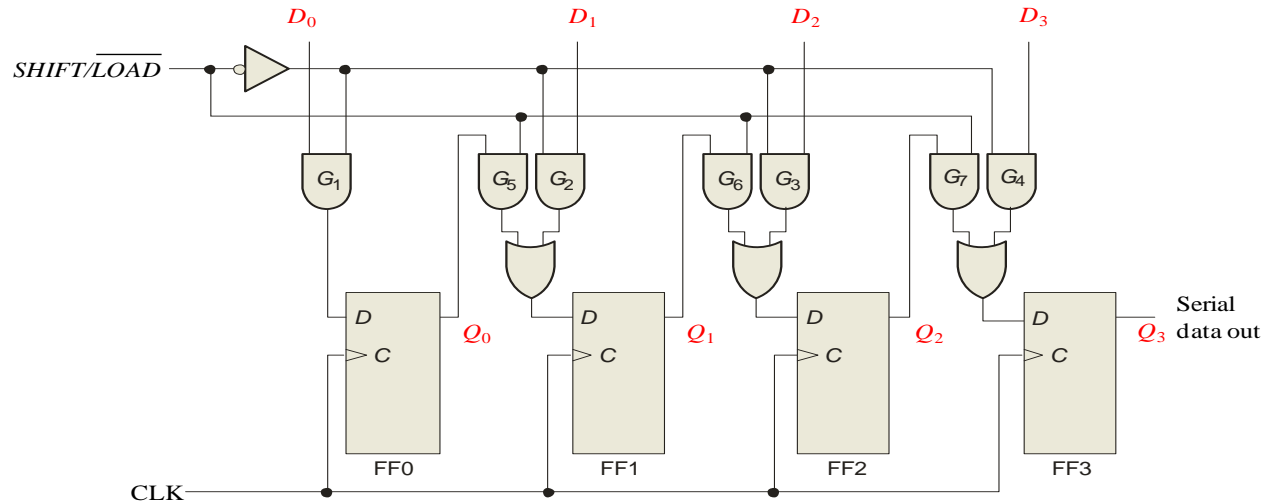
- A léptető regiszterek párhuzamos/soros adat átalakításhoz használhatók



- 74HC165A – CMOS 8 bites P-S shift regiszter
- $\overline{SH/LD}$ = LOW Aszinkron párhuzamos betöltés
- $\overline{SH/LD}$ = HIGH Szinkron léptetés



P-S shift regiszter

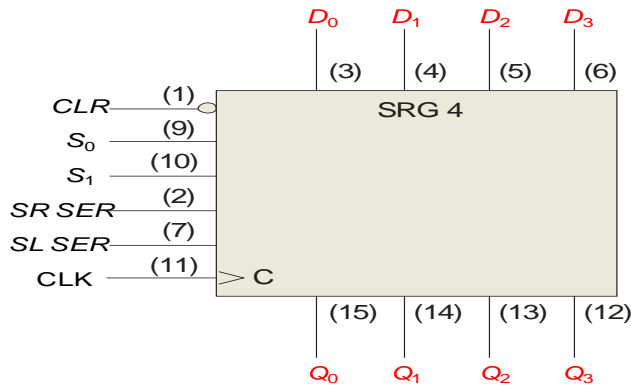


```
module v_shift_registers_ps (input CLK, SHIFT_LOAD, input
[3:0] D, output PO);
reg [3:0] Q
  always @(posedge C)
    begin
      if (SHIFT_LOAD ==1'b0)
        Q <= D;
      else
        Q <= {Q[2:0], SI};
      end
      assign SO = Q[3];
    endmodule
```

Univerzális shift regiszterek

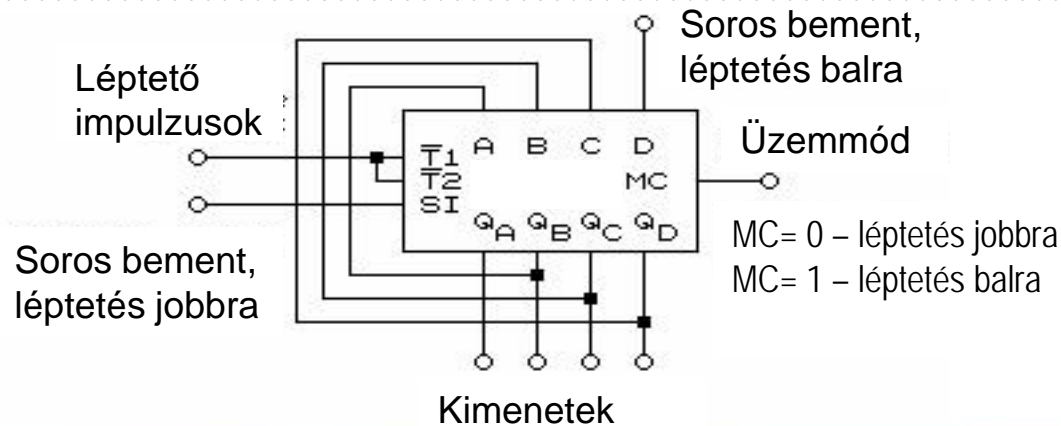
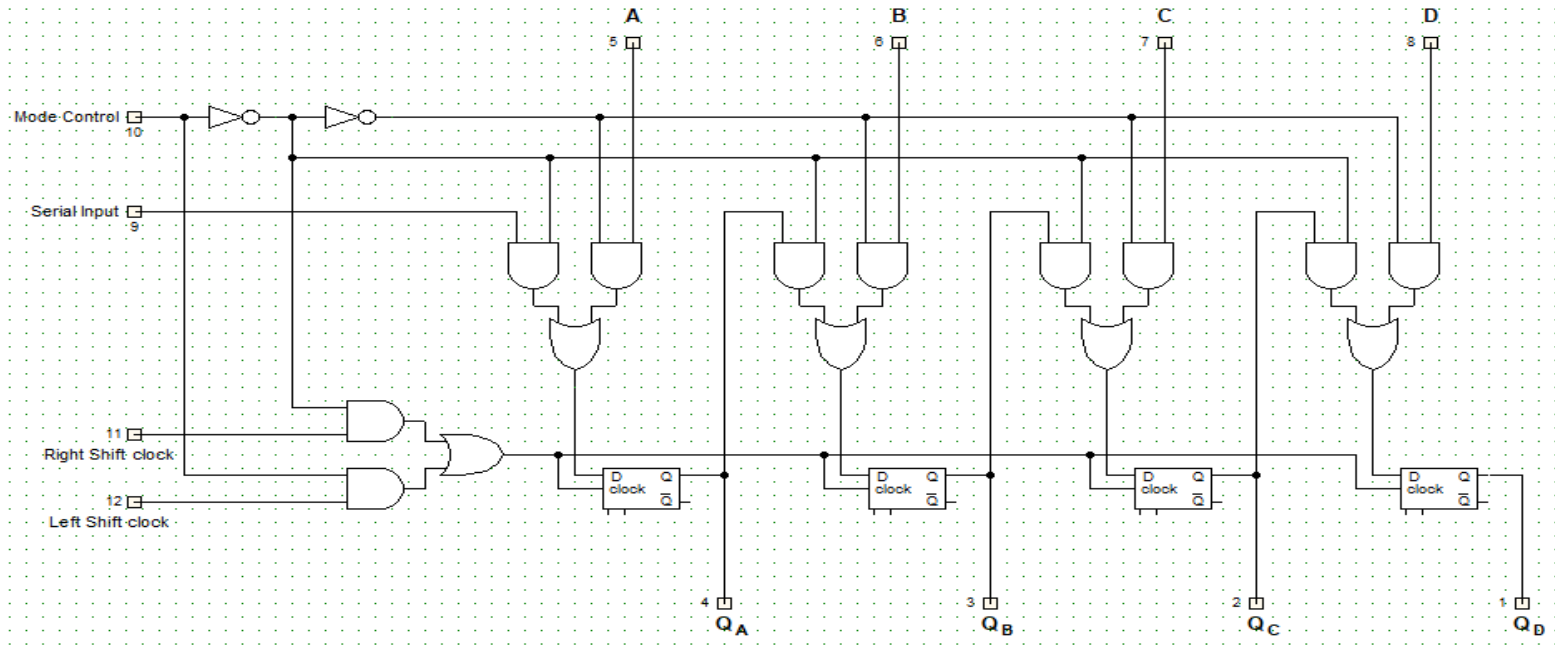
- Az univerzális shift regiszterek képesek
 - az adatok soros és párhuzamos fogadására,
 - párhuzamos megjelenítésére,
 - az adatok két irányban léptetésére
 - az adatok törlésére

- 74HC194A – CMOS univerzális shift regiszter



S1	S2	Üzem mód
0	0	szinkron törlés
0	1	léptetés balra
1	0	léptetés jobbra
1	1	párhuzamos beírás

SN7495 univerzális shift regiszterek



Univerzális shift regiszterek Verilog kódolása

```
//  
// 8-bit Shift-Left/Shift-Right Register with Positive-Edge Clock,  
// Serial In, and Parallel Out  
//  
  
module v_shift_registers_8 (C, SI, LEFT_RIGHT, PO);  
    input C,SI,LEFT_RIGHT;  
    output PO;  
    reg [7:0] Q;  
  
    always @(posedge C)  
    begin  
        if (LEFT_RIGHT==1'b0)  
            Q <= {Q[6:0], SI};  
        else  
            Q <= {SI, Q[7:1]};  
        end  
  
    assign PO = Q;  
  
endmodule
```

Gyűrűs számlálók

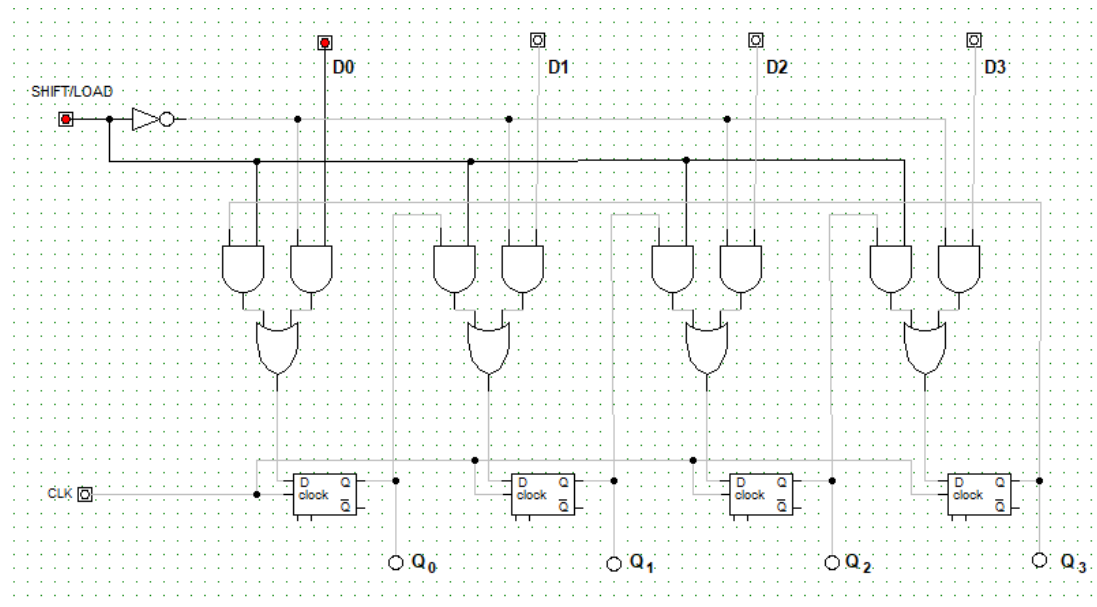
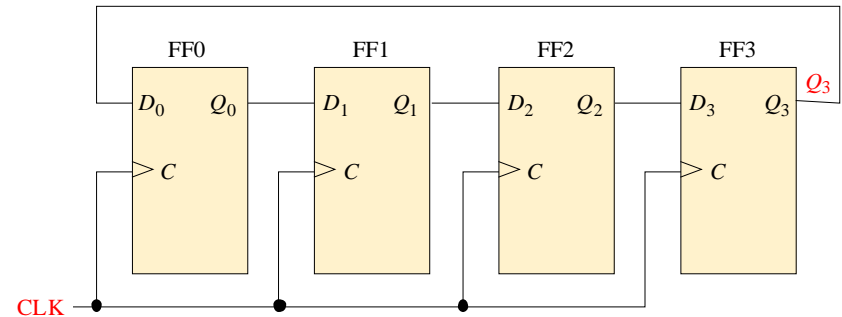
- Egyszerű visszacsatolással ellátott shift regiszterek
- N-ből 1 számláló
- Johnson számláló

N-ből 1 számláló

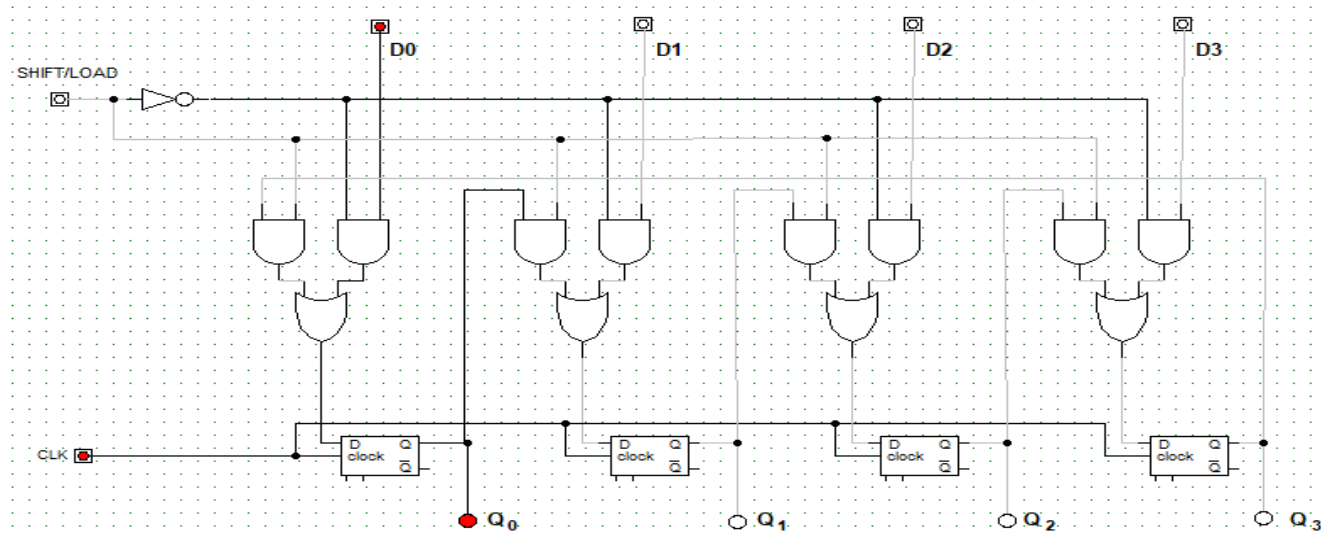
- utolsó kimenet visszacsatolva az első bementre

Órajel ciklus	Q3	Q2	Q1	Q0
alaphelyzet	1	0	0	0
1 órajel	0	0	0	1
2 órajel	0	0	1	0
3 órajel	0	1	0	0
4 órajel	1	0	0	0
	•	•	•	•
	•	•	•	•
	•	•	•	•

- n tároló n-ig számol ($\neq 2^n$)
- a kimenteket nem kell dekódolni
- szinkron számolás
- nagyobb számolási sebesség



Gyűrűs számlálók Verilog kódolása

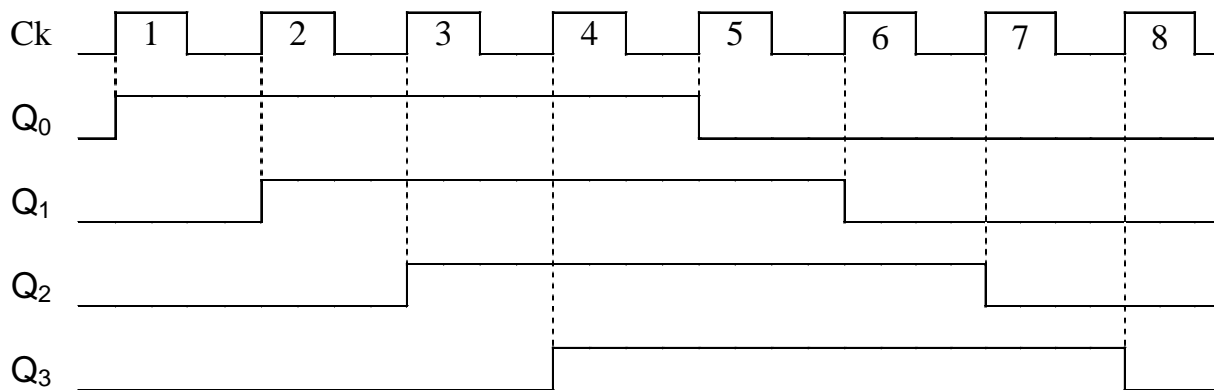
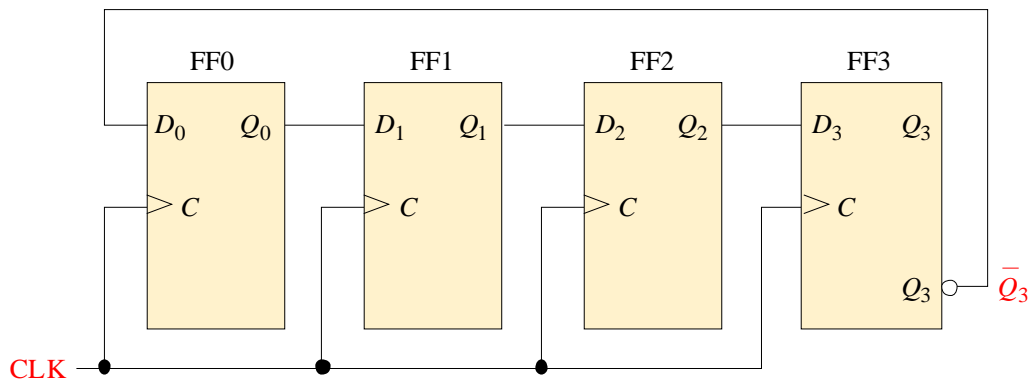


```
module Ring4(  
    input clk, load, en, input [3:0] d, output reg [3:0] q  
);  
  
always @ (posedge clk)  
    if (load==1'd0)      q <= d;  
    else  
        if (en) q <= {q[0], q[3:1]};  
  
endmodule
```


Johnson számláló

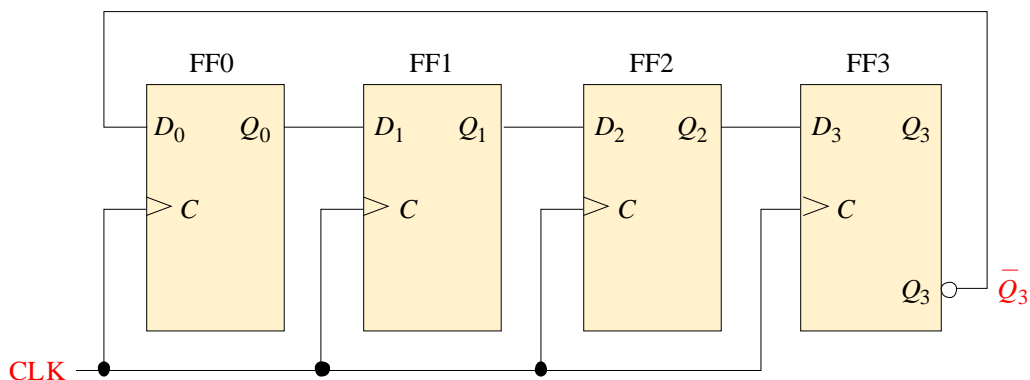
- utolsó kimenet negáltja visszacsatolva az első bementre
- n tároló $2n$ -ig számol ($\neq 2^n$)
- a kimenteket dekódolni kell

CLK	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

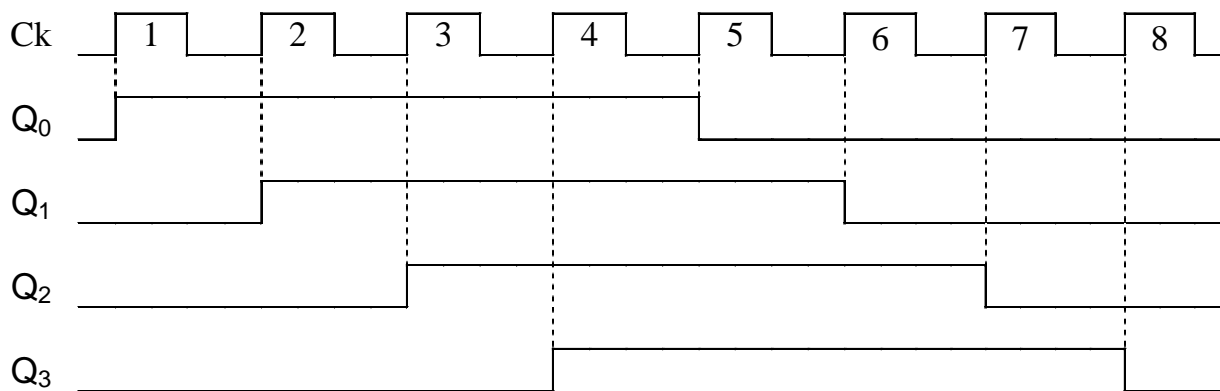


Johnson számláló

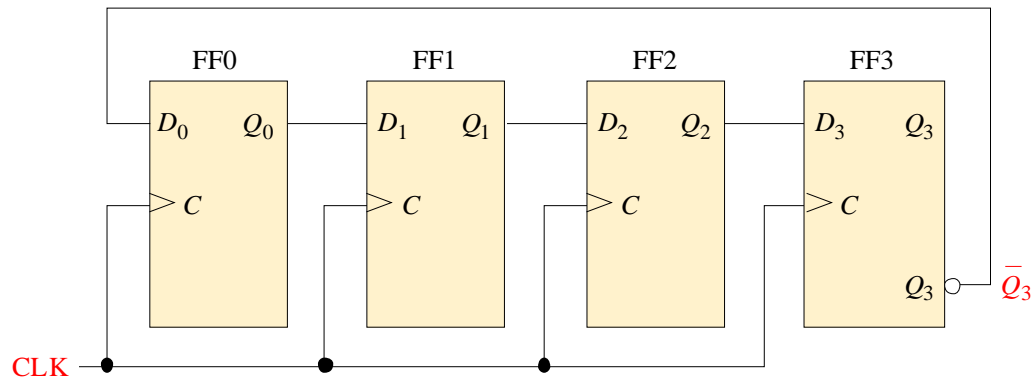
- utolsó kimenet negáltja visszacsatolva az első bementre
- n tároló $2n$ -ig számol ($\neq 2^n$)
- a kimenteket dekódolni kell



CLK	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

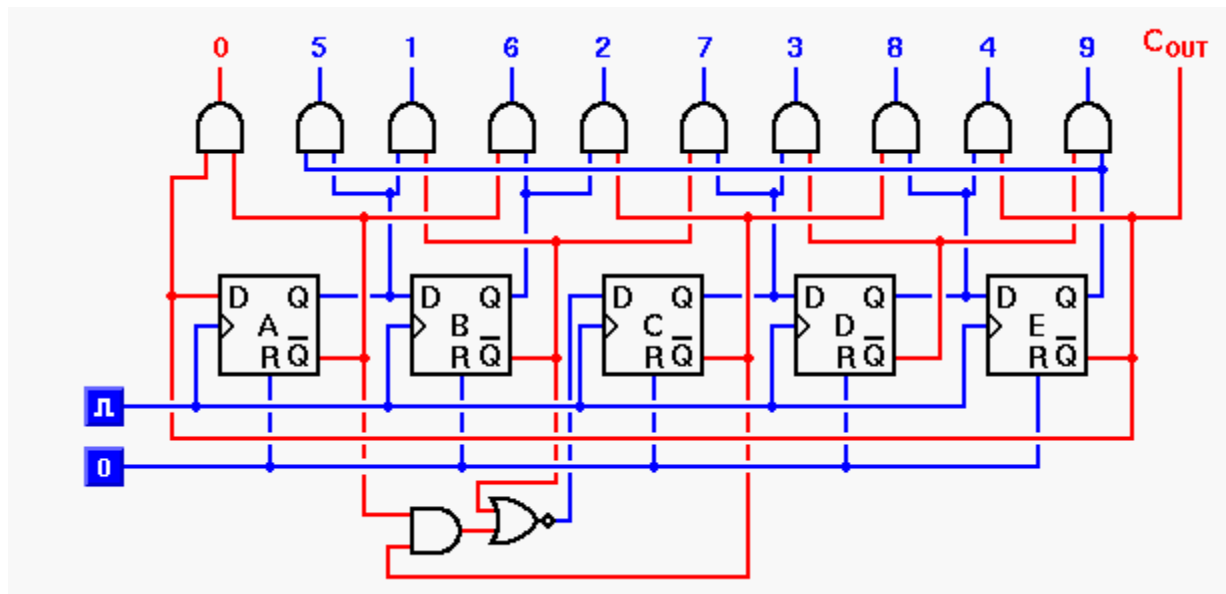


Johnson számláló Verilog kódolása



```
module Jhns4(input clk, output reg [3:0] q);  
  
    always @ (posedge clk)  
        q <= {~q[0], q[3:1]};  
  
endmodule
```

Johnson számláló



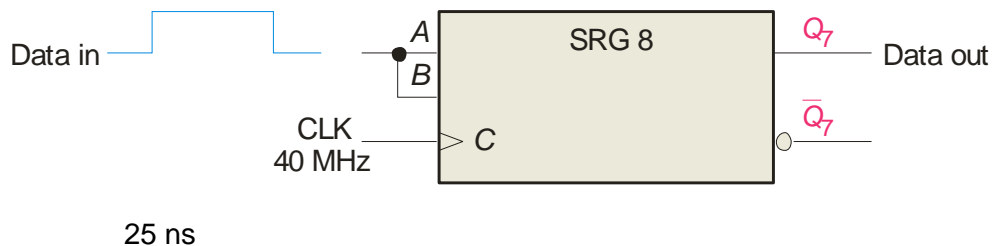
A	B	C	D	E	
0	0	0	0	0	0
1	0	0	0	0	1
1	1	0	0	0	2
1	1	1	0	0	3
1	1	1	1	0	4
1	1	1	1	1	5
0	1	1	1	1	6
0	0	1	1	1	7
0	0	0	1	1	8
0	0	0	0	1	9

http://www.play-hookey.com/digital/counters/johnson_counter.html

Shift regiszter alkalmazások

Digitális jelek késleltetése

Menyi az jel késleltetése egy 40MHz órajeles 8-bites soros be/ki menetű shift regiszteren.



Egy órajel késleltetése
 $1/40 \text{ MHz} = 25 \text{ ns}$

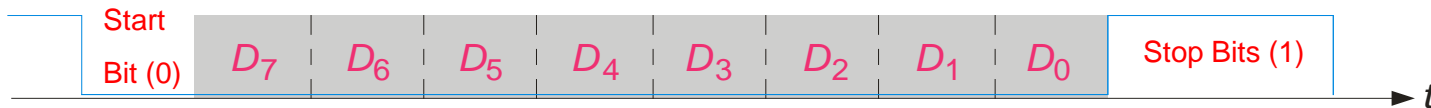
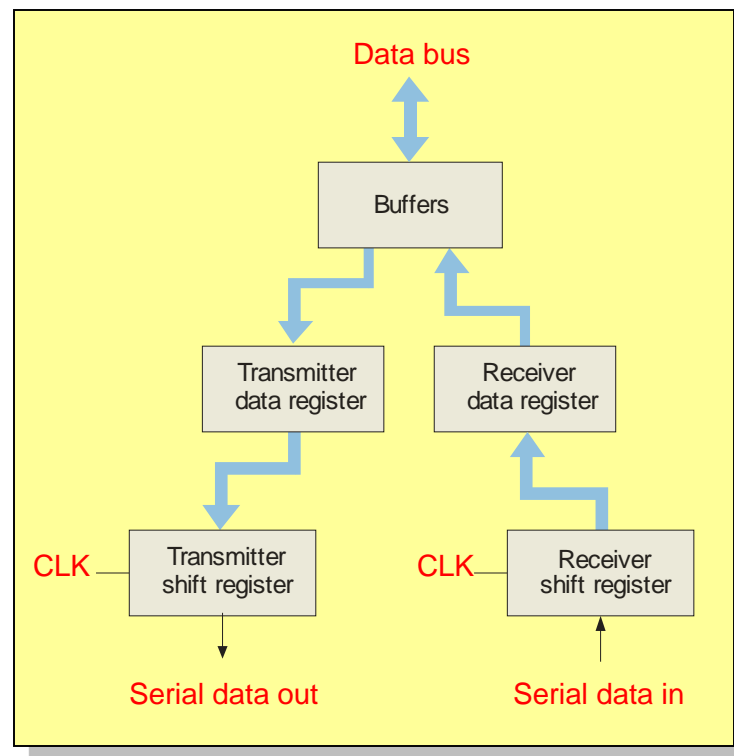
Összesítet
késleltetés 8×25
ns = **200 ns**

= 200 ns

Shift regiszter alkalmazások

UART (Universal Asynchronous Receiver Transmitter)

- egy soros-párhuzamos átalakító
- egy párhuzamos-soros átalakító



Billentyűzet encoder

- Gyűrűs számláló.

Két 74HC195 shift regiszter - 8-bit gyűrűs számláló, alaphelyzetbe egy 0 betöltve.

