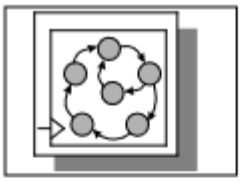
Universitatea
Transilvania
din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR

Str. Politehnicii 1
500024 – Brașov
tel.: (+40) 268.474.718 | fax: (+40)
268.474.718
f-iesc@unitbv.ro | www.unitbv.ro/fiesc

Limbaje de descriere hardware

Tema 5: Gray Code Counter
DW03_cntr_gray

Cadru didactic: Prof. Nicula Dan
Student: Neagu Lucian-Alexandru
Specializarea: Electronică Aplicată
Anul:3
Grupa: 4LF291

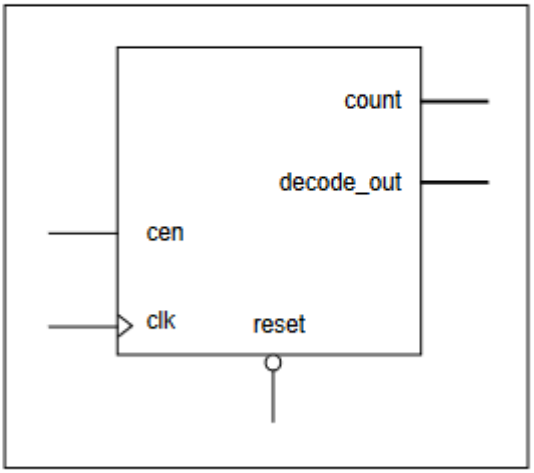


DW03_cntr_gray

Gray Code Counter

Features

- Gray encoded outputs
- Asynchronous reset
- Count enable
- Glitch-free outputs



Description

DW03_cntr_gray is a gray code counter with gray to bit decoder. The counter is *width* bits wide and has 2^{width} states. The counter is clocked on the positive edge of the *clk* input. Because the count sequence is Gray Code, that is, only one counter bit changes state between successive states, the decoded outputs are glitch-free.

reset, active low, provides for an asynchronous reset of the counter to "000...0". If the *reset* pin is connected to '1', then the reset logic is not synthesized, resulting in a smaller and faster counter.

When the count enable pin, *cen*, is HIGH, the counter is active. When *cen* is LOW, the counter is disabled, and *count* remains at the same value.

Table 1 - Pin Description

Pin Name	Size	Type	Function
cen	1	Input	Count enable, active high
clk	1	Input	Clock
reset	1	Input	Asynchronous reset, active low
count	width	Output	Gray coded counter output
decode_out	2^{width}	Output	Bit decode of current count

Table 2 - Parameter Description

Parameter	Function	Legal Range
width	Word length of counter	≥ 1

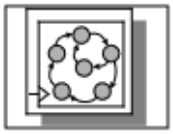


Table 3 - Counter Operation Truth Table

reset	cen	Operation
0	X	Reset
1	0	Standby
1	1	Count

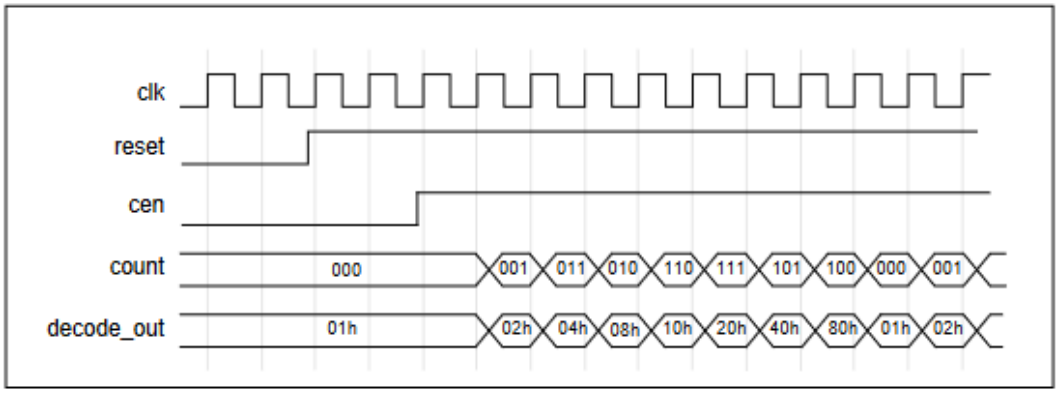
Table 4 - Synthesis Implementations

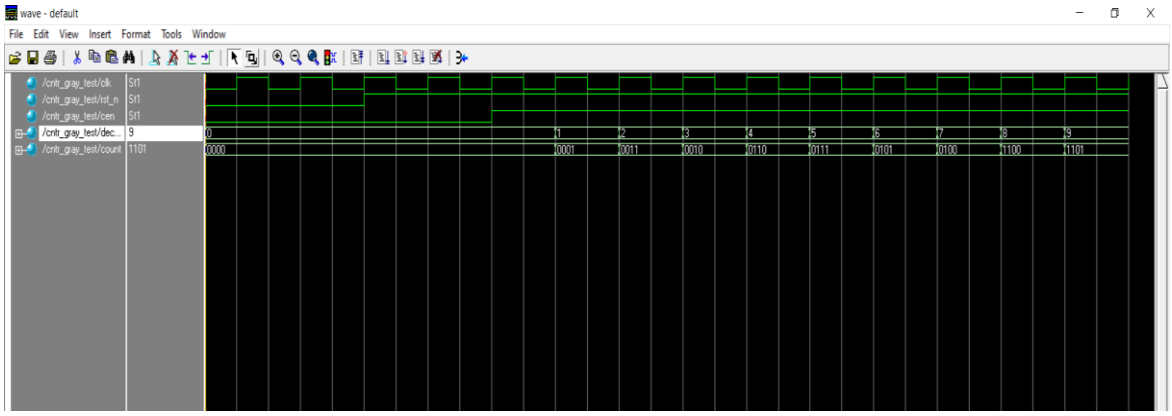
Implementation Name	Function	License Required
str	Synthesis model	DesignWare-Foundation

Table 5 - Simulation Models

Model	Function
DW03.DW03_CNTR_GRAY_CFG_SIM	Design unit name for VHDL simulation
dw/dw03/src/DW03_cntr_gray_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW03_cntr_gray.v	Verilog simulation model source code

Functional Operation (width = 3)





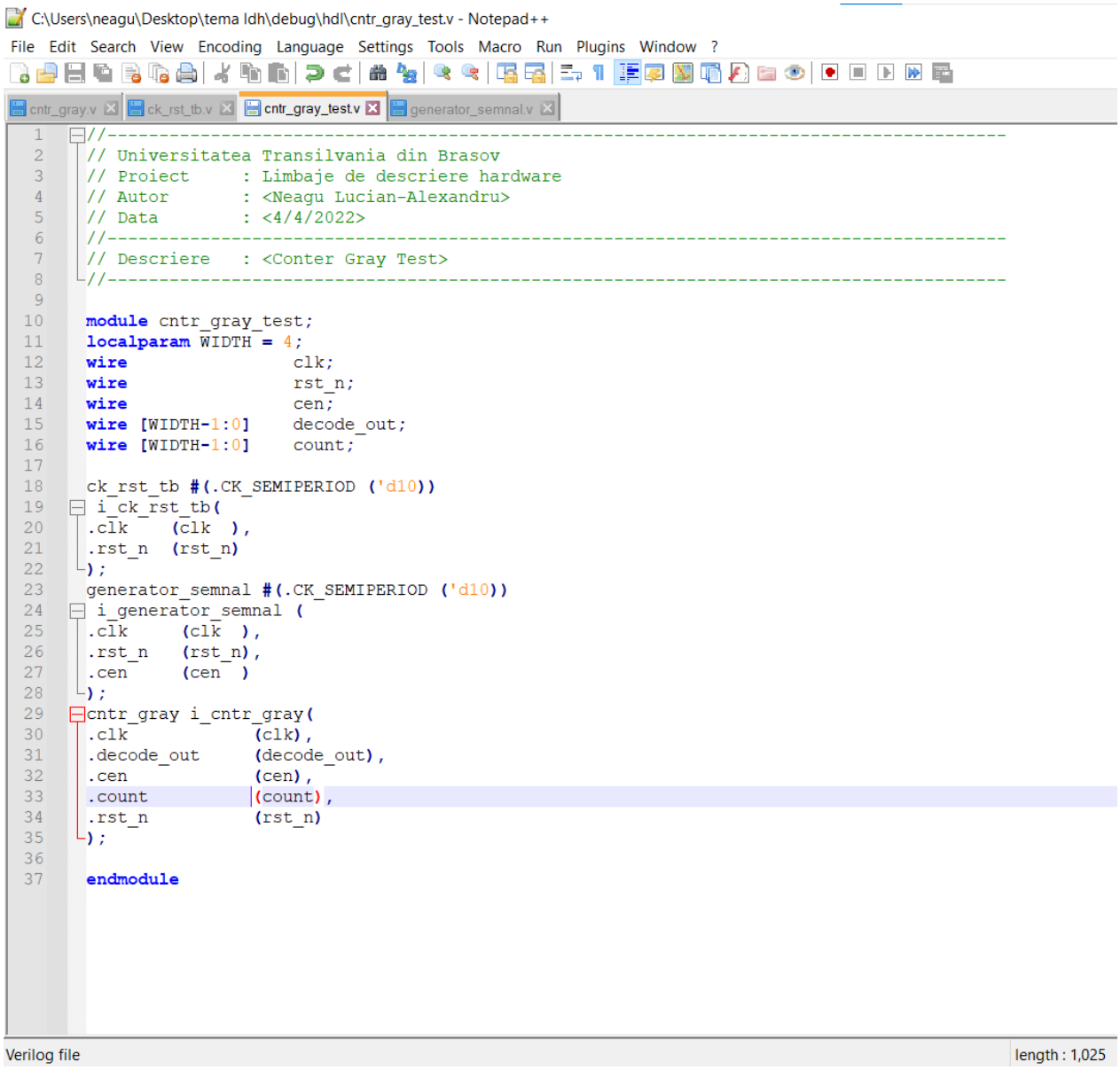
C:\Users\neagu\Desktop\tema ldh\hdl\cntr_gray.v - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

cntr_gray.v | ck_rst_tb.v | cntr_gray_test.v | generator_semnal.v

```
1  //----------------------------------------------------------------------------------------------
2  // Universitatea Transilvania din Brasov
3  // Proiect     : Limbaje de descriere hardware
4  // Autor       : <Neagu Lucian-Alexandru>
5  // Data        : <4/4/2022>
6  //----------------------------------------------------------------------------------------------
7  // Descriere   : <Gray Code Counter>
8  //----------------------------------------------------------------------------------------------
9
10 module cntr_gray #(parameter WIDTH = 4)(        // latimea registrului
11 input                       clk,
12 input                       rst_n,
13 input                       cen,
14 output reg   [WIDTH-1:0]    binary,          // cod natural
15 output wire  [WIDTH-1:0]    count,           //  cod Gray
16 output wire  [WIDTH-1:0]    decode_out
17
18 );
19
20  always @(posedge clk or negedge rst_n) begin
21         if(~rst_n)              begin binary <= 'd0; end               //resetarea valorii numarato
22         else if(cen == 1)       begin binary <= binary + 'd1; end      //incrementarea valorii numa
23
24   end
25
26     assign count = {binary[WIDTH-1], binary[WIDTH-1:1] ^ binary[WIDTH-2:0]};
27     assign decode_out[3]  = count[3];                    //LSB
28     assign decode_out[2]  = count[3] ^ count[2];         // decodare byte 3
29     assign decode_out[1]  = decode_out[2] ^ count[1];    // decodare byte 2
30     assign decode_out[0]  = decode_out[1] ^ count[0];    // decodare byte 1
31
32 endmodule
33 /*
34 module gray_counter(clk, rst, out);
35  input clk, rst;
36  output reg [3:0] out;
37                                                  `ifdef FOR_LOOP
38  reg q0, q1, q2;                                     for (int i=0;i<WIDTH-1;i=i++) begin
39  reg [3:0] count;                                    count[i] <=q[i+1]^q[i];
40                                                          decode_out <=  1 << q ;
41  always @ (posedge clk)                              end
42  begin                                               count[WIDTH-1]<=q[WIDTH-1];
43   if (rst)                                       `else
```

Verilog file | length : 2,619


C:\Users\neagu\Desktop\tema ldh\debug\hdl\cntr_gray_test.v - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
cntr_gray.v
ck_rst_tb.v
cntr_gray_test.v
generator_semnal.v
//--
// Universitatea Transilvania din Brasov
// Proiect : Limbaje de descriere hardware
// Autor : <Neagu Lucian-Alexandru>
// Data : <4/4/2022>
//--
// Descriere : <Conter Gray Test>
//--
module cntr_gray_test;
localparam WIDTH = 4;
wire clk;
wire rst_n;
wire cen;
wire [WIDTH-1:0] decode_out;
wire [WIDTH-1:0] count;
ck_rst_tb #(.CK_SEMIPERIOD ('d10))
i_ck_rst_tb(
.clk (clk),
.rst_n (rst_n)
);
generator_semnal #(.CK_SEMIPERIOD ('d10))
i_generator_semnal (
.clk (clk),
.rst_n (rst_n),
.cen (cen)
);
cntr_gray i_cntr_gray(
.clk (clk),
.decode_out (decode_out),
.cen (cen),
.count (count),
.rst_n (rst_n)
);
endmodule
Verilog file
length : 1,025

C:\Users\neagu\Desktop\tema ldh\debug\hdl\ck_rst_tb.v - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

cntr_gray.v | ck_rst_tb.v | cntr_gray_test.v | generator_semnal.v

```
1  //-------------------------------------------------------------------------------
2  // Universitatea Transilvania din Brasov
3  // Departamentul de Electronica si Calculatoare
4  // Proiect     : Laborator HDL
5  // Modul       : ck_rst_tb
6  // Autor       : Dan NICULA (DN)
7  // Data        : Oct. 1, 2019
8  //-------------------------------------------------------------------------------
9  // Descriere   : Generator de semnal de ceas si de reset asincron (activ 0)
10 //-------------------------------------------------------------------------------
11 // Modificari  :
12 // Oct. 1, 2019 (DN): Initial
13 //-------------------------------------------------------------------------------
14
15 module ck_rst_tb #(
16 parameter CK_SEMIPERIOD = 'd10        // semi-perioada semnalului de ceas
17 )(
18 output reg              clk         , // ceas
19 output reg              rst_n         // reset asincron activ 0
20 );
21  initial
22 begin
23   clk = 1'b0;             // valoare initiala 0
24   forever #CK_SEMIPERIOD  // valoare complementata la fiecare semi-perioada
25     clk = ~clk;
26 end
27
28 initial begin
29   rst_n <= 1'b0;    // initial inactiv
30   repeat(3)@(posedge clk);
31   rst_n <= 1'b1;    // activare sincrona
32   @(posedge clk);
33
34 end
35
36 endmodule // ck_rst_tb
```

Verilog file length : 1,235

C:\Users\neagu\Desktop\tema ldh\debug\hdl\generator_semnal.v - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

cntr_gray.v | ck_rst_tb.v | cntr_gray_test.v | generator_semnal.v

```
1  //------------------------------------------------------------------------------------------------
2  // Universitatea Transilvania din Brasov
3  // Proiect     : Limbaje de descriere hardware
4  // Autor       : <Neagu Lucian-Alexandru>
5  // Data        : <4/4/2022>
6  //------------------------------------------------------------------------------------------------
7  // Descriere   : <Generator Semnal>
8  //------------------------------------------------------------------------------------------------
9
10 module generator_semnal
11 #(parameter CK_SEMIPERIOD = 'd10,
12 parameter WIDTH = 3)
13 (
14     input               clk,
15     input               rst_n,
16     output reg          cen
17
18 );
19 initial
20 begin
21
22     cen <=1'b0;
23     @(posedge clk);
24     repeat(4) @(posedge clk);
25     cen <= 1'b1;
26     repeat(10) @(posedge clk);
27     $stop;
28
29 end
30 endmodule
31
```

Verilog file | length : 831

Neagu Lucian - Alexandru
Gr 4LF291
Gray Code Counter - Circuit Simulator