

Projektowanie Systemów Cyfrowych

- *Prowadzący: dr inż. Andrzej Skoczeń,*

Dodatek 1
2017

Październik 2017

System ISE Xilinx:

- Symulacja na różnych poziomach**
- Biblioteki komórek standardowych i modele symulacyjne**

<http://www.fis.agh.edu.pl/~skoczen/hdl>

Rodzaje symulacji w systemie ISE Xilinx

Poziom	Nazwa	Opis	UNISIM	SIMPRIM	SDF
0	Behawioralny	RTL	×	×	×
I	Po-Syntezie ×	Post-Synthesis (Pre-NGDBuild) GL-funkcyjny	✓	×	×
II	Po-Tłumaczeniu	Post-NGDBuild (Pre-Map) GL-czasowy	×	✓	×
III	Po-Mapowaniu	Post-Map BD-annotowany	×	✓	✓
IV	Po-Rutowaniu	Post-P&R B&ND-annotowany	×	✓	✓

Objaśnienia skrótów:

RTL	Register Transfer Level	Poziom przesłań rejestrowych
GL	Gate Level	Poziom bramek (komórek standardowych)
BD	Block Delay	SDF zawiera tylko opóźnienia komórek standardowych
B&ND	Block & Net Delay	SDF zawiera także opóźnienia węzłów
SDF	Standard Delay Format	Opóźnienia obliczone przez syntezytor
UNISIM		Funkcyjna biblioteka komórek standardowych
SIMPRIM		Czasowa biblioteka komórek standardowych

Generacja modelu symulacyjnego po-syntezie

Poziom I

Nie dostępny w ISE

Podkatalog: netgen/synthesis

Wymaga użycia funkcjonalnej biblioteki UNISIM do poprawnej symulacji

Automatyczne dołączenie modułu globalnej inicjalizacji FPGA

```
Started : "Generate Post-Synthesis Simulation Model".
```

```
Running netgen...
```

```
Command Line:
```

```
netgen -intstyle ise -insert_glbl true -w -dir netgen/synthesis  
-ofmt verilog -sim muxNtol.ngc muxNtol_synthesis.v
```

```
INFO:NetListWriters:633 - The generated Verilog netlist contains Xilinx UNISIM  
simulation primitives and has to be used with UNISIM simulation library for  
correct compilation and simulation.
```

Wynik z syntezaora XST
(Xilinx Synthesis Technology)

```
Process "Generate Post-Synthesis Simulation Model" completed successfully
```

Program NetGen przekształca wynik z syntezy, który jest w formacie NGC do modelu symulacyjnego w Verilog-u lub VHDL-u z użyciem biblioteki komórek standardowych UNISIM.

The simulation model can be used to verify that the functionality is correct after synthesis by running a post-synthesis simulation in your simulator.

Translacja

```
Process "Synthesize - XST" completed successfully
Started : "Translate".
Running ngdbuild...
Command Line:
ngdbuild -intstyle ise -dd _ngo -nt timestamp -i
          -p xc5v1x50t-ff1136-2 uart.ngc uart.ngd
...
NGDBUILD Design Results Summary:
  Number of errors:      0
  Number of warnings:   0
Writing NGD file "uart.ngd" ...
Total REAL time to NGDBUILD completion:  3 sec
Total CPU time to NGDBUILD completion:   2 sec
Writing NGDBUILD log file "uart.bld"...
NGDBUILD done.
Process "Translate" completed successfully
```

Program NGDBuild wykonuje translację.

Translacja jest pierwszym krokiem w procesie implementacji. Łączy on netlisty z syntezy (NGC) i ograniczenia projektowe (UCF - User Constraints File) i tworzy plik w formacie Xilinx® Native Generic Database (NGD).

Generacja modelu symulacyjnego po-tłumaczeniu

Poziom II

Podkatalog: netgen/translate

Symulacja używa czasowej biblioteki SIMPRIM

Started : "Generate Post-Translate Simulation Model".

Running netgen...

Command Line:

```
netgen -intstyle ise -insert_glbl true -w -dir netgen/translate  
-ofmt verilog -sim muxNto1.ngd muxNto1_translate.v
```

INFO:NetListWriters:633 - The generated Verilog netlist contains Xilinx SIMPRIM simulation primitives and has to be used with SIMPRIM simulation library for correct compilation and simulation.

Wynik z programu NGDBuild

Process "Generate Post-Translate Simulation Model" completed successfully

Program NGDBuild wykonuje translację.

Translacja jest pierwszym krokiem w procesie implementacji. Łączy on netlisty z syntezy (NGC) i ograniczenia projektowe (UCF - User Constraints File) i tworzy plik w formacie Xilinx® Native Generic Database (NGD).

Generacja modelu symulacyjnego po-mapowaniu

Poziom III

Nie dostępny w ISE

Podkatalog: netgen/map

Symulacja używa czasowej biblioteki SIMPRIM wraz z plikiem przypisów czasowych SDF z przełącznikiem TYP

Started : "Generate Post-Map Simulation Model".

Running netgen...

Command Line:

```
netgen -intstyle ise -s 4 -pcf muxNto1.pcf -sdf_anno true  
-sdf_path "netgen/map" -insert_global true -w -dir netgen/map  
-ofmt verilog -sim muxNto1_map.ncd muxNto1_map.v
```

INFO:NetListWriters:633 - The generated Verilog netlist contains Xilinx SIMPRIM simulation primitives and has to be used with SIMPRIM simulation library for correct compilation and simulation.

INFO:NetListWriters - Xilinx recommends running separate simulations to check for setup by specifying the MAX field in the SDF file and for hold by specifying the MIN field in the SDF file. Please refer to Simulator documentation for more details on specifying MIN and MAX field in the SDF.

INFO:NetListWriters:665 - For more information on how to pass the SDF switches to the simulator, see your Simulator tool documentation.

Wynik z mapowania

Process "Generate Post-Map Simulation Model" completed successfully

Generacja modelu symulacyjnego po-rutowaniu

Poziom IV

Podkatalog: netgen/par

Symulacja używa czasowej biblioteki SIMPRIM wraz z plikiem przypisów czasowych SDF z przełącznikiem TYP

Started : "Generate Post-Place & Route Simulation Model".

Running netgen...

Command Line:

```
netgen -intstyle ise -s 4 -pcf muxNto1.pcf -sdf_anno true  
-sdf_path "netgen/par" -insert_global true  
-insert_pp_buffers true -w -dir netgen/par -ofmt verilog  
-sim muxNto1.ncd muxNto1_timesim.v
```

INFO:NetListWriters:633 - The generated Verilog netlist contains Xilinx SIMPRIM simulation primitives and has to be used with SIMPRIM simulation library for correct compilation and simulation.

INFO:NetListWriters - Xilinx recommends running separate simulations to check for setup by specifying the MAX field in the SDF file and for hold by specifying the MIN field in the SDF file. Please refer to Simulator documentation for more details on specifying MIN and MAX field in the SDF.

INFO:NetListWriters:665 - For more information on how to pass the SDF switches to the simulator, see your Simulator tool documentation.

Wynik z programu PAR

Process "Generate Post-Place & Route Simulation Model" completed successfully

Wbudowana inicjalizacja FPGA

Moduł sterujący inicjalizacją wysyła dwa globalne sygnały:

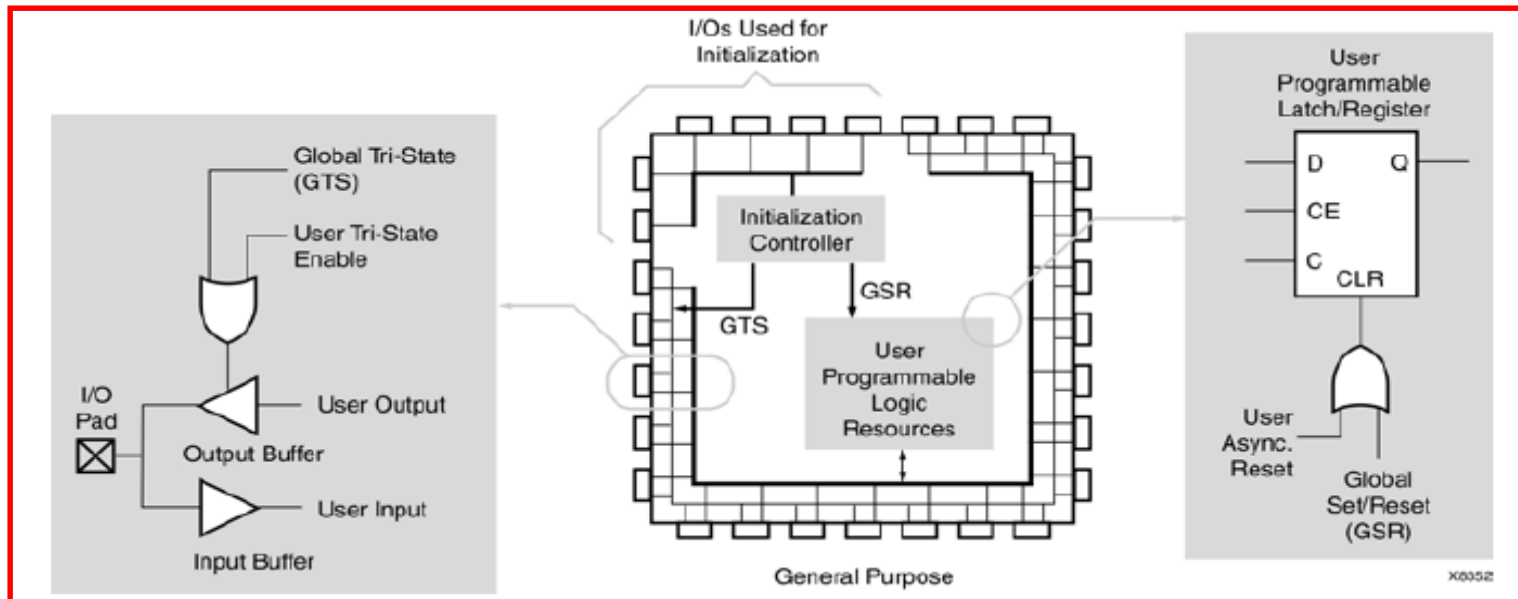
- ❑ Global Set/Reset (GSR) i
- ❑ Global Tristate (GTS) .

Są one zdefiniowane w module `glbl` znajdującym się w pliku `$XILINX/verilog/src/glbl.v`

Konieczne jest aby ten plik kompilować razem z plikami projektu i modułu testowego do symulacji.

Moduł `glbl` w zerowej chwili czasu symulacji wysyła impuls GSR o czasie trwania 100ns.

Użycie opcji `-insert_glbl true` przy wywołaniu programu `netgen` zapewnia automatyczne wstawienie tego modułu do modelu symulacyjnego. Opcję tą można wyłączyć we własnościach procesu generacji netlisty (*Generate Post-XXX Simulation Model*) w wersji zaawansowanej (*Advanced*).



Globalny reset w testbenchu

Przy domyślnych opcjach programu netgen sygnał **GSR** trzeba uwzględnić w testbenchu:

```
module tb;
  reg clk, rst, set, d;
  wire q;

  assigntry uut (.clk(clk), .rst(rst),
               .set(set), .d(d), .q(q));

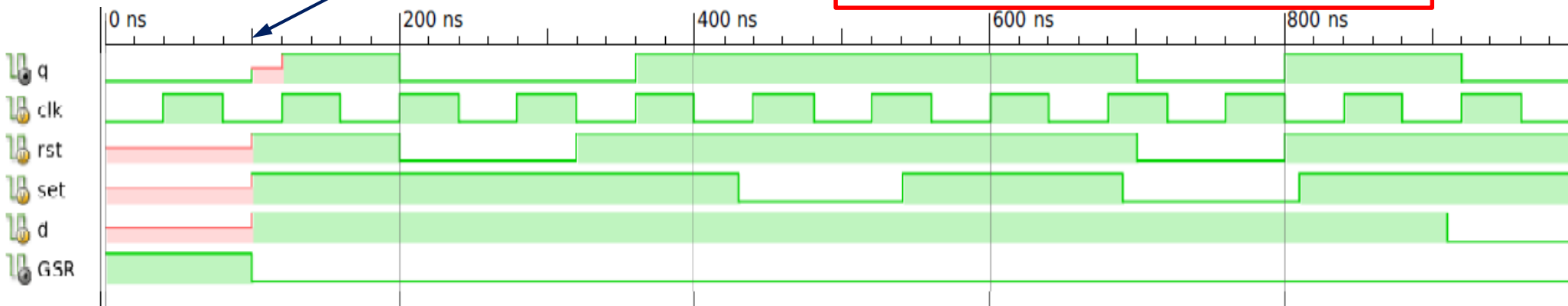
  wire GSR = glbl.GSR;

  initial begin
    clk = 0;
    forever #40 clk = ~clk;
  end
end
```

```
initial begin
  @(negedge GSR);
  rst = 1'b1;
  set = 1'b1;
  d = 1'b1;
  #100 rst = 1'b0;
  #120 rst = 1'b1;
  #110 set = 1'b0;
  #110 set = 1'b1;
  #150 set = 1'b0;
  #10 rst = 1'b0;
  #100 rst = 1'b1;
  #10 set = 1'b1;
  #100 d = 1'b0;
end

initial #1000 $finish;
endmodule
```

Koniec resetu globalnego GSR 100ns



Biblioteki Xilinx'a

Xilinx® dostarcza następujące biblioteki symulacyjne dla symulacji komórek pierwotnych i rdzeni:

- ❑ **UNISIM** – funkcjonalne symulacje komórek pierwotnych
- ❑ UniMacro - funkcjonalne symulacje makr
- ❑ XilinxCoreLib - funkcjonalne symulacje rdzeni
- ❑ Xilinx EDK – behawioralne symulacje elementów IP Xilinx Embedded Development Kit EDK
- ❑ **SIMPRIM** – czasowe symulacje komórek pierwotnych
- ❑ SmartModel/SecureIP – funkcjonalne i czasowe symulacje Xilinx Hard-IP, takich jak PPC, PCIe®, GT, and TEMAC IP.

Biblioteka	Typ	ścieżka
UNISIM	funkcjonalna	\$XILINX/verilog/src/unisims
SIMPRIM	czasowa	\$XILINX/verilog/src/simprims

Opis komórek standardowych najważniejszych bibliotek (UNISIM, SIMPRIM) dla rodziny Spartan-3A znajduje się w dokumencie UG613 „*Spartan-3A and Spartan-3A DSP Libraries Guide for HDL Designs*”

Biblioteki Xilinx'a

Najprostszy bufor

Biblioteka funkcjonalna UNISIM

```
`timescale 1 ps / 1 ps
module BUF (O, I);
    output O;
    input I;

    buf B1 (O, I);
endmodule
```

Biblioteka czasowa SIMPRIM

```
`timescale 1 ps/1 ps
module X_BUF (O, I);
    parameter LOC = "UNPLACED";
    output O;
    input I;

    buf (O, I);

    specify
        (I => O) = (0:0:0, 0:0:0);
        specparam PATHPULSE$ = 0;
    endspecify
endmodule
```

Wyjaśnimy o co tu chodzi jeśli
będziemy mówić o *timing checks*

Biblioteki Xilinx'a

Najprostszy przerzutnik – instancja modelu funkcjonalnego i model czasowy

```
module X_FF (O, CE, CLK, I, RST, SET);
...
ffsrce (o_out, CLK_dly, I_dly, CE_dly, set_int, rst_int, notifier1);
  specify
    (CLK => O) = (100:100:100, 100:100:100);
    (SET => O) = (0:0:0, 0:0:0);
    (RST => O) = (0:0:0, 0:0:0);

    $setuphold (posedge CLK, posedge CE &&& (ce_clk_enable1!=0), 0:0:0, 0:0:0, notifier,,,CLK_dly,CE_dly);
    $setuphold (posedge CLK, negedge CE &&& (ce_clk_enable1!=0), 0:0:0, 0:0:0, notifier,,,CLK_dly,CE_dly);
    $setuphold (posedge CLK, posedge I &&& (in_clk_enable1!=0), 0:0:0, 0:0:0, notifier,,,CLK_dly,I_dly);
    $setuphold (posedge CLK, negedge I &&& (in_clk_enable1!=0), 0:0:0, 0:0:0, notifier,,,CLK_dly,I_dly);

    $recrem (negedge RST, posedge CLK &&& (rst_clk_enable1!=0), 0:0:0, 0:0:0, notifier,,,RST_dly, CLK_dly);
    $recrem (negedge SET, posedge CLK &&& (set_clk_enable1!=0), 0:0:0, 0:0:0, notifier,,,SET_dly, CLK_dly);

    $period (posedge CLK &&& CE, 0:0:0, notifier);
    $width (posedge CLK &&& CE, 0:0:0, 0, notifier);
    $width (negedge CLK &&& CE, 0:0:0, 0, notifier);
    $width (posedge RST, 0:0:0, 0, notifier);
    $width (posedge SET, 0:0:0, 0, notifier);

    specparam PATHPULSE$ = 0;
  endspecify
endmodule
```

Biblioteki Xilinx'a

Model funkcjonalny z uwzględnieniem powiadomienia `notifier` o złamaniu reguł czasowych.

```
primitive ffsrce (output reg q,  
input clk, d, ce, set, rst, notifier);
```

```
table  
//clk    d      ce    set    rst    notifier    q      q+;  
?       ?      ?     1     0     ?      :    ?      :    1;  
?       ?      ?     ?     1     ?      :    ?      :    0;  
  
(01)    0      1     0     0     ?      :    ?      :    0;  
(01)    1      1     0     0     ?      :    ?      :    1;  
(01)    x      1     0     0     ?      :    ?      :    x;  
(01)    0      x     0     0     ?      :    0      :    0;  
(01)    1      x     0     0     ?      :    1      :    1;  
  
(??)    ?      0     ?     ?     ?      :    ?      :    -;  
(1?)    ?      ?     ?     ?     ?      :    ?      :    -;  
(?0)    ?      ?     ?     ?     ?      :    ?      :    -;  
  
(01)    0      1     0     x     ?      :    ?      :    0;  
(01)    1      1     x     0     ?      :    ?      :    1;  
  
?       ?      ?     0     (?x)  ?      :    0      :    0;  
?       ?      ?     (?x)  0     ?      :    1      :    1;  
  
(?1)    1      ?     ?     0     ?      :    1      :    1;  
(?1)    0      ?     0     ?     ?      :    0      :    0;  
(0?)    1      ?     ?     0     ?      :    1      :    1;  
(0?)    0      ?     0     ?     ?      :    0      :    0;  
  
?       (??)   ?     ?     ?     ?      :    ?      :    -;  
?       ?     (??)  ?     ?     ?      :    ?      :    -;  
?       ?     ?     (?0)  ?     ?      :    ?      :    -;  
  
?       ?     ?     x     (?0)  ?      :    ?      :    x;  
?       ?     ?     0     (?0)  ?      :    ?      :    -;  
  
?       ?     ?     ?     ?     *      :    ?      :    x;  
endtable
```