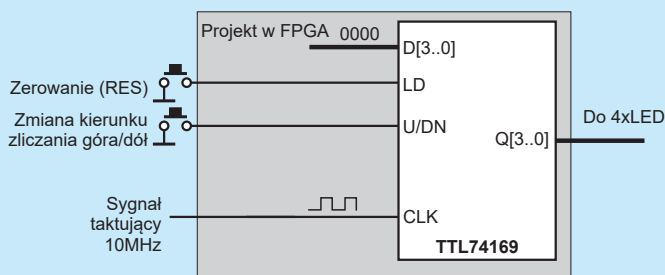


Pierwsze kroki z FPGA (2)

Szkoła MAXimatora – pierwszy projekt z edytorem schematów

W pierwszej części kursu przedstawiliśmy podstawowe informacje na temat tego skąd i jakie narzędzia programowe będą nam potrzebne do rozpoczęcia przygody z układami FPGA i zestawem MAXimator. W tej części pokażemy jak krok-po-kroku przygotować pierwszy projekt. Ze względu na dostępny w bezpłatnej wersji programu Quartus Prime zestaw bogato wyposażonych bibliotek oraz łatwość przygotowania projektu z wykorzystaniem edytora schematów, w naszych pierwszych krokach skupimy się na implementacji projektu bazującego na modelach znanych elektronikom modelach układów z biblioteki TTL-7400.



Rysunek 1. Schemat blokowy pierwszego projektu

Schemat blokowy pierwszego projektu, jaki zaimplementujemy w zestawie MAXimator, pokazano na **rysunku 1**. Jest on – jak widać – nieskomplikowany i podczas jego implementacji nie zużyjemy zbyt wielu zasobów FPGA. Ta prostota ułatwi nam skupienie uwagi na przebiegu realizacji projektu w środowisku Quartus Prime Lite z wykorzystaniem wbudowanego w nie edytora schematów.

W implementowanym projekcie użyjemy 4-bitowego, binarnego, synchronicznego licznika znanego z rodziny TTL, oznaczonego symbolem 74169 (**rysunek 2**). Układ ten jest pozbawiony dedykowanego wejścia zerującego, dlatego do zerowania licznika użyjemy 4-bitowego

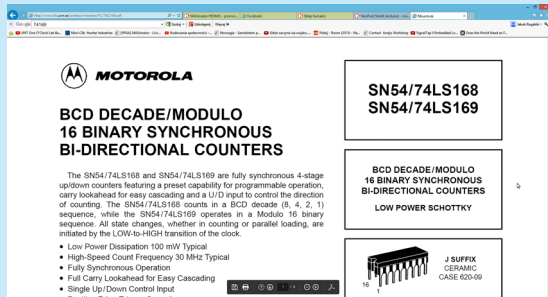
Quartus Prime i Windows 32?

Quartus Prime w wersjach obsługujących układy MAX10 jest dostępny dla Windows wyłącznie 64-bitowych.

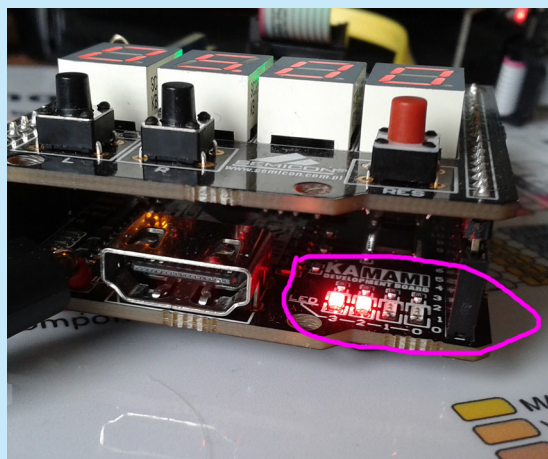
wejścia równoległego D3...D0 (na którego liniach są przypisane stany 0000) i linii ładowania wartości początkowej LD. Licznik 74169 wyposażono w wejście zmiany kierunku zliczania U/DN, na którym stan logiczny określa czy stan licznika po każdym impulsie zegarowym ulega zwiększeniu czy zmniejszeniu o jeden. Do wejść LED i U/DN są dołączone przyciski, które fizycznie są zamontowane na shieldzie wchodzącym w skład zestawu MAXimator. Wyjścia licznika 74169 dołączono do 4 LED będących elementem wyposażenia płytki bazowej MAXimator (fotografia 3). Służą one do monitorowania stanów wyjść licznika Q3...Q0, przy czym trzeba pamiętać, że ich katody są sterowane z wyjść FPGA co oznacza, że diody świecą wtedy, gdy na wyjściach FPGA są stany „0”.

Tworzenie projektu

Pracę rozpoczniemy od utworzenia projektu za pomocą kreatora wbudowanego w oprogramowanie, co wymaga – po uruchomieniu programu Quartus Prime Lite – wybrania opcji w menu *File>New Project Wizard* (rysunek 4). Kolejno będą wyświetlane okna kolejnych kroków kreatora, w których musimy podać nazwę projektu (w przykładzie jako nazwę przyjęliśmy *licznik_74169*) i lokalizację plików wchodzących w jego skład (rysunek 5). W kolejnych krokach pomijamy możliwość modyfikowania domyślnych ustawień, dopiero w oknie *Family & Device Settings* wybieramy rodzinę MAX10



Rysunek 2. 4-bitowy, binarny licznik synchroniczny 74169



Fotografia 3. Wyjścia licznika dotychczas do 4 LED będących elementem wyposażenia płytki bazowej MAXimator

(rysunek 6) i typ docelowego układu 10M08DAF256C8GES (rysunek 7). Na tym kończymy wstępną konfigurację projektu i przechodzimy do utworzenia i edycji schematu.

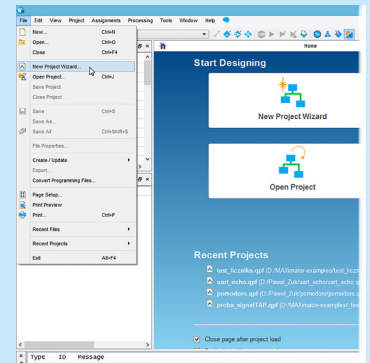
Rysowanie schematu

W projekcie założonym według wcześniejszego opisu tworzymy planszę schematu, co wymaga wybrania w menu opcji *File>New*, w wyniku czego wyświetla się okno pokazane na rysunku 8. Wybieramy w nim opcję (w sekcji *Design Files*) *Block Diagram/Schematic File* w wyniku czego do projektu *licznik_74169* zostanie dodana czysta plansza schematu, jak pokazano na rysunku 9.

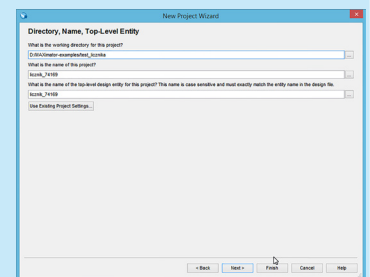
Dwukrotne kliknięcie prawym przyciskiem myszy na planszy schematu powoduje otwarcie okna dostępu do predefiniowanych graficznych symboli bibliotecznych (rysunek 10), z których użyjemy element z biblioteki *others>max2plus* o nazwie 74169 (rysunek 11). Element wybrany z listy kładziemy na planszy schematu (rysunek 12), teraz „obrysujemy” go, wykonując lokalne połączenia, zgodnie z opisem na wstępie i dokumentacją licznika 74169. Potrzebny nam symbol „masy” (odpowiadający stanowi logicznemu „0”) także pobieramy z biblioteki elementów graficznych, jest on dostępny w katalogu *primitives>other* (rysunek 13). Połączenia wykonujemy za pomocą narzędzia *Orthogonal Node Tool*, które jest dostępne w menu programu (rysunek 14).

MAXimator – zacznij przygodę z FPGA

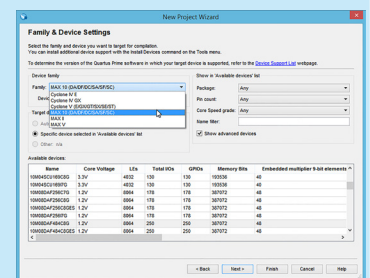
Zestaw MAXimator jest atrakcyjną propozycją nie tylko dla początkujących z FPGA. Jednym z jego poważnych atutów jest bardzo niska cena, co udało się producentowi uzyskać dzięki ściślej współpracy z kilkoma firmami, które wsparły zamysł edukacyjny pomysłu. I tak: firmy Altera i EBV dostarczyły podzespoły do zestawu, firma NCAB wyprodukowała płytki drukowane zestawu, firma Semicon zamontowała elementy, a KAMAMI opracowała zestaw, stronę internetową, dokumentację i materiały supportowe.



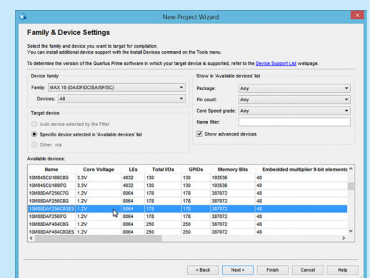
Rysunek 4. Pracę rozpoczniemy od utworzenia projektu za pomocą kreatora wbudowanego w oprogramowanie



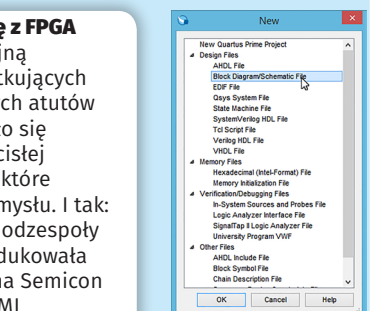
Rysunek 5. Wskazanie lokalizacji plików



Rysunek 6. Wybór rodziny MAX10



Rysunek 7. Wybór typu docelowego układu 10M08DAF256C8GES

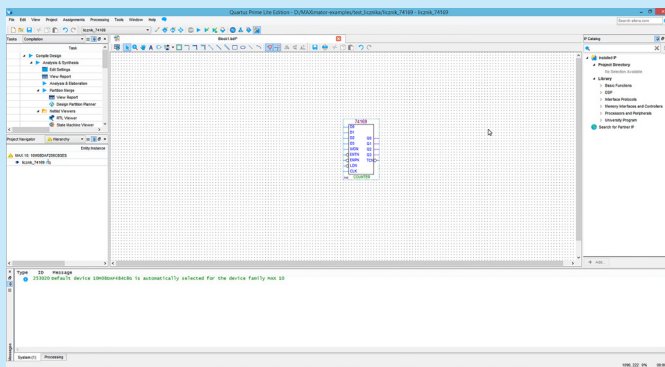


Rysunek 8. Tworzenie planszy schematu

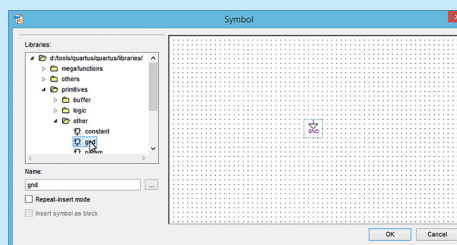
Po narysowaniu linii połączeniowych przypisujemy do nich symbole wejść i wyjść, do czego służy narzędzie *Pin Tool* dostępne w menu (**rysunek 15**). Dostępne są w nim trzy opcje: wejście, wyjście i linia dwukierunkowa. Wybieramy typy właściwe dla poszczególnych linii, w razie wątpliwości kierując się opisem symbolu bibliotecznego (**rysunek 16** – kolumna *Directions*), który można przyciemnić klikając na symbol prawym przyciskiem myszy i wybierając opcję *Properties>Ports*. Na **rysunku 17** pokazano planszę schematu z narysowanymi połączeniami i przypisanymi od nich symbolami wejść i wyjść zgodnie z kierunkami linii licznika 74169.

W kolejnym kroku nadamy własne nazwy liniom I/O. W tym celu należy dwukrotnie kliknąć na symbolu linii lub naciskając prawy przycisk myszy wybrać opcję *Properties*, co spowoduje wyświetlenie okna pokazanego na **rysunku 18**. W linii *Pin name(s)* wprowadzamy nazwy poszczególnych linii, które powinny być łatwe do skojarzenia, są bowiem używane nie tylko podczas implementacji projektu w układzie FPGA, ale także analizie funkcjonalnej i czasowej projektu – chodzi o to, aby nazwa wyraźnie sugerowała funkcję i pochodzenie sygnału. Na **rysunku 19** pokazano proponowane nazwy sygnałów.

Kolejnym krokiem jest wstępna kompilacja projektu, co jest niezbędne do weryfikacji poprawności



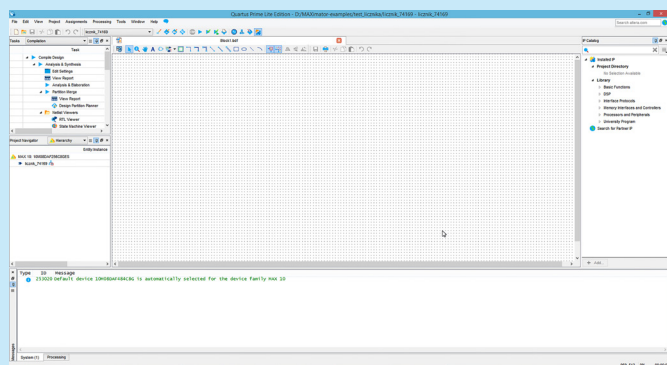
Rysunek 12. Ułożenie komponentu na planszy



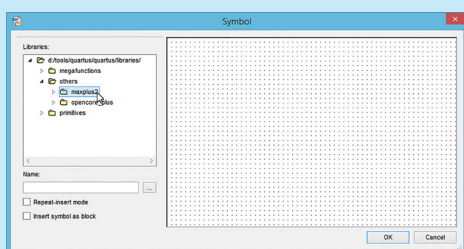
Rysunek 13. Pobranie symbolu masy

przygotowanego opisu, a także wygenerowaniu przez narzędzie do syntezy logicznej listy sygnałów oraz linii I/O. Jest ona nam niezbędna do ustalenia przypisań linii I/O do fizycznych wyprowadzeń układu FPGA. Kompilację uruchamia przycisk pokazany na **rysunku 20**.

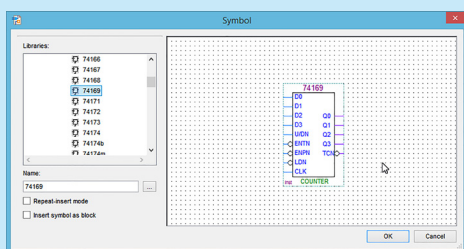
Przypisania linii I/O do fizycznych wyprowadzeń układu FPGA wykonuje się za pomocą specjalnego edytora o nazwie *Pin Planner*, który jest uruchamiany za pomocą dwukrotnego kliknięcia opcji *Pin Planner* (**rysunek 21**). Otwarte okno edytora przypisań pokazano na **rysunku 22**. Najważniejsza dla nas na tym etapie będzie tabela ulokowana w dolnej części okna, w której wylistowane są linie I/O, ich kierunkowość, przypisania do fizycznych wyprowadzeń zaproponowane wstępnie przez fitter (kolumna *Fitter Location*), domyślny standard napięciowy linii I/O, a także kilku innych parametrach mniej istotnych dla naszego projektu. Modyfikacje poszczególnych opcji są możliwe po dwukrotnym kliknięciu w wybrane okno tabeli. Zaczynamy od ustalenia standardu napięciowego linii, zmieniając w każdym wierszu domyślną wartość 2,5 V na 3,3-V LVTTTL. Następnie w kolumnie *Location* przypisujemy wyprowadzenia implementowanego licznika w taki sposób, żeby wykorzystać przycisk zerujący oraz przycisk L (do zmiany kierunku zliczania) zamontowane na shieldzie, diody LED zamontowane na płytce MAXimatora do monitorowania stanów wyjść licznika, a sygnał zegarowy na jednej z wolnych linii I/O złącza Arduino. Ustalenie oznaczeń fizycznych linii FPGA, do których będą przypisane sygnały projektu ułatwi mapa wyprowadzeń (**rysunek 23**) dostępna do pobrania na stronie *maximator-fpga.org*. Przypisania ustalone dla projektu przedstawiono na **rysunku 24**. Dodanie przypisań do fizycznych wyprowadzeń FPGA po ich wprowadzeniu staje się także widoczne na schemacie, jak to widać na **rysunku 25**.



Rysunek 9. Czysta plansza schematu

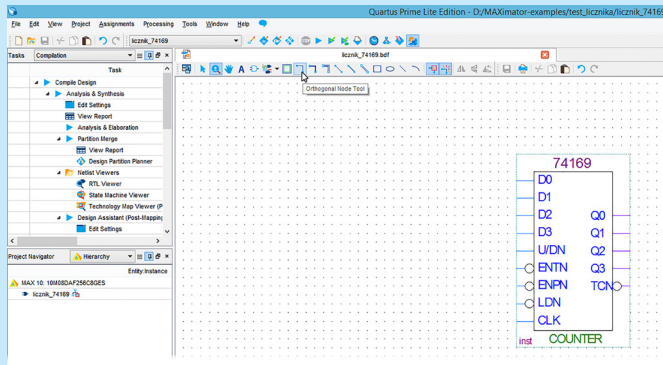


Rysunek 10. Okno dostępu do predefiniowanych symboli

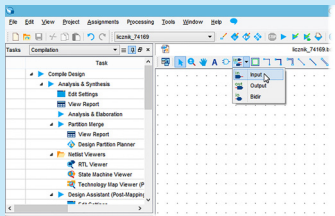


Rysunek 11. Wybór komponentu 74169

Szukasz supportu?
 Projekty przykładowe, dokumentacje oraz filmy poświęcone MAXimatorowi są dostępne na stronie www.maximator-fpga.org.



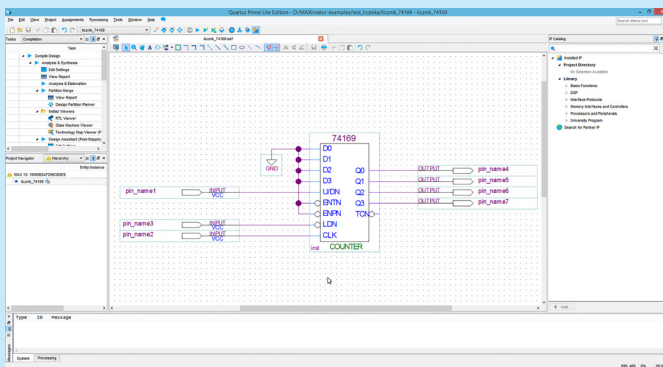
Rysunek 14. Połączenia wykonujemy za pomocą narzędzia Orthogonal Node Tool



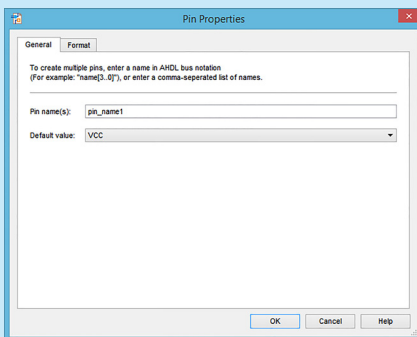
Rysunek 15. Narzędzie Pin Tool

Name	Alias	Inversion	Status	Direction	Hide Alias
1	CLK	None	Used	INPUT	No
2	D0	None	Used	INPUT	No
3	D1	None	Used	INPUT	No
4	D2	None	Used	INPUT	No
5	D3	None	Used	INPUT	No
6	ENTN	None	Used	INPUT	No
7	ENTN	None	Used	INPUT	No
8	ENTN	None	Used	INPUT	No
9	ENTN	None	Used	INPUT	No
10	LDN	None	Used	INPUT	No
11	LDN	None	Used	INPUT	No
12	LDN	None	Used	INPUT	No
13	LDN	None	Used	INPUT	No
14	TCN	None	Used	OUTPUT	No
15	TCN	None	Used	OUTPUT	No
16	TCN	None	Used	OUTPUT	No
17	TCN	None	Used	OUTPUT	No

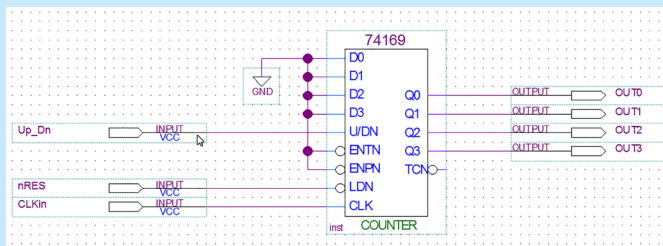
Rysunek 16. Wybieramy typy właściwe dla poszczególnych linii



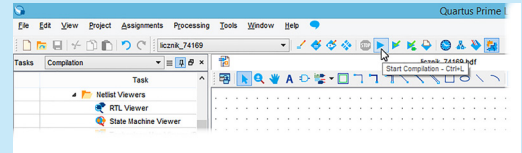
Rysunek 17. Plansza schematu z narysowanymi połączeniami i przypisanymi od nich symbolami wejść i wyjść



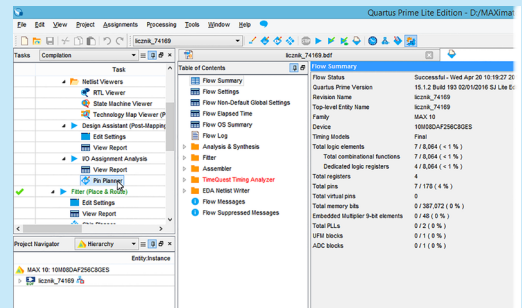
Rysunek 18. Okno Properties



Rysunek 19. Proponowane nazwy sygnałów



Rysunek 20. Przycisk uruchamiający kompilator

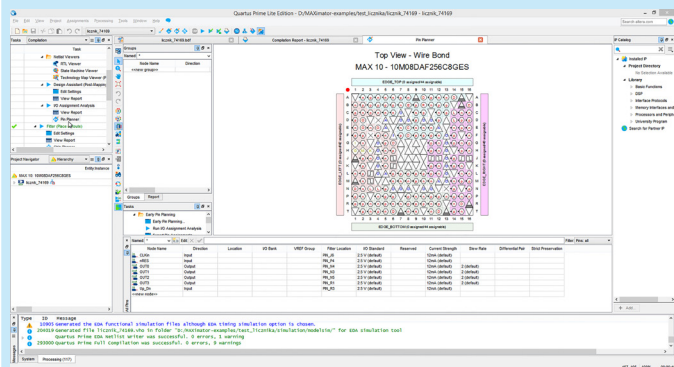


Rysunek 21. Uruchomienie edytora Pin Planner

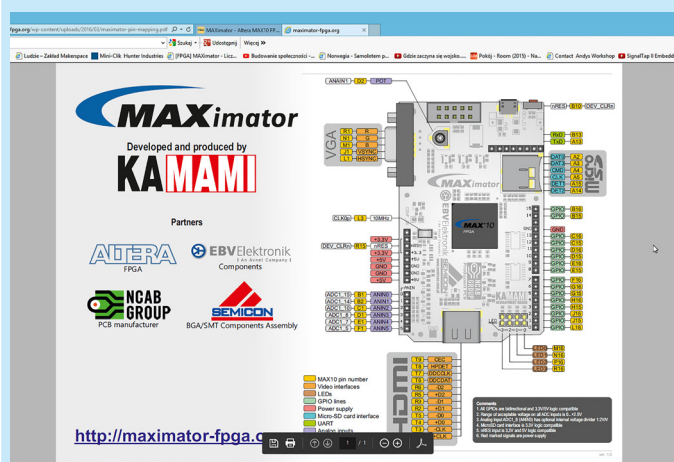
Po wprowadzeniu przypisani ponownie kompilujemy projekt (za pomocą przycisku pokazanego na rysunku 20) i w przypadku poprawnego jej zakończenia możemy zaprogramować układ FPGA z zestawie MAXimator. W tym celu musimy mieć dołączony do komputera programator USB Blaster (wchodzący w skład zestawu MAXimator – przypominam o tym, że zazwyczaj konieczne jest ręczne wskazanie systemowi ścieżki do driverów programatora, co zostało opisane w EP4/2016), a sam programator za pomocą 10-żyłowego przewodu złącza JTAG do płytki MAXimator z dołączonym zasilaniem.

Program obsługujący programator jest uruchamiany za pomocą ikony w menu pokazanej na **rysunku 26**. Jego okno pokazano na **rysunku 27**. Jeżeli programator USB Blaster nie zostanie samoczynnie wykryty przez pakiet Quartus, klikamy przycisk *Hardware Setup* i w wyświetlonym oknie, w zakładce *Hardware Settings* wybieramy w oknie wyboru *Currently selected hardware* aktywny programator (**rysunek 28**). W kolejnym kroku naciskamy w oknie obsługującym programator przycisk *Add File...* (**rysunek 29**) i w wyświetlonym oknie wskazujemy plik wynikowy, który użyjemy do programowania FPGA. Pliki wynikowe są domyślnie ulokowane w podkatalogu o nazwie *output_files* ulokowanym w katalogu projektu. Quartus domyślnie generuje dwa rodzaje plików:

- o rozszerzeniu nazwy *sof*, który jest używany do bezpośredniej konfiguracji pamięci RAM w FPGA. Jego użycie zalecamy podczas eksperymentów z implementacją projektów, częsta modyfikacja zawartości RAM nie wpływa na żywotność pamięci konfiguracji Flash,
 - o rozszerzeniu nazwy *pof*, który służy do zapisania konfiguracji projektu w nieulotnej pamięci konfiguracji typu Flash. Jego użycie zalecamy do zaprogramowania FPGA ostateczną wersją implementowanego projektu.
- Dodanie pliku *sof* powoduje wyświetlenie symbolu układu w oknie programu obsługującego programator (**rysunek 30**) z domyślnie



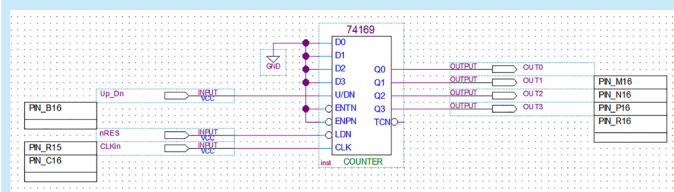
Rysunek 22. Otwarte okno edytora przypisań



Rysunek 23. Ustalenie oznaczeń fizycznych linii FPGA

Node Name	Direction	Location	ID Bank	VREF Group	Filter Location	ID Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Stitch Preservation
CLK0	Input	PR1_C16	5	BS_H0	PR1_R1	3.3V LVTTL		8mA (default)			
rRES	Input	PR1_R15	5	BS_H0	PR1_R4	3.3V LVTTL		8mA (default)			
OUT0	Output	PR1_P16	5	BS_H0	PR1_R4	3.3V LVTTL		8mA (default)	2 (default)		
OUT1	Output	PR1_P16	5	BS_H0	PR1_R4	3.3V LVTTL		8mA (default)	2 (default)		
OUT2	Output	PR1_P16	5	BS_H0	PR1_R4	3.3V LVTTL		8mA (default)	2 (default)		
OUT3	Output	PR1_P16	5	BS_H0	PR1_R1	3.3V LVTTL		8mA (default)	2 (default)		
IN0	Input	PR1_P16	6	BS_H0	PR1_R1	3.3V LVTTL		8mA (default)			

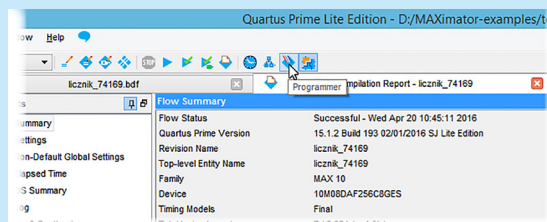
Rysunek 24. Przypisania ustalone dla projektu



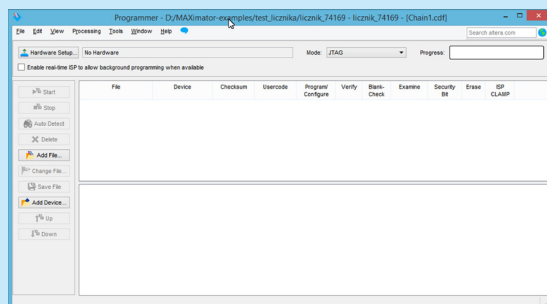
Rysunek 25. Dodanie przypisań do fizycznych wyprowadzeń FPGA

aktywną opcją *Program/Configure*, co umożliwia szybkie zapisanie konfiguracji do pamięci RAM FPGA. W przypadku wybrania pliku *pof* dostępne opcje są nieco inne (rysunek 31), omówimy je w kolejnych odcinkach kursu.

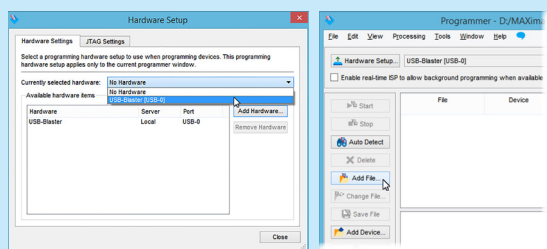
Przygotowanie i zapisanie do pamięci konfiguracji FPGA projektu przygotowanego według opisu gwarantuje jego poprawne działanie, przy czym należy pamiętać o dostarczeniu do linii D13 (C16) sygnału zegarowego o częstotliwości kilkunastu...kilkuset Hz, który umożliwi obserwację na LED pracy licznika. Zestaw MAXimator jest wyposażony w dwa własne źródła sygnałów zegarowych (generator kwarcowy 10 MHz i wewnętrzny generator zegarowy wbudowany w układ FPGA MAX10), ale częstotliwości generowanych przez nie sygnałów są zbyt duże, żeby można było zaobserwować zmiany stanów LED. Można zastosować preskaler dzielący częstotliwość sygnału generatora kwarcowego



Rysunek 26. Uruchomienie programatora

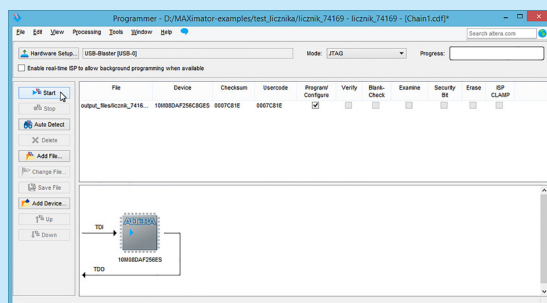


Rysunek 27. Okno programatora



Rysunek 28. Wybranie aktywnego programatora

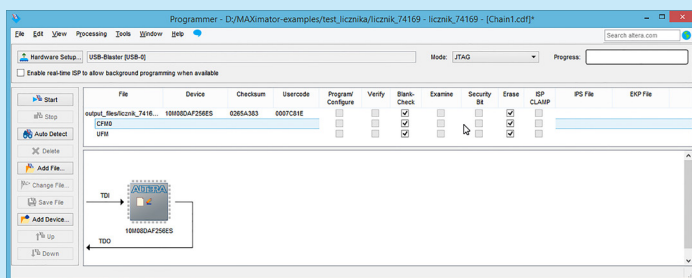
Rysunek 29. Wskazanie pliku do programowania FPGA



Rysunek 30. Dodanie pliku sof powoduje wyświetlenie symbolu układu w oknie programu obsługującego programator

do wartości >200 Hz, co umożliwi obserwację pracy licznika bez konieczności stosowania dodatkowego generatora sygnału zegarowego – temu poświęcamy nasz konkurs.

Piotr Zbysiński, EP



Rysunek 31. W przypadku wybrania pliku pof dostępne opcje są nieco inne