

Egy egybemenetű sorrendi automata feladata, hogy a bemenetére sorosan érkező 3 bites számokról eldöntse, megegyeznek-e egy megadott számmal, s a végén jelezzen. N=001

Kapcsolás Moore modell szerint
Verilog leírás Mealy modellhez

Állapottábla: (Moore)

y\X	0	1	
A	B,0	C,0	Kezdés
B	D,0	F,0	0 benn
C	F,0	F,0	1 benn, ez már nem lehet jó, 2 bitet még azért várunk
D	A,0	E,0	00 benn
E	B,1	C,1	Z=1 (virtuális A állapot ez)
F	A,0	A,0	Nem jó, 1 bitet még várunk azért, majd A állapot

Végezzük el az állapotkódolást bármiféle megfontolás nélkül, például rendeljük a kódokat növekvő sorrendben az állapotokhoz. (Szinkron hálózatot építünk, tehát nincs jelentősége)

	y1	y2	y3
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	0
E	1	0	0
F	1	0	1

Írjuk fel a kódolt állapottáblát. Ez nevetséges behelyettesítés csupán.

y\X	0	1
000	001,0	010,0
001	011,0	101,0
010	101,0	101,0
011	000,0	100,0
100	001,1	010,1
101	000,0	000,0

Használjunk mondjuk D flip-flopokat, mivel annak nagyon egyszerű a vezérlési táblája.

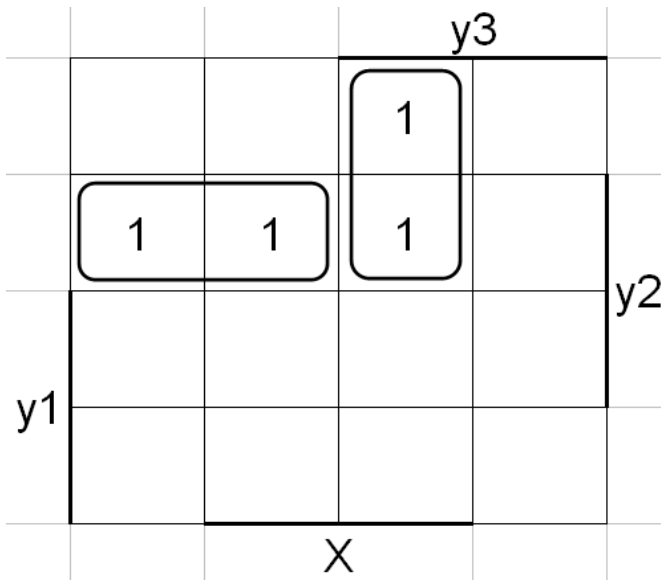
váltás	D
0→0	0
0→1	1
1→1	1
1→0	0

Látható, hogy a kimenet nem függ a megelőző állapottól, hanem mindig csak a Data-tól.

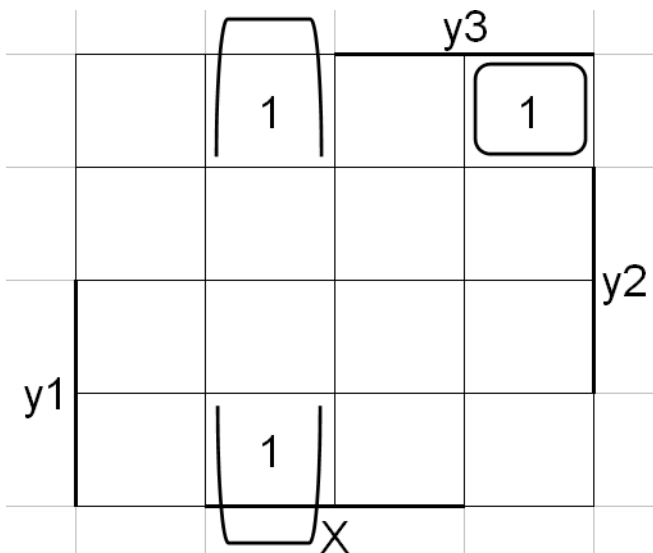
Írjuk fel elsőnek a kimenetet. Ez könnyű lesz, mivel egyetlen $Z=1$ állapotunk van. Nyilvánvaló, hogy a kimenet nem függhet X -től, hisz ez a Moore modell lényege.

$$Z = y1 \cdot \overline{y2} \cdot \overline{y3}$$

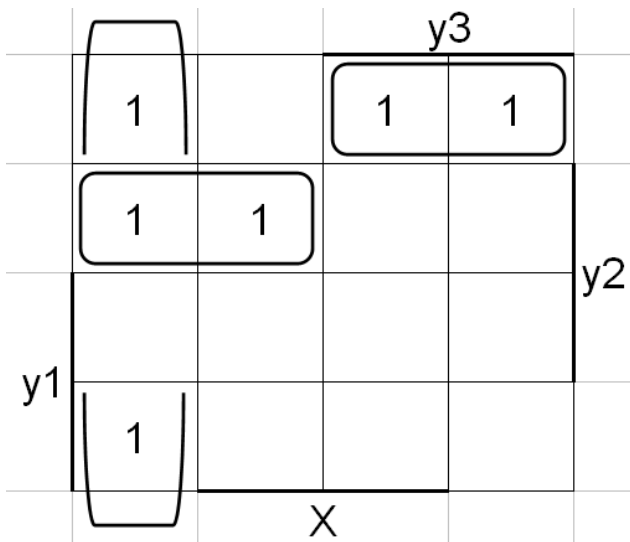
Találjuk ki a D flip-flopok vezérléseit. Tervezzünk mondjuk simán AND-OR összeállításban. Ekkor az egyeseket kell majd tömbösíteni a V-K táblán, és a változókat normálisan kell figyelembe venni (amelyik ponáltan szerepel, azt ponáltan..)



$$D1 = \overline{y1} \cdot y2 \cdot \overline{y3} + \overline{y1} \cdot y3 \cdot X$$



$$D2 = \overline{y2} \cdot y3 \cdot X + y1 \cdot \overline{y2} \cdot y3 \cdot \overline{X}$$

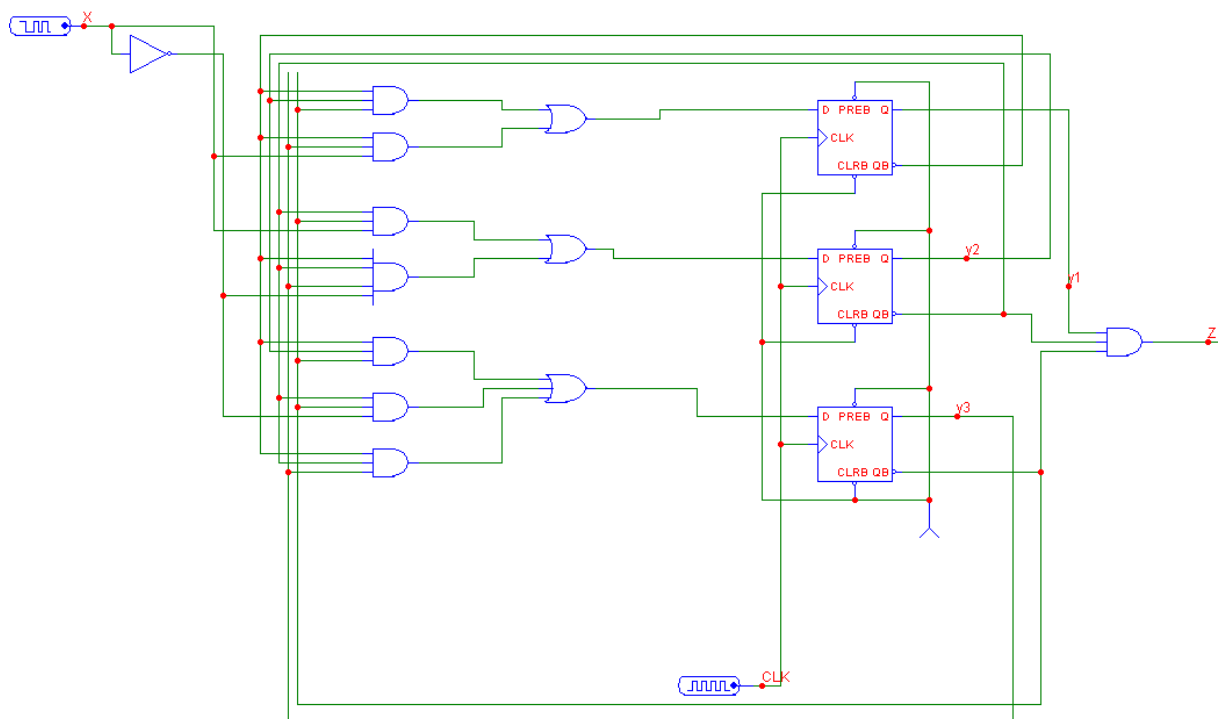


$$D3 = \overline{y1} \cdot \overline{y2} \cdot y3 + \overline{y1} \cdot y2 \cdot \overline{y3} + \overline{y2} \cdot \overline{y3} \cdot \overline{X}$$

(Mivel szinkron hálózatot építünk, statikus hazardokkal nem kell foglalkoznunk, nem lesz hatásuk)

A Karnaugh táblák felrajzolása nagyon egyszerű, mivel D flip-flopokat használunk. Például a B állapotnál (kódja 001) azt látjuk, X=1 bemenet esetén az új állapotkód 1. bitje (y1) 1-es (az új állapotkód 101). Vagyis 001 jelenlegi állapot esetén, és ha a bemenet 1, akkor y1 1-be kell váltson. Azaz D1-hez tartozó Karnaugh táblán $\overline{y1}y2y3X$ -hez beírhatunk egy 1-est.

Építsük meg a kapcsolást például a Microcap 10 Working demo / Student verziójában.
<http://www.spectrum-soft.com>

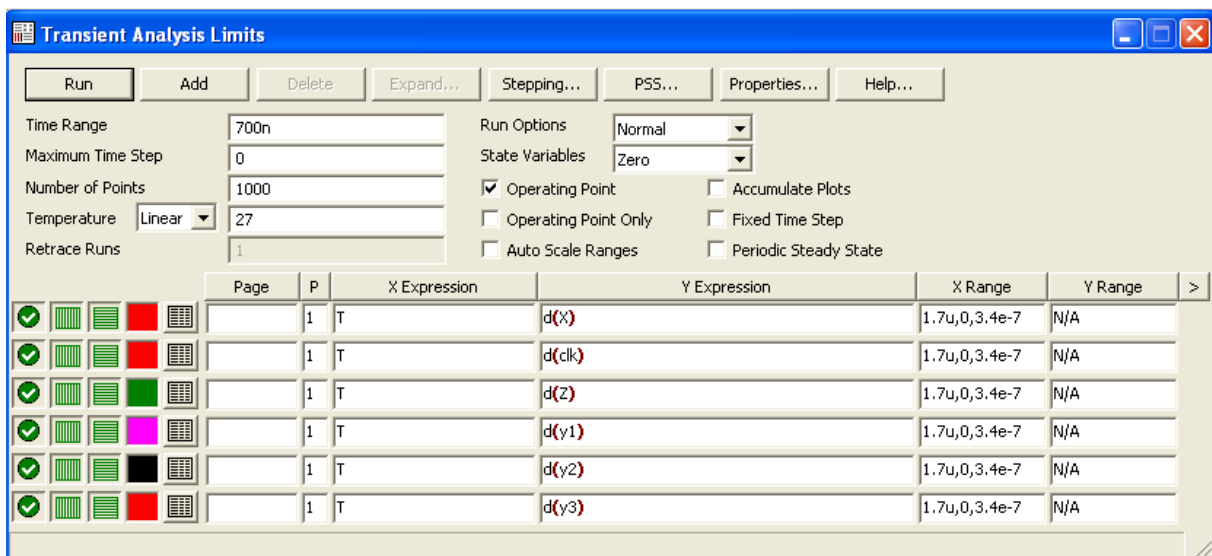


A X-et előállító stimulus generátor programozása legyen olyan, hogy a teljes állapottáblát bejárjuk. Például az első sorozat legyen 001 (ezt el kéne fogadnia az automatának), majd jöjjön pl 111 (ezt el kéne utasítania).

```
.define Vizsgjel1
+0n 0
+100n 0
+200n 1
```

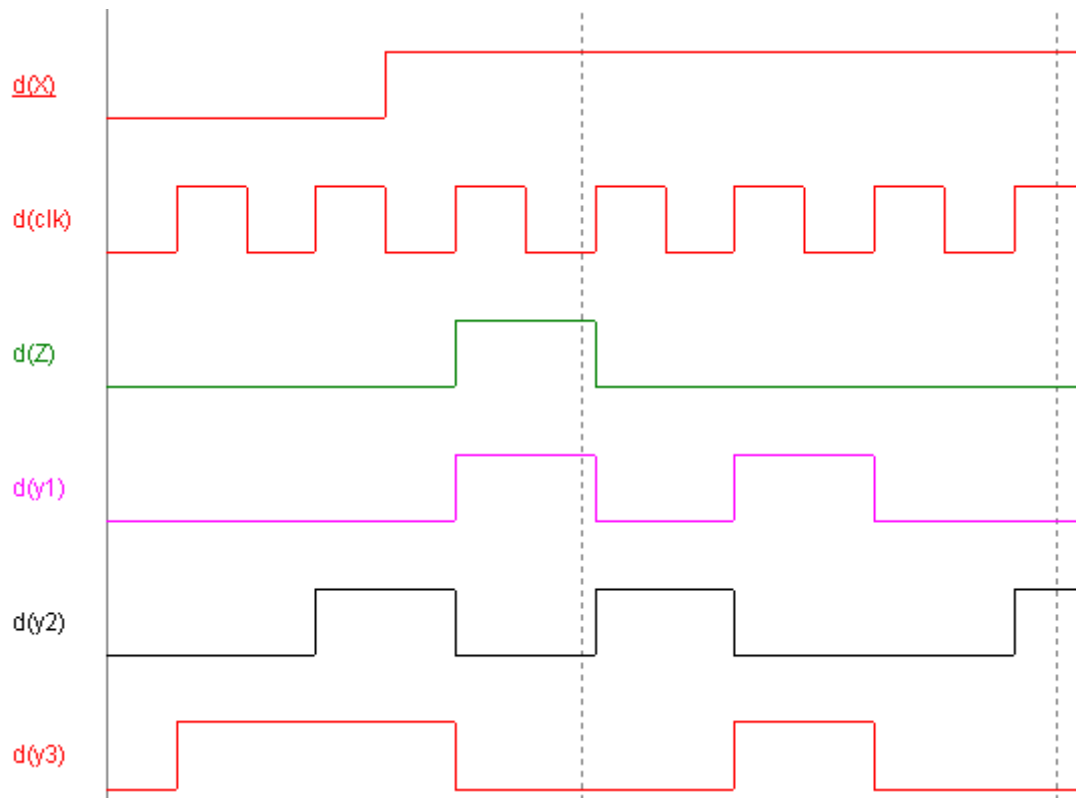
Így 200n után folyamatosan 1 lesz a bemenet, ez jó nekünk. (A CLK olyan, hogy 100ns-enként jön felfutó él).

A sokcsatornás oszcilloszkópon jelenítsünk meg mindent. A vizsgálat tartson mondjuk 700ns-ig, így minden ki fog férni.



Futtassuk le.

(Mivel a D flip-flop olyan, hogy a kimenet nem függ a megelőző állapotoktól, RESET-ekkel nem kell fárasztani magunkat).



Látható, hogy a hálózat a 001 bemeneti sorozatot elfogadja, a végén $Z=1$ -el jelez. A második sorozatot (111) az automata elutasítja. Érdekes még megfigyelni, hogy a 6. felfutó órajelre a visszacsatoló ágak $y1=y2=y3=0$ állapotba jutnak, mely az A állapotnak felel meg, vagyis a hálózat várja a 3. bemeneti sorozatot. Aki akarja, a többi állapotot is végigkövetheti.

Hozzuk létre a Mealy modell szerint működő állapottáblát is.

$y \backslash X$	0	1	
A	B,0	C,0	Kezdés
B	D,0	F,0	0 benn
C	F,0	F,0	1 benn, ez már nem lehet jó, 2 bitet még azért várunk
D	A,0	A,1	00 benn, itt döntünk a kimenetről
F	A,0	A,0	Nem jó, 1 bitet még várunk azért, majd A állapot

Ugyanaz, mint a Moore modell, de az E állapot felesleges, hanem a D állapotban döntjük el (itt 00 van már benn), hogy a kimenet 0 vagy 1 lesz, aszerint, hogy mi a 3. bit.

A Verilog kód megírása ez alapján szinte technikumi favágás.

```

module Automata
(
    input X,
    input clk,
    input res,
    input enable,
    output wire Z,
    output reg[2:0] state
);
parameter A = 3'b000;
parameter B = 3'b001;
parameter C = 3'b010;
parameter D = 3'b011;
parameter F = 3'b101;

assign Z = ((state == D & X);

always @(posedge clk or posedge res)
    begin if (res)                state <= A;
          else
            if (enable)
                begin
                    case (state)
                    A:    if ( !X)                state <= B;
                          else                    state <= C;

                    B:    if ( !X)                state <= D;
                          else                    state <= F;

                    C:    state <= F;

                    D:    state <= A;

                    F:    state <= A;

                    default:                state <= A;
                    endcase
                end
            end
    end
endmodule

```

A házik ilyen szintű kidolgozása nagyon sok időt és figyelmet kíván. Aki elakad, annak azt javaslom, főként az állapotáblára és az abból kreált Verilog kódra összpontosítson, mivel a laboron azt kell használni. Ha egyikkötőknek sincs működő kódja, az elég rossz, mert semmit sem tudtok majd csinálni.