Laboratórium 1 felkészülési feladat

Hallgató: Vuchetich Bálint (A9T953)
Mérés sorszáma: 10

Egy 4 < n < 8 állapotú, egyetlen X bemenettel rendelkező sorrendi automata a következőképpen működik. Bekapcsolás után a kezdőállapotban marad mindaddig, amíg X=0 és itt Z=0-át ad. Ha X 1-re változik, akkor kilép a kezdő állapotból, majd X-től függetlenül minden órajelre újabb állaptba lép, és a következő állapotokban egy n-1 hosszúságú kimeneti sorozat következő elemét adja a Z kimenetén. Ezt mindaddig folytatja, míg az n-edik állapotba nem lép. Az n-edik állapotban a sorozat utolsó elemét adja és X=1 esetén itt marad. X=0 esetén az n-edik állapotból a kezdőállapotba lép. A kimeneti sorozat nem lehet konstans és az utolsó bitje mindig 1.
Példa n=5 és y=1101:
X:001xxx11110001x
Z:000110111110001

Készítse el a fent leírt funkciót megvalósító Verilog modult **n = 5, y = 0111** esetén!
A mérés kezdetén be kell mutatni a működő Verilog kódot és a működést igazoló szimulációt. A modulnak legyen órajel engedélyező bemenete is! (Ez a tesztelhetőség szempontjából fontos.)

Kötelező kritériumok a beadandó anyagra: Ezek be nem tartása esetén a feladtot nem fogadjuk el!
Hozza el a teljes működő ISE projektet! Készítsen egy word (vagy pdf) dokumentumot, amiben szövegesen összefoglalja a kód felépítését, működését! Nem elég csupán a forráskódot bemásolni!
E dokumentumnak kötelezően tartalmaznia kell a működést igazoló szimulációs hullámformát és annak szöveges értelmezését. Az értelmezés nélküli képeket nem értékeljük, nem fogadjuk el.
A fájl kinyomtatása nem szükséges, ennél a mérésnél elektronikus dokumentációt kérünk.

A beadás tudnivalói:

- **Az önállóan kidolgozott feladatot a következő mérési gyakorlat elején a mérésvezetőnek kell bemutatni. A feladatot a 4. ... 9. mérések esetén kézzel írott, a 10. és 11. mérések esetén elektronikus formában kell beadni.**
- A felkészülési feladat utólag már nem adható be. Pótlására a szorgalmi időszak végén egy alkalommal, az adott mérési gyakorlat pótlásával egy időben van lehetőség.

A feladatokat önállóan, meg nem engedett segítség igénybevétele nélkül oldottam meg:

.....
aláírás

A működést leíró modul kódja, és kommentben a magyarázatok.

Környezet: Ubuntu 16.04 LTE, VMware Workstation 12 Player-rel virtualizált (tanszéki) MS Windows XP-n futtatott ISE Project Navigator.

File→ New project:
név megadása, next, next, finish.
(Implementation view) Project → New source:
New verilog module, a be- és kimentek definiálása, next, next, finish.

A működés implementálása.

```
`timescale 1ns / 1ps


// modul be- es kimeneteinek definialasa
module main(
	// jel bemenet
	input x,
	// engedelyezo bemenet
	input allow,
	// orajel bemenet
	input clk,
	// jel kimente
	output z
);

// az allapot eltarolasara szolgalo regiszter
reg [2:0] state;
// a kimenetet tarolo regiszter
reg rout;


// a kezdeti ertekek beallitasa
initial begin
state <= 0;
rout <= 0;
end


// a modul mukodesenek leirasa
always @(posedge clk)
if(allow)
```

```
    case(state)
        0: begin
                rout <= 0;
                if(x)
                    state <= 1;
            end
        1: begin
            rout <= 0;
            state <= 2;
        end
        2: begin
            rout <= 1;
            state <= 3;
        end
        3: begin
            rout <= 1;
            state <= 4;
        end
        4: begin
            if(x || rout)
            rout <= 1;
            if(x==0)
                begin
                    rout <= 0;
                    state <= 0;
                end
        end
    endcase

// kimeneti regiszter erteket rakotjuk a kimenetre
assign z = rout;

endmodule
```

A tesztelést leíró kód, és kommentben a magyarázatok.

Átváltunk Simulation view-ra, Project → New source:
Verilog Text Fixture, beállítjuk az egyetlen (main) modulra, next, next, finish.

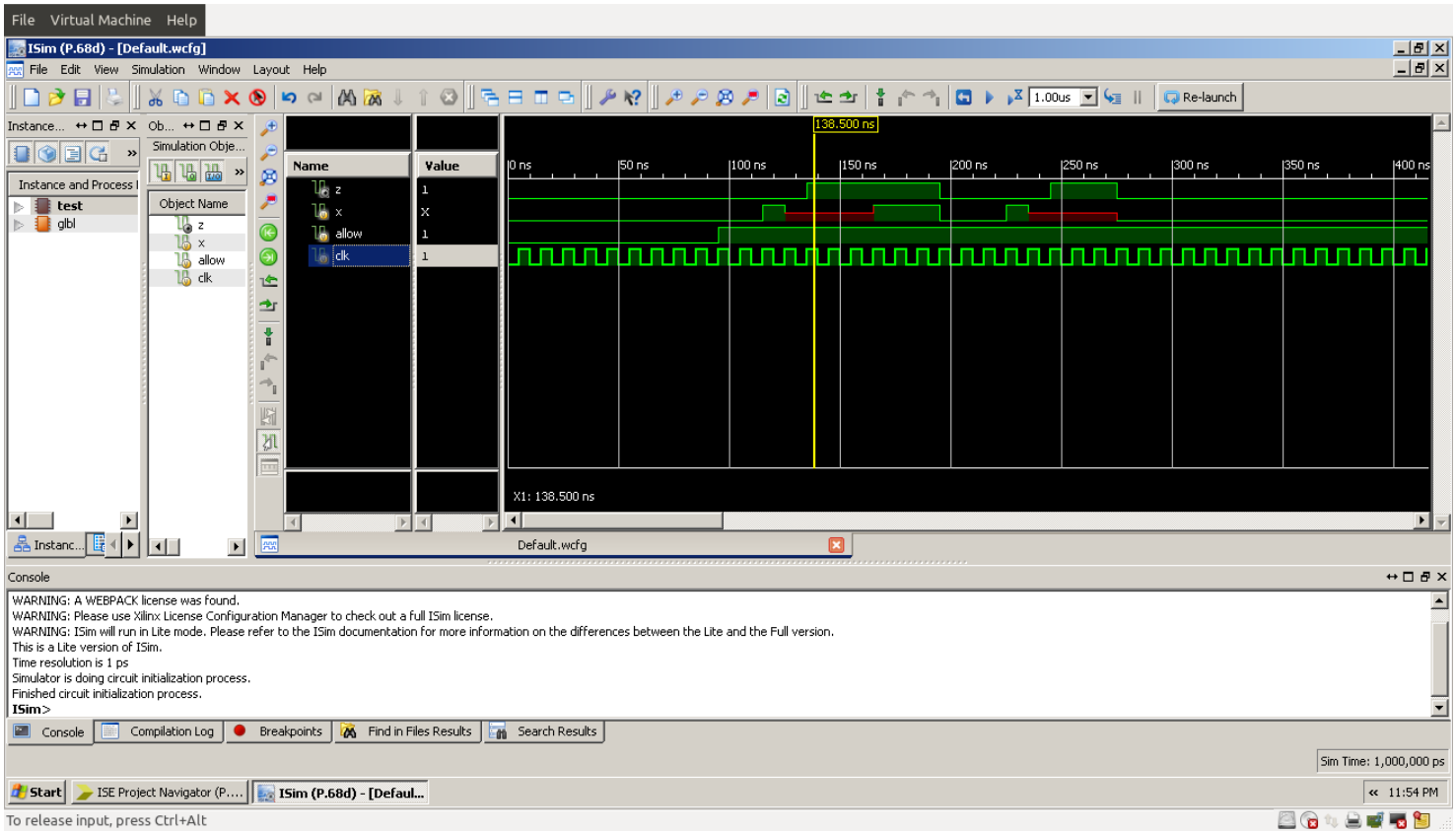
Implementáljuk a tesztelést leíró kódot.

```
`timescale 1ns / 1ps

module test;
	reg x;
	reg allow;
	reg clk;
	wire z;

	main uut (
		.x(x),
		.allow(allow),
		.clk(clk),
		.z(z)
	);

	initial begin
		//beallitjuk a kezdeti ertekeket
		x = 0;
		allow = 0;
		clk = 0;

		// 95 ns utan raadjuk az engedelyezo jelet, a bemeneten meg tartjuk a 0 jelet
		#95;
		allow = 1;
		x <= 0;

		// 20ns utan a bemenetre 1-et adunk
		#20
		x <= 1;

		// ezt egy orajel hosszusagig tartjuk, majd DC allapotba allitjuk
		#10
		x <= 1'bx;

		// 4 orajel mulva 1-re allitjuk a bemenetet, igy a rendszerunk tartani fogja az utolso allapotat
		#40
```

```
        x <= 1;

        // 3 orajel utan 0-ra allitjuk a bemenetet
        #30
        x <= 0;

        // 3 orajellel kesobb ismet 1-re allitjuk a bemenetet, a halozat ujra valaszol
        #30
        x <= 1;

        // megtortent a beallitas, DC allapotra allitjuk a bementunket
        #10
        x <= 1'bx;

        // 4 orajellel kesobb 0-ra allitjuk a bemenetet, most nem kell tartania az
utolso allapotot
        #40
        x <= 0;
    end

    // 5ns-onkent invertaljuk az orajelet, igy lesz a periodusido 10ns
    always #5
    clk <= ~clk;

endmodule
```

A szimulációs ábra, és értelmezése

Simulation view-ban a test modult kiválasztva rákattintunk a ISim Simulator-ban a Simulate Behavior Model-re. Kicsit átállítjuk a zoomolást, hogy egyben lássuk a folyamatot.



Értelmezés.

Periódus (ns)	Működés
0 – 95	Az engedélyező jel 0-n van, a kimenet is 0.
95 – 115	Az engedélyező jel 1-re vált, a bemenet még mindig 0, ezért a rendszer 0 kimenettel vár.
115 – 125	Egy priodusnyi 1-et kap a bemeneten, ezért megkezdődik a kimeneten a megadott y = [0,1,1,1] jelsorozat kiadása – a bemenettől függetlenül.
125 – 165	A bement DC állapotban, a kimeneten az előírt jelsorozat.
165 – 195	A bemenetre 1-et adunk, így a kimenet tartja az 1-es állapotát.
195 – 225	A bemenetre 0-t adunk, ezért a kimenet visszaáll 0-ra.
225 – 235	A bementre ismét ráadunk 1-et, a rendszer elkezdi kiadni az előírt jelsorozatot.
235 – 275	A bemenet DC állapotban, a kimeneten az előírt jelsorozat.
275 –	A bemeneten 0 jel van, így a rendszer visszaáll alapállapotba, a kimenet felveszi a 0-ás jelszintet, mivel a továbbiakban a bemenet nem változik, ezért a kimenet is marad a 0 állapotban.