**Limbaje de descriere hardware**

Tema 7: LFSR Counter with Static Count-to Flag

Student: Parvan Ionut Catalin, grupa 4752

Profesor: Dan Nicula

Anul universitar 2017-2018

**Enuntul problemei:**

## Să se modeleze în Verilog și să se simuleze funcționarea circuitul primit, pornind de la descrierea sa și simbolul bloc.

## Descrierea circuitului:

## Circuitul DW03\_lfsr\_scnto este un numerator crescator parametrizabil cu un indicator al valorii pana la care numara.

## Acesta implementeaza un numerator LFSR, care este un pseudorandom numerator.

## Un numerator LFSR este mult mai rapid decat un numerator binar deoarece un singur bit(primul) este calculate iar ceilalti biti sunt siftati.

## 

Modul DW03\_lfsr\_scnto

**module** DW03\_lfsr\_scnto**#(parameter** width **=** 4**,** **parameter** count\_to **=** 7**)**

**(**clk**,** reset**,** data**,** load**,** cen**,** count**,** tercnt**);**

// Lista porturi

**input** clk **;** // Ceas

**input** reset **;** // Reset

**input** **[**width**-**1**:**0**]** data **;** // Valoare de start pentru numarare

**input** load **;** // Incarcare

**input** cen **;** // Activare cip

**output** **reg** **[**width**-**1**:**0**]** count **;** // Rezultat

**output** tercnt **;** // Finalizare numarare

//Modelare tercnt

**assign** tercnt **=** **(** count\_to**==**count **)** **?** 'b1**:**'b0**;**

//Modelare count

**always** **@(posedge** clk **or** **negedge** reset**)**

**if** **(!**reset**)** count **<=** 'b0**;** **else**

**if** **(**cen**)**

**if** **(!**load**)** count **<=** data**;** **else**

count **<=** **{(**count**[**0**]** **^** **~**count**[**1**]),** count**[**width**-**1**:**1**]};**

**endmodule**

Modul test\_bench\_DW03

`timescale 1ns**/**1ns

**module** test\_bench\_DW03**#(**//Lista parametrii

**parameter** width **=** 4**,**

**parameter** count\_to **=** 7**)(**//Lista posrturi

**output** **reg** load **,**// Incarcare

**output** **reg** rst\_n **,**// Reset

**output** clk **,**// Ceas

**output** **reg** cen **,**// Validare cip

**output** **reg** **[**width**-**1**:**0**]** data // Valoarea de start

**);**

//Construire ceas

clk\_gen **#(**

// Lista parametrii

**.**PER **(**10**)**

**)**

i\_clk\_gen**(**

// Lista porturi

**.**clk**(**clk**)**

**);**

// Construire stimuli

**initial**

**begin**

data **<=** **{(**width**-**1**){**1'b0**}};**

cen **<=** 1'b1**;**

load **<=** 1'b1**;**

rst\_n**<=** 1'b0**;**

**#**15 rst\_n**<=** 1'b1**;**

**#**30 cen **<=** 1'b0**;**

**#**10 cen **<=** 1'b1**;**

**#**20 data **<=** **{(**width**-**1**){**1'b0**}};**

**#**10 load **<=** 1'b0**;**

**#**10 load **<=** 1'b1**;**

**end**

**endmodule**

Modul test\_dw03

**module** test\_dw03**();**// Lista de porturi goala

//Lista parametrii locali

**localparam** width **=** 4**;**

**localparam** count\_to **=** 7**;**

//Lista semnale interne

**wire** clk **;**// Ceas

**wire** rst\_n **;**// Reset

**wire** **[**width**-**1**:**0**]** data **;**// Valoare de start pentru numarare

**wire** load **;**// Incarcare

**wire** cen **;**// Activare cip

**wire** **[**width**-**1**:**0**]** out **;**// Rezultat

**wire** tcnt **;**// Finalizare numarare

//Instanta test bench DW03

test\_bench\_DW03 **#(**width**,** count\_to**)** i\_tb **(**

**.**load **(**load **),**// Iesire

**.**rst\_n **(**rst\_n **),**// Iesire

**.**clk **(**clk **),**// Iesire

**.**cen **(**cen **),**// Iesire

**.**data **(**data **)** // Iesire

**);**

// Instanta numarator DW03 (DUT)

DW03\_lfsr\_scnto **#(**width**,** count\_to**)** i\_dw03\_DUT **(**

**.**clk **(**clk **),** //Intrare

**.**reset **(**rst\_n **),** //Intrare

**.**data **(**data **),** //Intrare

**.**load **(**load **),** //Intrare

**.**cen **(**cen **),** //Intrare

**.**count **(**out **),** //Iesire

**.**tercnt **(**tcnt **)** //Iesire

**);**

**endmodule**

Modul clk\_gen

`timescale 1ns**/**1ns

**module** clk\_gen **#**

**(**// Lista parametrii

**parameter** PER **=** 10

**)** **(**// Lista porturi

**output** **reg** clk

**);**

// Modelare ceas

**initial** clk **=**0**;**

**always** **#(**PER**/**2**)** clk **<=~**clk**;**

**endmodule**

