Układy FPGA

– programowanie bramek w zestawie uruchomieniowym Basys3 firmy Digilent
-środowisko Vivado (2016.4)- firmy Xilings , język Verilog

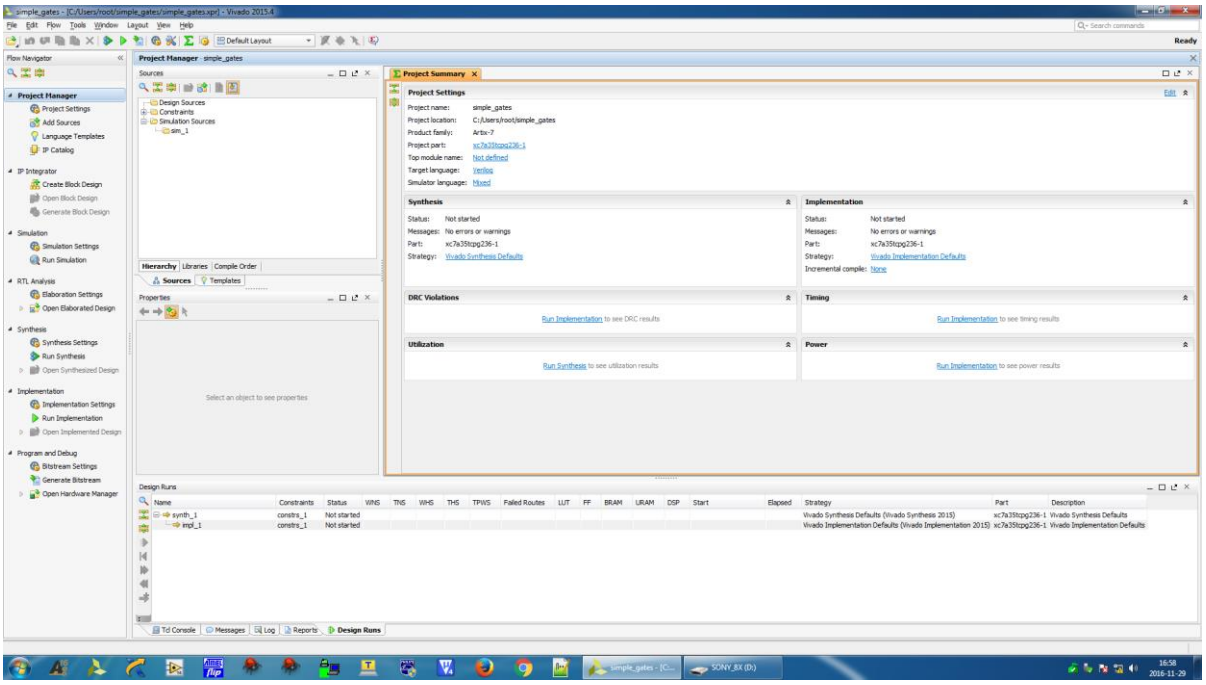
Pierwszy projektowany układ będzie zawierał dwie bramki logiczne, AND i OR, których wejścia będą podłączone do przełączników na płytce Basys3 (oznaczone od SW0 do SW15 - wykorzystamy SW0 - SW3), a stan wyjść będzie wskazywany przez diody LED (oznaczone od LD0 do LD15 - wykorzystamy LD0, LD1).

1.Tworzenie projektu

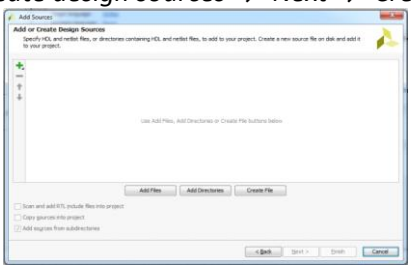
- *Uruchamiamy środowisko Vivado 2016.4 (na dolnym pasku)*
- *Klikamy na ikonę tworzenia nowego projektu ("Create New Project"). Otwiera sie okno "Create a New Vivado Project" → Next*
- *W oknie **Project name** nazwij projekt **simple_gates** i wybierz jego lokalizację jako np. **Project location**: C:/Users/student. Pole **Create project subdirectory** pozostaw zaznaczone. → Next*
- *W oknie **Project type** wybierz opcję „**RTL project**” (Register Transfer Level) z zaznaczeniem „**Do not specify resource at this time**” (kody źródłowe zostaną dodane w późniejszym etapie). → Next*

*W oknie **Default part** należy wybrać:*
Product category: General Purpose
Family: Artix-7
Sub-family: Artix-7
Package: cpg236
Speed grade: -1
Temperature grade: All Remaining
Z pozostałych do wyboru zaznaczyć xc7a35tcpg236-1. Model ten posiada 20800 tablic LUT. → Next

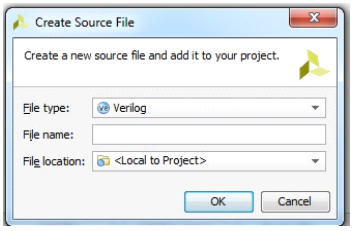
- *Następnie pojawia się okno: "New project Summary" a w nim opisane są parametry charakterystyczne projektu →Finish. Przez parę sekund tworzy się projekt (Create project ……) i otwiera się kolejne okno czyli projekt: simple_gates.*
- *Po utworzeniu projektu, z lewej strony okna powinien znajdować się panel Flow Navigator. Jeżeli jest on niewidoczny, wciśnij kombinację Ctrl+Q.*



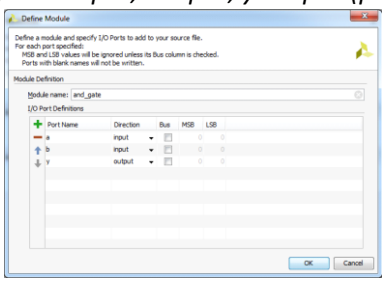
1.Dodawanie plików do projektu
W panelu Flow Navigator będziemy rozwijali zakładkę Project Manager. Utworzymy model bramki AND o nazwie „and_gate", która posiada wyjście „y" i dwa porty wejściowe „a" i „b";
(Project Manager -> Add sources -> Add or create design sources → Next → Create File…)



- *Pojawia sie małe okno: Create Source File*



- *File type: Verilog , File name: and_gate, File location: → Choose location: Local to Project → OK*
- *Pojawia się nowe okno z opisem i lokalizacją projektu →Finish*
- *Następne małe okno: Define Module . Sprawdzamy Module name: and_gate i definiujemy porty (I/OPort Definitions). Port Name/Direction kolejno: a input, b input, y output.(pole Bus zostaje niezaznaczone) → OK*



W oknie Project Manager – Sources pojawia się folder : Design Sources - plik (ve): and_gate.v ,(na dole okna powinna być aktywna zakładka „Hierarchy".
Otwieramy go klikając (prawy przycisk myszy) →Open File. Pojawia się zawartość pliku (opis modułu): są w nim tylko oznaczenia wejść i wyjść - bez połączeń i funkcji

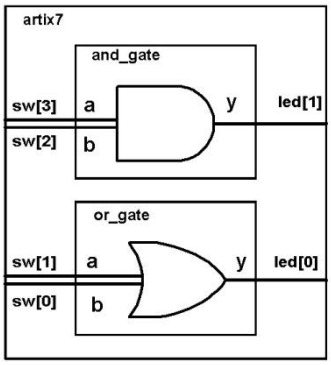
```
`timescale 1ns / 1ps

module and_gate
    (
     input a,
     input b,
     output y
     );
endmodule
```

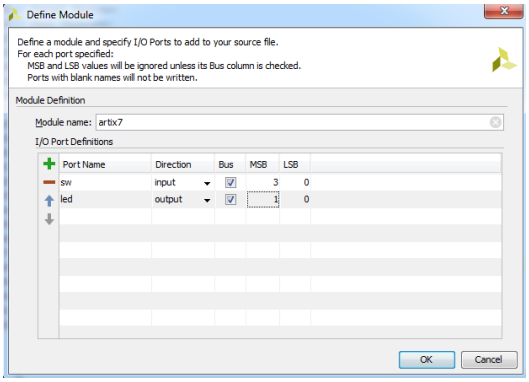
Uzupełnij ten kod kodu, aby wyglądał jak poniżej: (trzeba dokonać połączeń i przypisać funkcje)

```
`timescale 1ns / 1ps

module and_gate
    (
     input  wire a, // połączenie
     input  wire b,
     output wire y
     );
     assign y = a & b; //przypisanie ciągłe, & to AND bitowy
endmodule
```

Zapisujemy moduł (Ctrl + S). Po każdym zapisaniu pliku Vivado analizuje kod. Jeżeli są jakieś błędy składni, to informacja o nich zostaje wyświetlona w zakładce „Messages" okna „Messages"(na dole). Wygeneruj jakiś błąd (np. usuń średnik na końcu komendy assign) i sprawdź, jak wygląda taki komunikat błędu. Zwróć również uwagę, że pliki z błędami pojawiają się w oknie „Sources" w folderze „Design sources → Syntax Error Files"

- *Utwórz analogicznie moduł bramki OR o nazwie "or_gate". Pamiętaj o symbolu OR bitowy (str 12 Verilog Quick Reference). Po zapisaniu modułu bramki OR w folderze Design Sources sa dwa pliki: and_gate nadrzedny (oznaczony trzema punktami jako hierarchia) oraz or_gate podrzędny. Prawidłowa hierarchia ułozy się dopiero po skonfigurowaniu „artix7" .*
- *Utwórz model całości układu o nazwie „artix7" według schematu poniżej:*

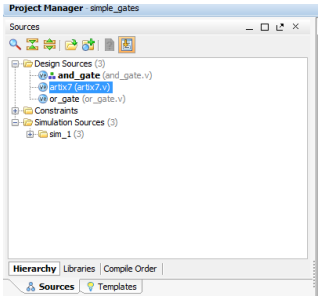




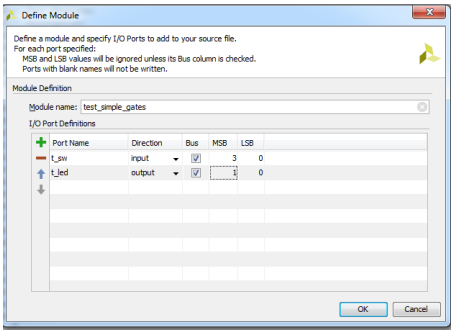
- *Jest to moduł hierarchiczny: przypisanie a,b odpowiednio do sw oraz y do led.*
- *.a(sw[1]) → . kropka oznacza strukturę hierarchiczną, np: a oznacza wejscie/wyjście (lub port) niższego poziomu , zawartość nawiasu (sw[1]) oznacza wejscie/wyjście (lub port) wyższego poziomu , np. sw[1] oznacza określony przełącznik, led[0] oznacza diodę.*
- *Utwórz model całości układu o nazwie „artix7" :*

```
`timescale 1ns / 1ps

module artix7
    (
        input  wire [3:0] sw,
        output wire [1:0] led        // uzupełnij analogicznie jak dla input wire …
    ) ;

    or_gate u_or_gate
    (
        .a(sw[1]),
        .b(sw[0]),
        .y(led[0])
    );

    and_gate u_and_gate   // uzupełnij analogicznie jak dla or_gate u_or_gate_

endmodule
```

Zapisujemy moduł. Teraz ustawia się prawidłowa hierarchia w Design Sources: artix7 – moduł nadrzędny i dwa moduły podrzędne (and_gate i or_gate).
Po opracowaniu modelu obejrzyj schemat układu. Jeżeli są jakieś błędy podczas łączenia projektu (elaboracji), to należy je usunąć. W zakładce Sources w folderze Design Sources są aktualne pliki zapisane hierarchicznie w postaci:



- ***SYMULACJA:*** *W celu przeprowadzenia symulacji potrzebujemy jeszcze minimum jednego modułu testowego, który zapewni nam generację przebiegów testowych (w folderze Simulation Sources).*
- *Utwórz moduł File name: „test_simple_gates". Project Manager →Add Sources→Add or create simulation sources→Next→Create File*



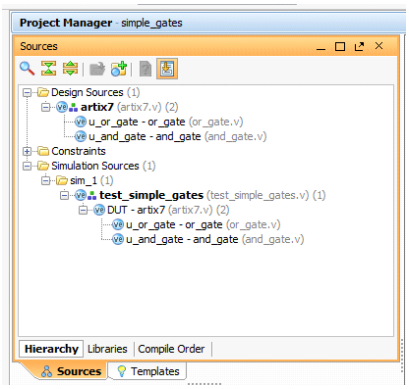
Okno Define Module:

Port Name:	*Direction:*	*Bus:*	*MSB:*	*LSB:*
t_sw	*input*	*v*	*3*	*0*
t_led	*output*	*v*	*1*	*0*

Moduł ma zapewnić przetestowanie bramek dla wszystkich możliwych konfiguracji wejść. Fragment kodu znajduje się poniżej.
W oknie Sources rozwijamy folder: sim_1 i otwieramy plik: test_simple_gates. Wpisujemy poleceniajak poniżej. Miejsca oznaczone ……. należy uzupełnić kodem (do przetestowania pozostałych konfiguracji). Użyj pętli for (lub repeat). Składnię for utwórz według "Verilog Quick Reference" - punkt 7.3 str. 15 (bez memory [i]=0).

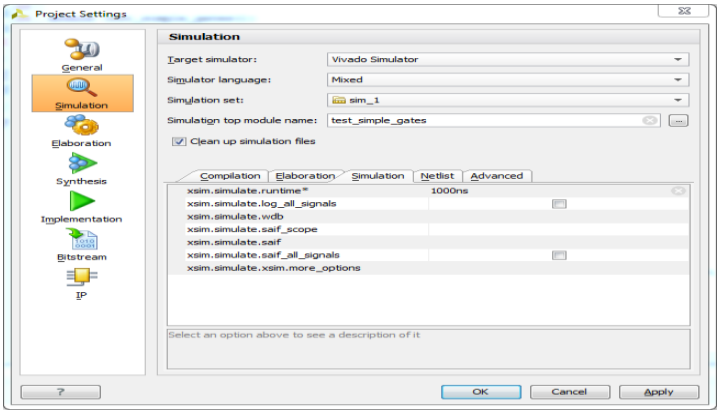
```
`timescale 1ns / 1ps

module test_simple_gates();

	reg  [3:0] t_sw;  // patrz Quick Ref. str 2, reg to 4-bitowy vector register
	wire [1:0] t_led;

	artix7 DUT   // DUT - Device Under Test
	(
	    .sw (t_sw),  // struktura hierarchiczna
	    .led(t_led)
	);
	..................  // deklaracja zmiennej typu integer do iteracji (dla pętli for)
	initial
	begin

		$display("%0t: Starting TPG.", $time);

		// set the inputs
		#10 t_sw = 4'b0000;

		for ...................  // pętla for patrz Quick Ref. str 15, p. 7.3
	    begin
		#10
		t_sw=i;  //zmienną przypisujemy do t_sw
		#1
		if(t_led == { ......... , ......... }) // wpisać kolejno funkcję AND , OR dla t_sw
	//według rysunku modułu hierarchicznego
		$display("%0t Test PASSED for pattern %b", $time, t_sw);
		else
		$display("%0t Test FAILED for pattern %b - result is %b", $time, t_sw, t_led);
	    end

		#10 $display("%0t: Finished TPG.", $time);

		$stop;

    end
endmodule
```

Zapisujemy, rozwijamy foldery. Po załadowaniu wszystkich źródeł zakładka „Sources" powinna wyglądać tak:

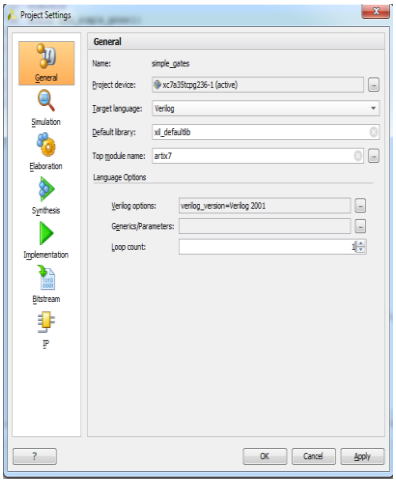


*Jeżeli plik **test_simple_gates** znajduje się w grupie „Design Sources" zamiast w „Simulation Sources", należy kliknąć (prawy przycisk myszy) i wybrać „Move to Simulation Sources".*

- *SYMULACJA : Uruchomienie symulacji*
  - *Aby skonfigurować parametry symulacji, w panelu Flow Navigator wybierz zakładkę Simulation -> Simulation settings.*
  - *Wybierz zakładkę Simulation/Simulation(w środku okna). W polu xsim.simulation.runtime dopasuj odpowiednio czas trwania pojedynczej symulacji. Pamiętaj, że nawet jeżeli ustawisz dłuższy czas trwania symulacji, to zostanie ona zatrzymana komendą $stop. Jeżeli ustawisz zbyt krótki, zawsze możesz po zatrzymaniu kontynuować. Pole wyboru czasu trwania pozostawić 1000 ns.→OK*



- *Wybierz zakładkę General. W polu Loop count ustawić "jeden" ilość powtórzeń symulacji przy pojedynczym jej wywołaniu (UWAGA: przecinek nie jest separatorem dziesiętnym!) : →OK*

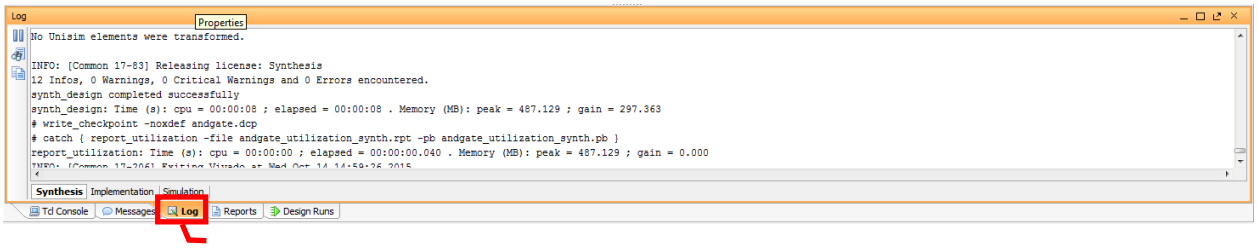


- *W panelu Flow Navigator wybierz zakładkę Simulation → Run Simulation →Run Behavioral Simulation (Przeprowadź symulację behawioralną). Przebiegi symulacji powinny wyświetlić się na grafie zbliżonym do:*

- *Najpierw należy otworzyć na pasku górnym zakładkę „Untitled1" a potem rozwinąć t_sw oraz t_led , a następnie na pasku pionowym kliknąć ikonkę „Zoom fit"*

W oknie po lewej zamiast „Sources" rozwija się okno „Scope".

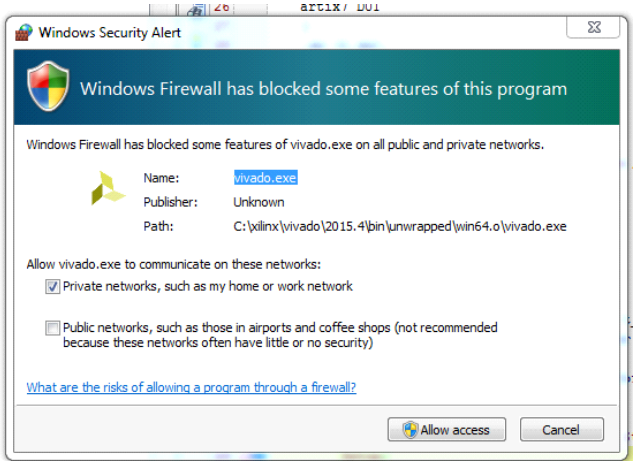
- *Klikając w zakładkę Log znajdującą się w dole okna programu można podglądać zapisy logów:*



Zakładka Log

- ***Implementacja w układzie FPGA***

- ***SYNTEZA:*** *Uruchom syntezę (-> Run Synthesis). Zwróć uwagę na pasek statusu w prawym górnym rogu ekranu. Synteza przekształca opis w języku HDL na schemat w postaci bramek . Może pojawić sie komunikat dotyczacy blokady, należy wybrać Allow Access.*



- *Należy śledzić w prawym górnym rogu ekranu przebieg syntezy i czekać na komunikat „Synthesis Succesfully Completed". Po zakończeniu, na razie, nie wybieramy „Run Implementation" tylko "View Reports"→OK*

- *Sprawdź, ile procent dostępnych tablic LUT, buforów wejściowych IBUF i buforów wyjściowych OBUF wykorzystał na układ („Reports → Utilization Raport).*

- *Zauważ, że w menu Run Synthesis pojawiły się dwie nowe opcje (nie musisz ich teraz uruchamiać).*

Aby przeprowadzić implementację sprzętową, musimy wprowadzić ograniczenie projektowe (ang. constraints), które powiedzą Vivado, które wyprowadzenia układu FPGA są używane i jak są skonfigurowane.

W tym celu:
1)dodaj do projektu plik ograniczeń XDC: Basys3 Master.xdc (który otrzymacie od prowadzącego) .
2)Następnie należy zapisać poprzez: Add Sources → Add or Create Constraints →Add Files (z projektu: simple_gates). Odkomentuj potrzebne linijki w pliku .Należy odkomentować wszystkie używane switche(sw 0-3) i ledy(led0-1) po dwa wiersze na każdy switch i led.
3)Sprawdzić czy plik xdc jest w folderze „Constraints".

- ***IMPLEMENTACJA:*** *Uruchom implementację („→ Run Implementation"). Obserwuj log. Po implementacji zauważ, że pojawiły się nowe zestawy raportów „Place Design" i „Route Design". W większości z uwagi na prostotę projektu będą one większości puste.*

  - *Znajdź w raporcie informację o procencie wykorzystanych zasobów IO (Place Design->Utilization Raport, Bonded IOB) .*

  - *Otwórz topografię układu (→ Implemented Design), znajdź jedną z tablic LUT (look-up table), która został wykorzystana do budowy bramki AND lub OR, podświetl połączenia wejściowe i wyjściowe .*

  - *Zaznacz na topografii slice i otwórz jego schemat (prawy przycisk myszy ->Schematic lub F4). Powinien on zawierać 4 bufory wejściowe, dwa elementy tablicy LUT i dwa bufory wyjściowe.*

  - *W „Messages" mogą się pojawić teraz ostrzeżenia, które ignorujemy:*

  [Timing 38-313] There are no user specified timing constraints. Timing constraints are needed for proper timing analysis.

  [Chipscope 16-3] Cannot debug net 'led[0]'; it is not accessible from the fabric routing.(12 more like this)

- ***PROGRAMOWANIE I DEBUGOWANIE:*** *Należy zaznaczyć „bin file" w Bitstream Settings. Wygeneruj bitstream (plik zawierający dane do programowania FPGA) → Generate Bitstream.*

- *Upewnij się, że przełączniki na układzie Basys3 są w odpowiednim ustawieniu (obie niebieskie zworki "bliżej środka" płytki):*
  -JP1 (Mode) → USB
  -JP2 (Power) → USB

- *Podłącz układ Basys3 przy pomocy kabla USB do komputera i włącz go przełącznikiem „Power Switch".*
  UWAGA: *przed wsunięciem wtyku kabla do gniazda płytki Basys3* ***UPEWNIĆ SIĘ ,*** *że wtyk microUSB („typ smartfonowy") jest prawidłowo ułożony - wsuwamy do gniazdka „ząbkami do góry".*

- *Skonfiguruj podłączenie do programowania płytki Basys3:*
  -Program and Debug -> Hardware Manager -> Open Target -> Open New Target …
  -Wybierz: local server, zegar programowania domyślny 15MHz

- *Program device -> xc7a35t_0*
  -Bitstream File powinien być domyślnie ustawiony poprawnie (sprawdzić w okienku)
  -Dostaniemy ostrzeżenie, które w tym przypadku możemy zignorować:

WARNING: [Labtools 27-3123] The debug hub core was not detected at User Scan Chain 1 or 3. (Okno Messages bez Errors).

- *Sprawdź, czy układ działa poprawnie. Jeżeli tak – gratulacje!*