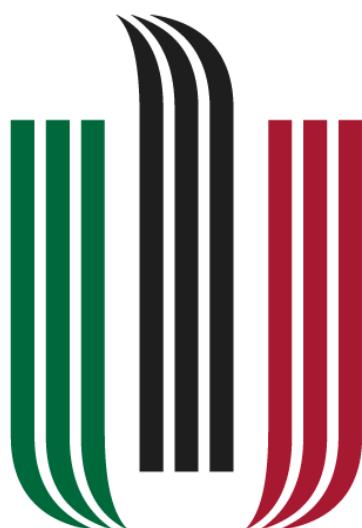


**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**

Wydział Informatyki, Elektroniki i Telekomunikacji
Katedra Informatyki



A G H

PROJEKT INŻNIERSKI

Uniwersalny moduł sprzętowego
przetwarzania danych oparty na FPGA

DOKUMENTACJA UŻYTKOWNIKA

Autor: Krzysztof Papciak
Opiekun: dr inż. Jacek Długopolski

Spis treści

1. Wstęp	3
2. Budowa i elementy zestawu SNF-0.....	4
2. 1. Opis elementów zestawu.....	4
2. 2. Opis elementów płytki PCB modułu SNF-0	5
2. 3. Schemat blokowy zestawu SNF-0.....	7
2. 4. Szczegółowe parametry podzespołów modułu SNF-0.....	7
3. Instrukcja uruchomienia i programowania modułu SNF-0.....	10
3.1. Peryferia sprzętowe.....	10
3.2. Zalecane oprogramowanie	10
3.3. Tworzenie projektu w programie Altera Quartus II	13
3.4. Przypisanie portów wejścia/wyjścia i ustawień układu FPGA.....	14
3.5. Programowanie układu FPGA	22
4. Opis przykładowego procesora graficznego.....	24
4.1. Opis procesora graficznego i przygotowanie do jego uruchomienia	24
4.2. Kompilacja i uruchomienie projektu na module SNF-0	24
4.3. Sterowanie przy pomocy klawiatury PS2.....	25
5. Instrukcja przygotowania własnych modeli do wyświetlenia przy pomocy procesora graficznego..	26
5.1. Przygotowanie modelu w programie Autodesk 3DS Max	26
5.2. Eksport modelu przy użyciu skryptu.....	27
5.3. Przygotowanie do wyświetlenia modelu na module SNF-0.....	28
5. Spis rysunków	30

1. Wstęp

Niniejszy dokument stanowi instrukcję użytkownika, prezentującą sposób uruchomienia i używania zestawu *SNF-0* wykonanego, jako projekt inżynierski *Uniwersalnego modułu sprzętowego przetwarzania danych opartego na FPGA*.

Przedstawione zostały elementy modułu uruchomieniowego, ich specyfikacja, opis dostępnych portów oraz podstawowe uwagi dotyczące zestawu.

Zaprezentowano proces tworzenia nowego projektu, który ma zostać uruchomiony na zestawie SNF-0 oraz jego komplikacji i zaprogramowania układu FPGA.

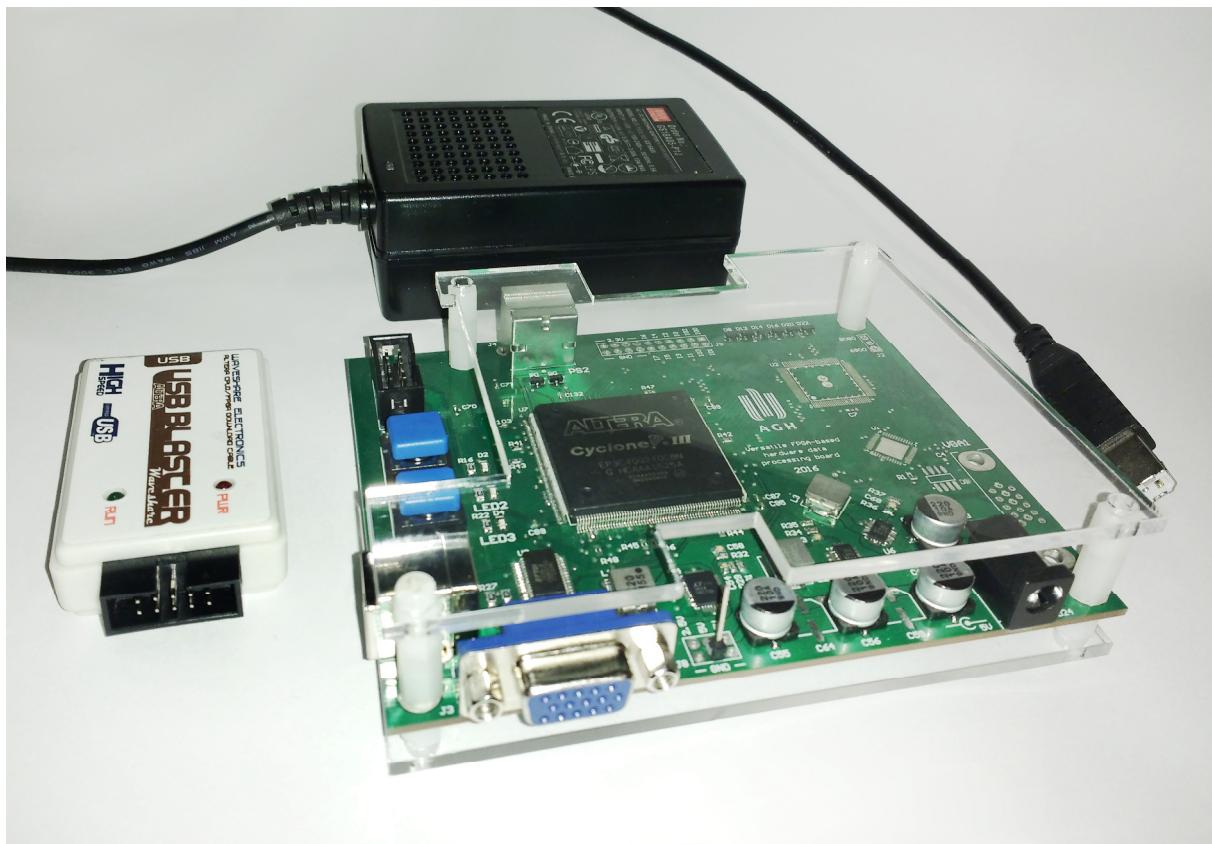
Kolejna część instrukcji opisuje sposób komplikacji i uruchomienia przykładowego procesora graficznego, który został stworzony w celu prezentacji możliwości zestawu.

Przykładowy procesor umożliwia wyświetlenie prostego modelu 3D, jako siatka krawędzi. Użytkownik może kontrolować obrót bryły oraz jej skalowanie, za pomocą klawiatury.

Ostatnia część instrukcji użytkownika przedstawia proces przygotowania własnego modelu 3D do wyświetlania na zestawie.

2. Budowa i elementy zestawu SNF-0

2. 1. Opis elementów zestawu

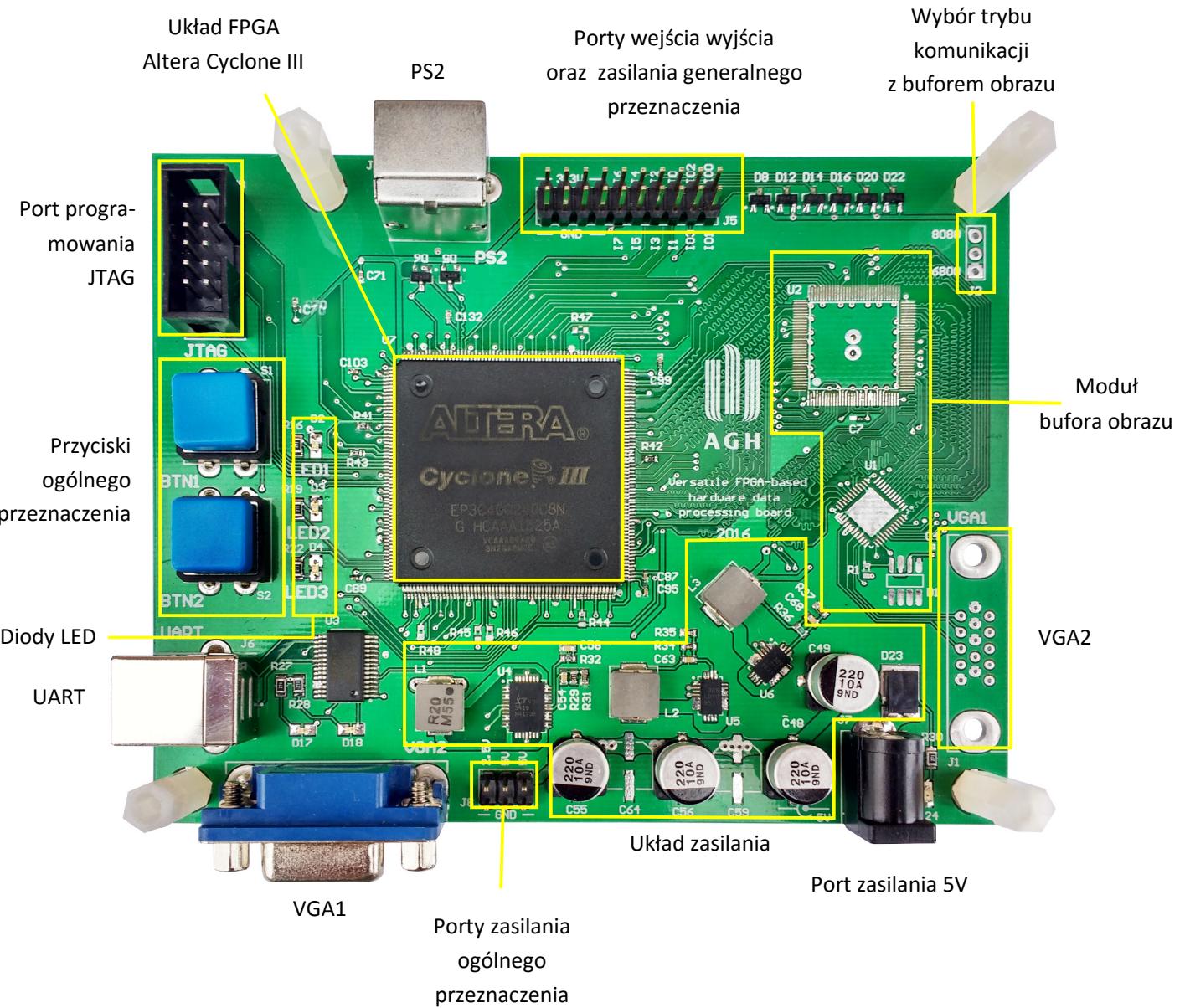


Rys. 1. Elementy zestawu uruchomieniowego SNF-0

Zestaw uruchomieniowy składa się z następujących elementów:

- moduł SNF-0
- programator zgodny z USB-Blaster II
- zasilacz 5V
- kabel USB-B do komunikacji przez port szeregowy

2. 2. Opis elementów płytki PCB modułu SNF-0



Rys. 2. Porty i układy modułu uruchomieniowego SNF-0

Zestaw SNF-0 posiada podstawowe porty wejścia/wyjścia do komunikacji z użytkownikiem i z komputerem. Kontrolki i przyciski pozwalają na proste sterowanie działaniem tworzonych projektów. Porty VGA umożliwiają zwizualizowanie wyników oraz tworzenie projektów generujących grafikę. Jednym z nich jest przykładowy projekt procesora graficznego, wykonany, jako część projektu inżynierskiego. Złącze PS2 zapewnia wygodną obsługę poleceń użytkownika za pomocą klawiatury lub myszki. Porty wejścia/wyjścia generalnego przeznaczenia oraz wyprowadzone napięcia, dostępne na płytce modułu dają możliwość podłączenia zewnętrznych modułów, a także wykorzystanie nowych typów portów komunikacyjnych. Niektóre z układów zostaną wykorzystane w przyszłości do innych projektów, dlatego nie zostały jeszcze zamontowane.

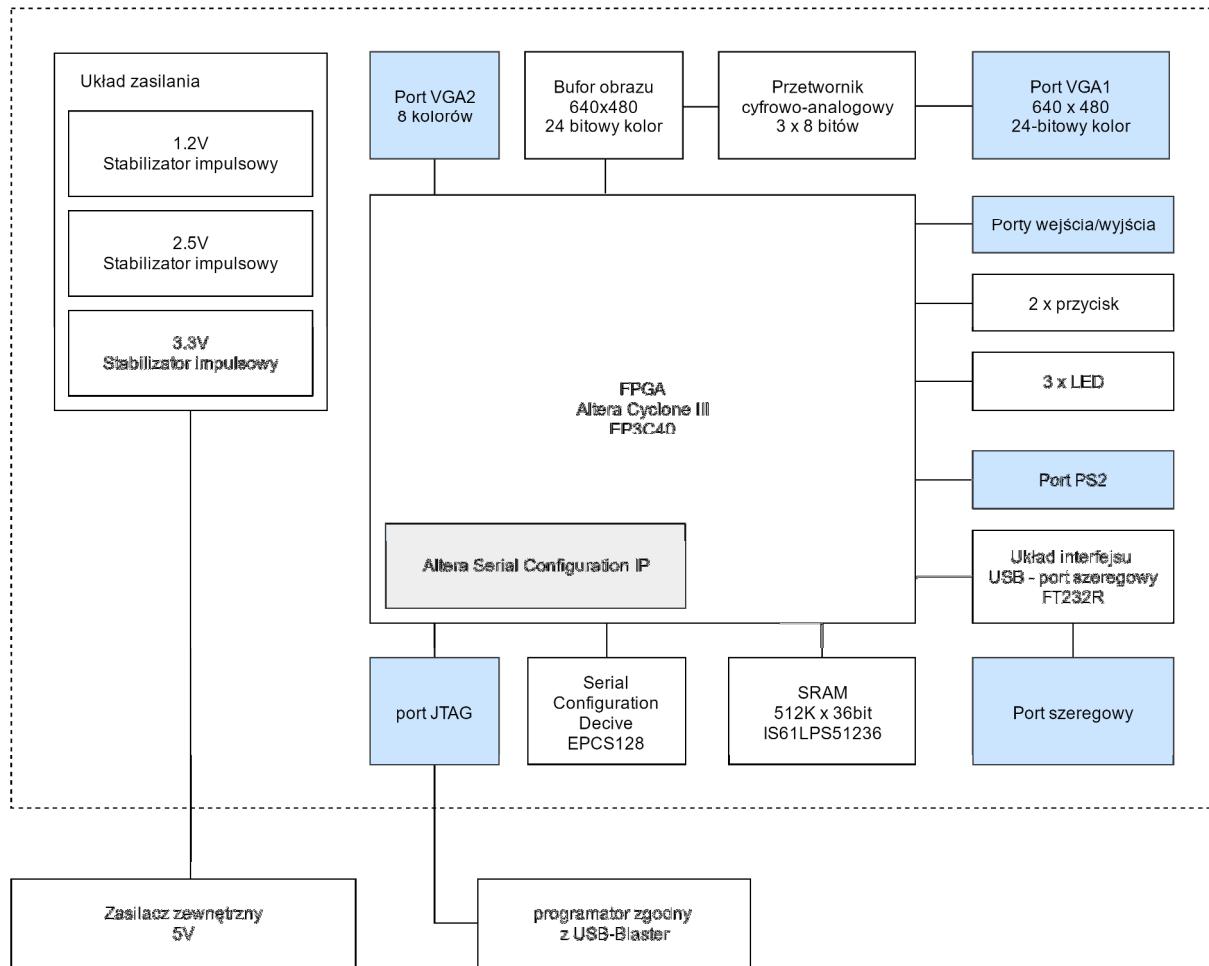
Dostępne porty:

- VGA1 (8 kolorów, dowolna rozdzielcość)
- VGA2 (24-bitowa głębia kolorów, dostępny bufor obrazu, maksymalna rozdzielcość: 640x480)
- PS2 (możliwa obsługa klawiatury i myszki)
- Port szeregowy, zrealizowany za pomocą konwertera z interfejsu USB
- JTAG (port programowania układu FPGA oraz pamięci FLASH przechowującej konfigurację)
- Porty wejścia/wyjścia oraz wejścia ogólnego przeznaczenia zabezpieczone diodami
- Wyprowadzenia dostępnych napięć (3.3V, 2.5V oraz 5V) do zasilania zewnętrznych modułów
- Port zasilania 5V

Układy dostępne na płytce SNF-0:

- Układ FPGA Altera Cyclone III EP3C40
- Pamięć EEPROM konfiguracji układu FPGA (zamiennik EPICS128)
- 2 przyciski ogólnego przeznaczenia
- 3 żółte diody LED
- Oscylator 50MHz podłączony do wejść zegarowych układu FPGA
- Układ interfejsu portu szeregowego do USB
- Bufor i kontroler obrazu SSD1963 (niezamontowany)
- 24-bitowy przetwornik cyfrowo-analogowy dla wyjścia VGA2 (niezamontowany)
- 2-megabajtowa pamięć SRAM (niezamontowana)

2. 3. Schemat blokowy zestawu SNF-0



Rys. 3. Schemat blokowy modułu

2. 4. Szczegółowe parametry podzespołów modułu SNF-0

Układ FPGA EP3C40

- 39 600 jednostek logicznych
- 1 161 216 bitów pamięci RAM (126 bloków M9K)
- 126 18-bitowych układów mnożących
- 4 układy PLL
- 128 portów wejścia/wyjścia
- Obudowa PQFP240

Moduł programowania

- Programowanie w trybie JTAG
- 128-megabitowa pamięć EEPROM (*Serial Configuration Device*) do przechowywania konfiguracji układu FPGA (zgodna z EPCS128)
- Możliwość programowania układu *Serial Configuration Device* przy użyciu portu JTAG
- Zewnętrzny programator zgodny z Altera USB-Blaster

Pamięć SRAM

- Pojemność: 2MB
- Szerokość magistrali danych: 32 bity + 4 bity parzystości
- *Burst mode* – pozwala na wydajniejsze przesyłanie porcji danych
- Maksymalna częstotliwość pracy: 200MHz

Przyciski i kontrolki LED

- 3 diody LED generalnego przeznaczenia
- 2 przyciski
- Zastosowano prosty filtr do eliminacji efektu odbicia styków

Wyjście VGA1

- 8 bitów na każdy kanał koloru
- Używa bufora obrazu mogącego pomieścić maksymalnie obraz o rozdzielcości 864 x 480 i 24-bitowej głębi koloru
- Bufor może komunikować się z układem FPGA przy użyciu dwóch trybów 6800 i 8080
- Bufor posiada także dodatkowe opcje zmiany parametrów obrazu, takie jak kontrola jasności, kontrastu, nasycenia i obrotu
- Wyjście bufora połączone do 3 kanałowego, 8 bitowego przetwornika cyfrowo-analogowego

Wyjście VGA2

- 1 bit na każdy kanał koloru
- Sterowany bezpośrednio przez układ FPGA (używając diod zabezpieczających)
- 8 kolorów możliwych do uzyskania
- Możliwość użycia dowolnej rozdzielczości

Porty szeregowe

- Port szeregowy zgodny ze standardem RS-232
- Moduł SNF-0 zawiera wbudowany konwerter port szeregowy – USB, wykorzystujący układ *FTDI FT232R*, co pozwala na łatwe podłączenie modułu do nowszych komputerów nieposiadających złącza standardu RS-232
- Port PS2, pozwalający na podłączenie klawiatury lub myszki

Porty wejścia/wyjścia ogólnego przeznaczenia

- 4 porty wejścia/wyjścia
- 8 portów wejściowych
- 4 wyprowadzenia napięcia zasilania 3.3V do zasilenia zewnętrznych modułów
- 1 wyprowadzenie napięcia 2.5V
- 2 wyprowadzenia napięcia 5V

Układ zasilania

- Napięcie wejściowe: 5V
- 3 stabilizatory impulsowe
- Stabilizatory 2.5V i 3.3V, 6A
- Stabilizator 1.2V, 8A

3. Instrukcja uruchomienia i programowania modułu SNF-0

3.1. Peryferia sprzętowe

- Komputer z systemem Microsoft Windows
- Programator zgodny z USB-Blaster
- Monitor ze złączem VGA
- Klawiatura lub myszka z portem PS2
- Kabel USB-B
- Zasilacz sieciowy 5V

Oprócz modułu SNF-0 w zestawie dostępny jest zasilacz sieciowy o napięciu wyjściowym 5V, który ze względów bezpieczeństwa, należy podłączyć do modułu przed innymi peryferiami.

Aby w pełni wykorzystać możliwości modułu SNF-0 potrzebny jest monitor ze złączem VGA, obsługujący rozdzielcość minimum 640 x 480 pikseli i potrafiący wyświetlać kolorowy obraz. Złącze monitora należy podpiąć do wyjścia VGA2 modułu uruchomieniowego.

Opcjonalne jest podłączenie modułu do komputera za pomocą kabla USB-B, które umożliwia komunikację z układem FPGA poprzez port szeregowy. Wymagane jest posiadanie odpowiednich sterowników oraz programu do komunikacji przez port szeregowy. Instalacja odpowiedniego oprogramowania zostanie omówiona w następnym podrozdziale.

Do modułu SNF-0 można podpiąć także klawiaturę lub myszkę z kablem PS2.

Przed przystąpieniem do programowania układu FPGA, należy przygotować programator zgodny z Altera *USB-Blaster* (dołączony do zestawu). Urządzenie należy podłączyć do portu JTAG na płytce modułu SNF-0 oraz do portu USB komputera. Instalacja sterowników i programowania układu FPGA zostanie opisana w dalszej części instrukcji.

3.2. Zalecane oprogramowanie

Do odpowiedniego wykorzystania zestawu, potrzebne będzie następujące oprogramowanie:

- System Microsoft Windows XP lub nowszy
- Altera Quartus II z bibliotekami Cyclone III (do wersji 13.1)
- Sterownik Altera USB-Blaster (dołączony do oprogramowania Quartus II)
- Sterownik interfejsu USB FTDI
- Autodesk 3DS Max 2010 lub nowszy (opcjonalnie)

- Terminal wspierający obsługę portu szeregowego (np. Realterm) (opcjonalnie)

Wspierane wersje oprogramowania Quartus II

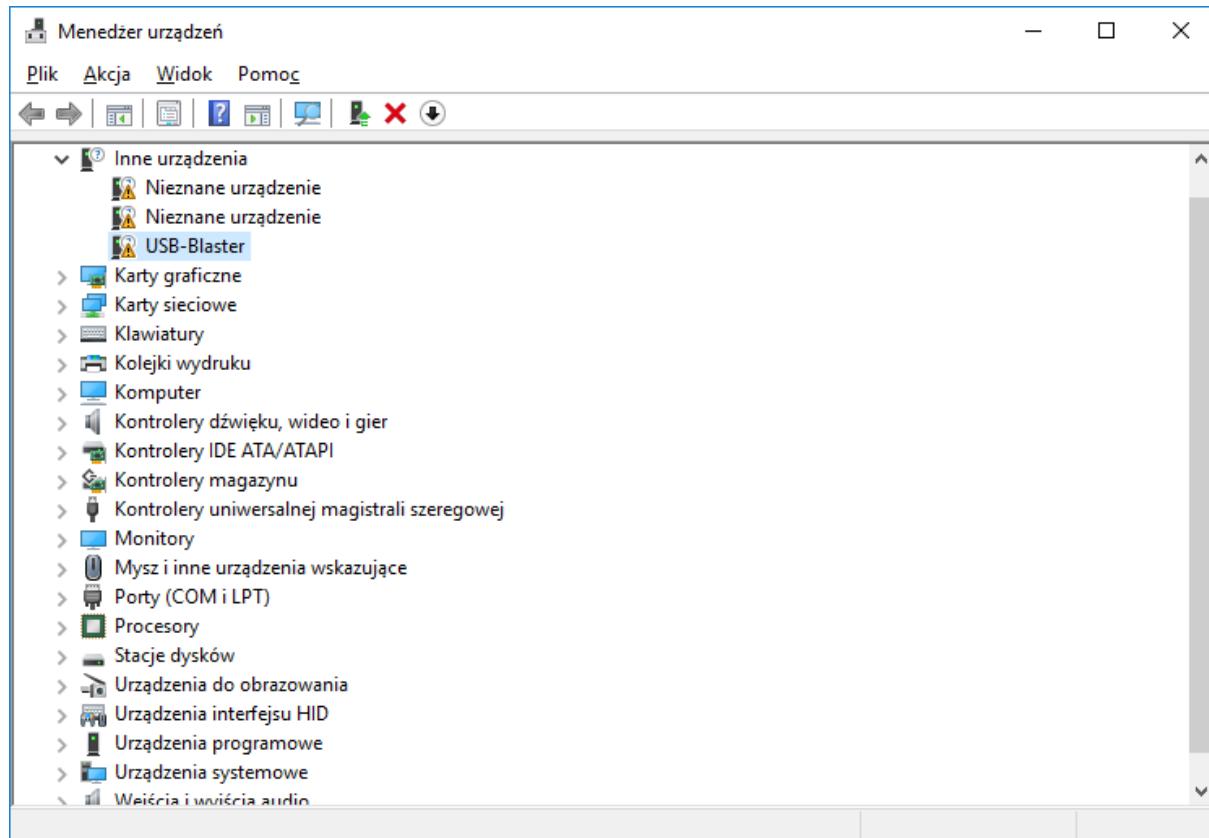
Do programowania i przygotowywania konfiguracji układu FPGA może zostać wykorzystane oprogramowanie *Quartus II* firmy *Altera* w wersji *Web Edition*, jak również w *Subscription Edition*.

Należy także zainstalować biblioteki dodające wsparcie dla układów FPGA *Cyclone III*. Maksymalna wersja programu *Quartus II* posiadająca takie wsparcie to 13.1.

Instalacja sterownika *Altera USB-Blaster*

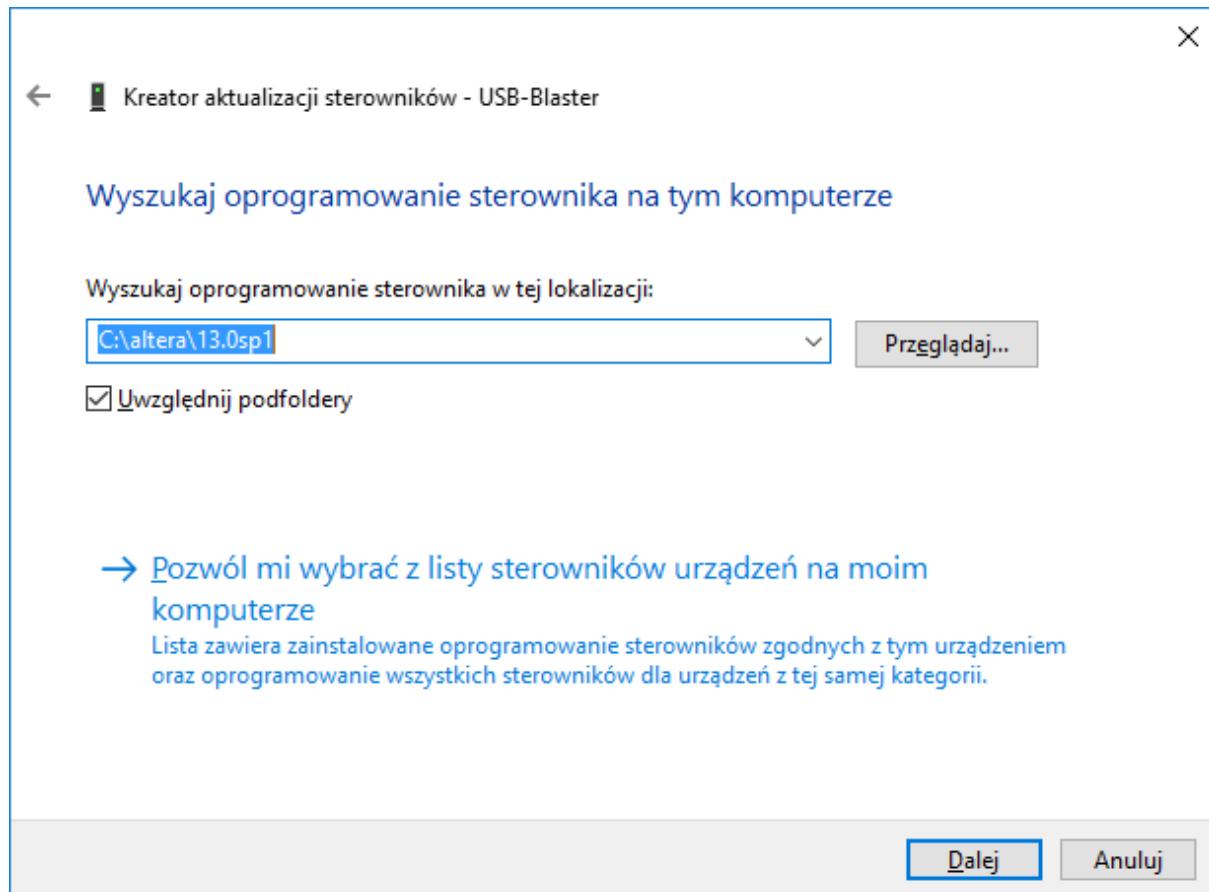
W celu instalacji sterownika programatora zgodnego z *USB-Blaster*, można skorzystać ze sterowników dostarczonych razem z oprogramowaniem *Quartus II*.

Po pierwszym podłączeniu urządzenia do złącza USB komputera, uruchamiamy *Manager urządzeń*. Pozyция *USB-Blaster* powinna pojawić się w zakładce *inne urządzenia*.



Rys. 4. Menedżer urządzeń po pierwszym podłączeniu programatora USB-Blaster

Aby zainstalować sterowniki, należy kliknąć prawym przyciskiem myszy i wybrać opcję *Aktualizuj oprogramowanie sterownika...*. Następnym krokiem jest wybranie opcji *Przeglądaj mój komputer w poszukiwaniu oprogramowania sterownika* i wskazanie ścieżki do folderu instalacji oprogramowania *Altera Quartus II*.



Rys. 5. Wybór lokalizacji sterownika USB-Blaster

Po ukończonej instalacji, sterownik powinien znaleźć się w kategorii *Kontrolery uniwersalnej magistrali szeregowej* menadżera urządzeń.

Instalacja sterownika interfejsu FTDI

Aby móc skorzystać z komunikacji przy pomocy portu szeregowego należy zainstalować sterowniki układu *FT232R* firmy *FTDI Chips*.

Sterowniki należy pobrać ze strony producenta układu (odpowiednie dla danej wersji systemu operacyjnego):

<http://www.ftdichip.com/Drivers/D2XX.htm> (dostęp 12.10.2016)

Po zapisaniu sterowników na dysku, należy wypakować archiwum, a następnie zainstalować sterowniki w analogiczny sposób jak sterowniki urządzenia *USB-Blaster*.

3.3. Tworzenie projektu w programie Altera Quartus II

Aby utworzyć nowy projekt, przeznaczony do uruchomienia na module SNF-0, należy otworzyć oprogramowanie *Altera Quartus II*, a następnie wybrać opcję *File -> New project wizard*.

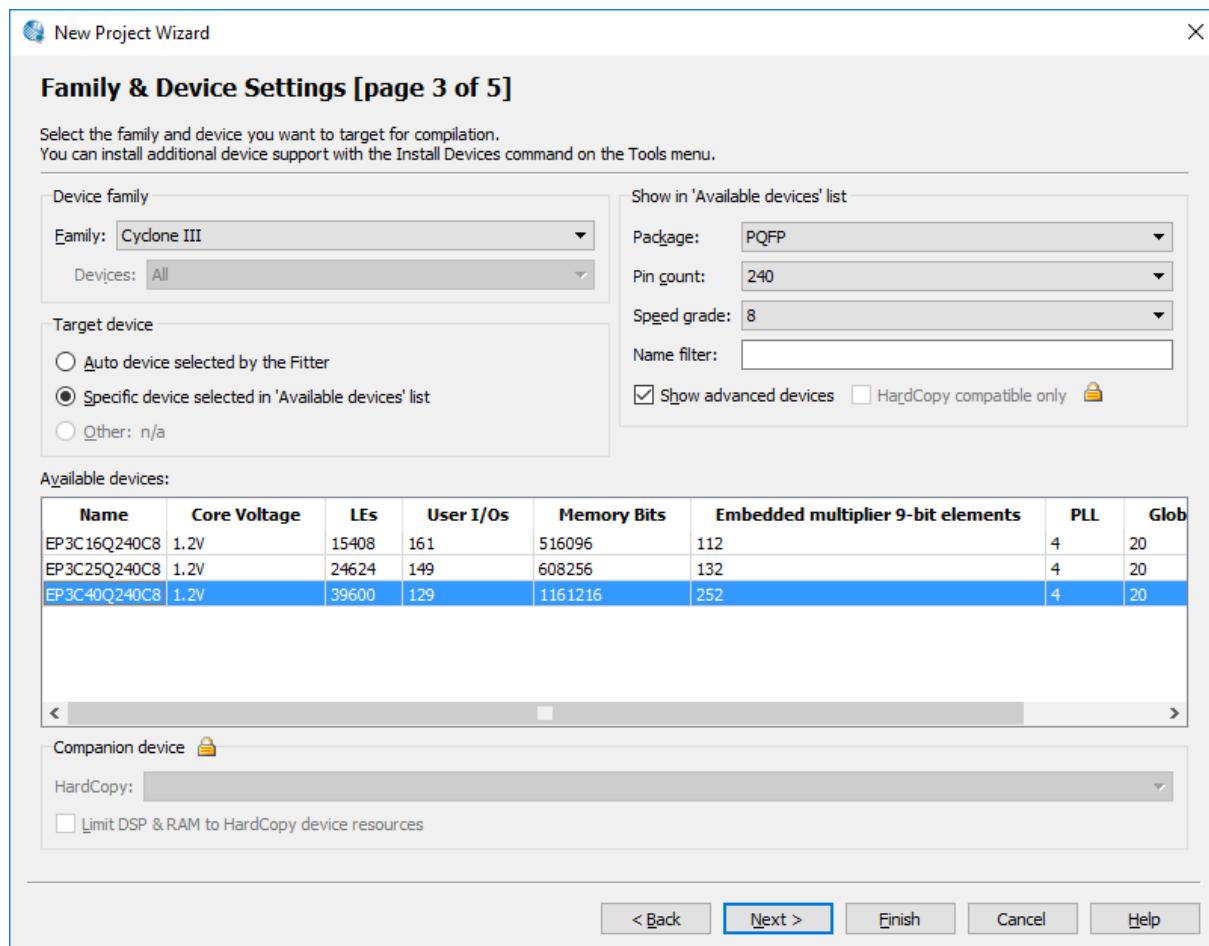
Po wybraniu nazwy i folderu, w którym ma zostać utworzony projekt, przechodzimy do konfiguracji rodziny i ustawień układu FPGA (strona *Family & Device Settings*). Należy wybrać opcję *Cyclone III* na liście rozwijanej *Family* w grupie *Device family*. Aby przefiltrować listę dostępnych układów, można wybrać dodatkowe opcje w polu *Show in 'available devices' list*:

Package: PQFP

Pin count: 240

Speed grade: 8

Z listy *available devices* należy wybrać układ: *EP3C40Q240C8*, a następnie dokonczyć tworzenie projektu.

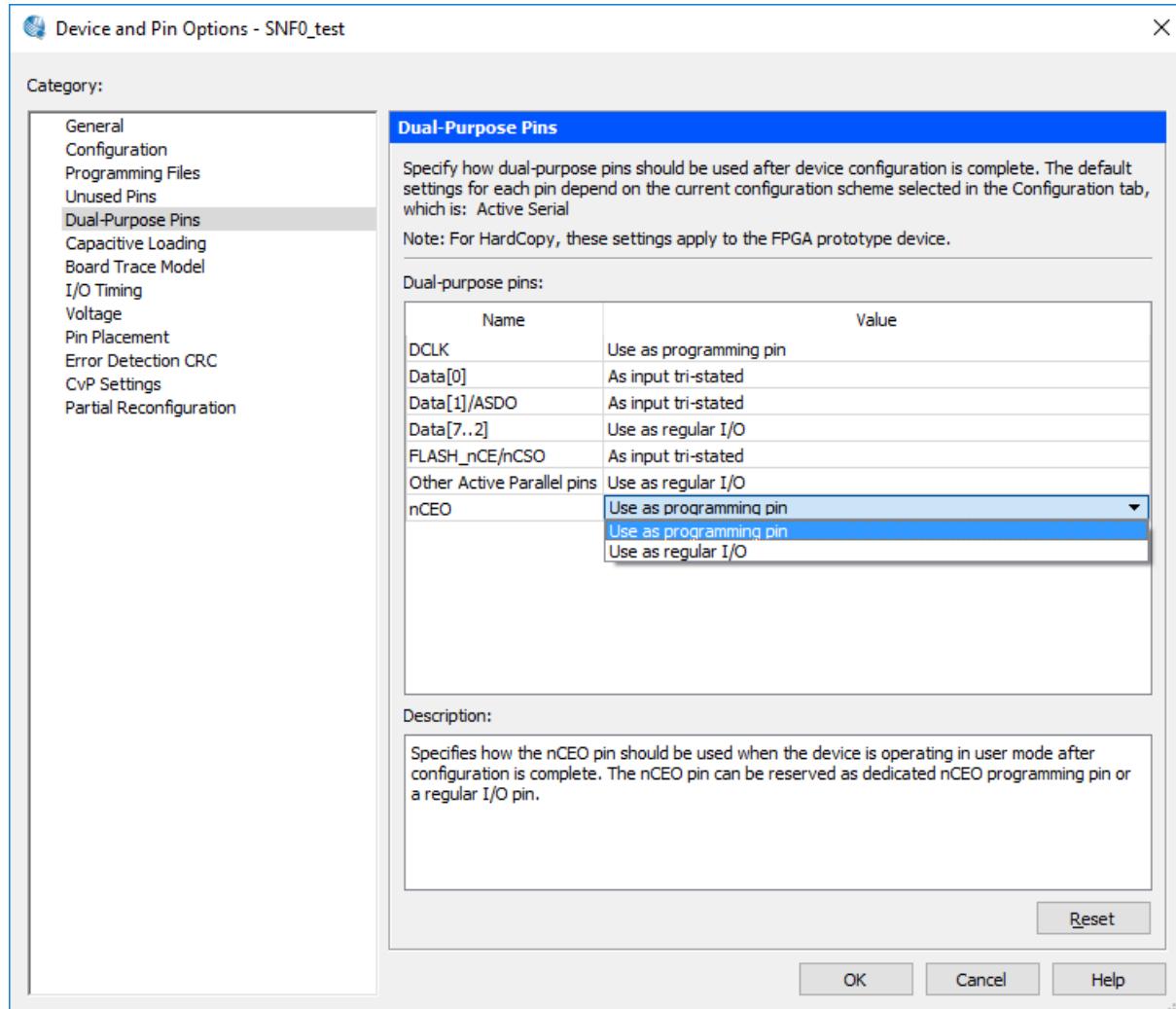


Rys. 6. Wybór układu FPGA podczas tworzenia nowego projektu w programie Altera Quartus II

3.4. Przypisanie portów wejścia/wyjścia i ustawień układu FPGA

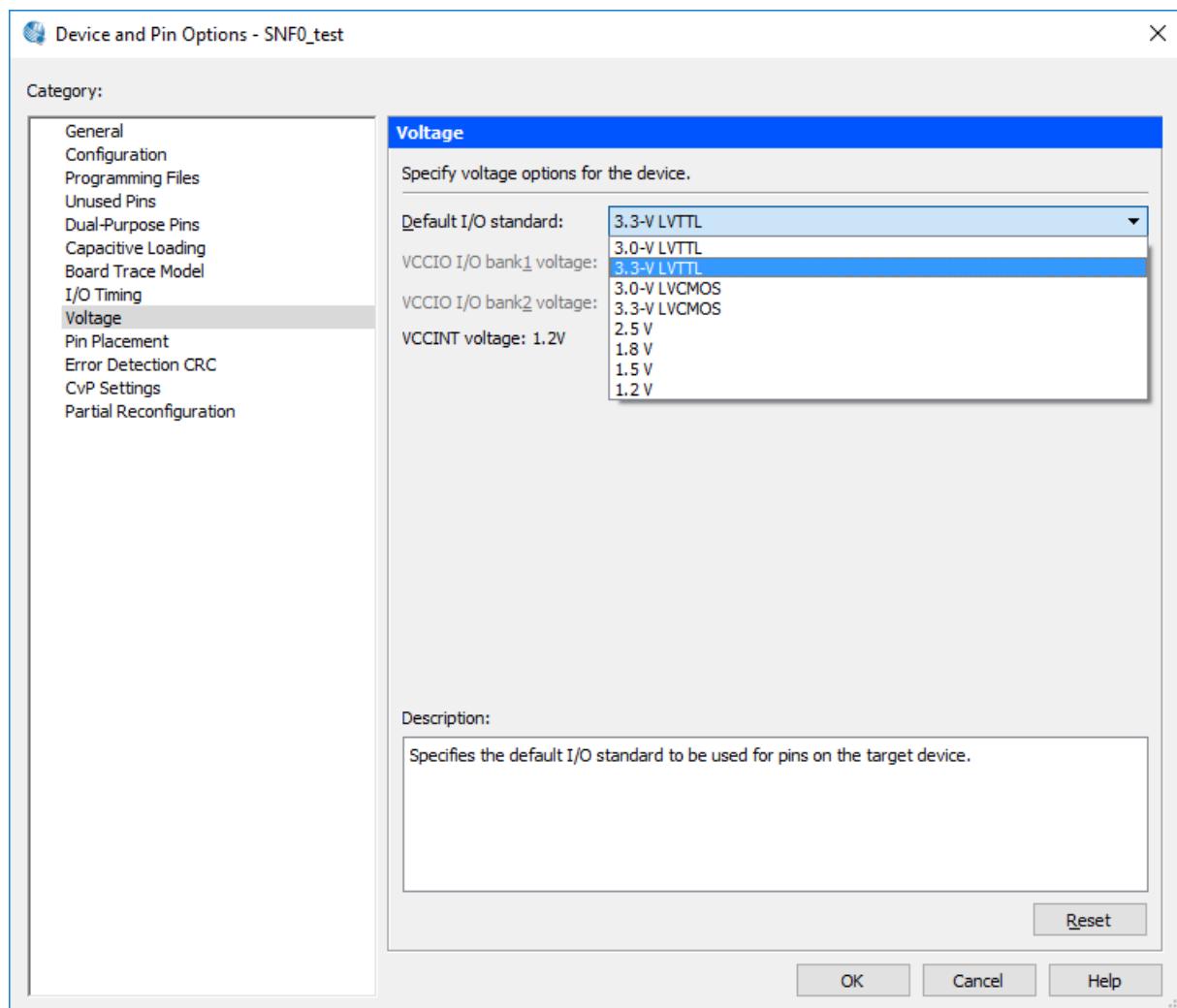
Ustawienia układu FPGA

Po stworzeniu nowego projektu, należy skonfigurować ustawienia układu FPGA. Aby to zrobić, wybieramy z menu *Assignments* opcję *Device...*. W otwartym oknie wybieramy z listy po lewej opcję *Dual-purpose pins*. Wybieramy opcję *Use as programming pin* dla wyprowadzenia *nCEO*.



Rys. 7. Ustawienie funkcji wyprowadzenia nCEO w opcjach układu FPGA

Następnie należy wybrać kategorię *Voltage* i wybrać z listy rozwijanej *Default I/O standard* opcję *3.3-V LVTTL*.

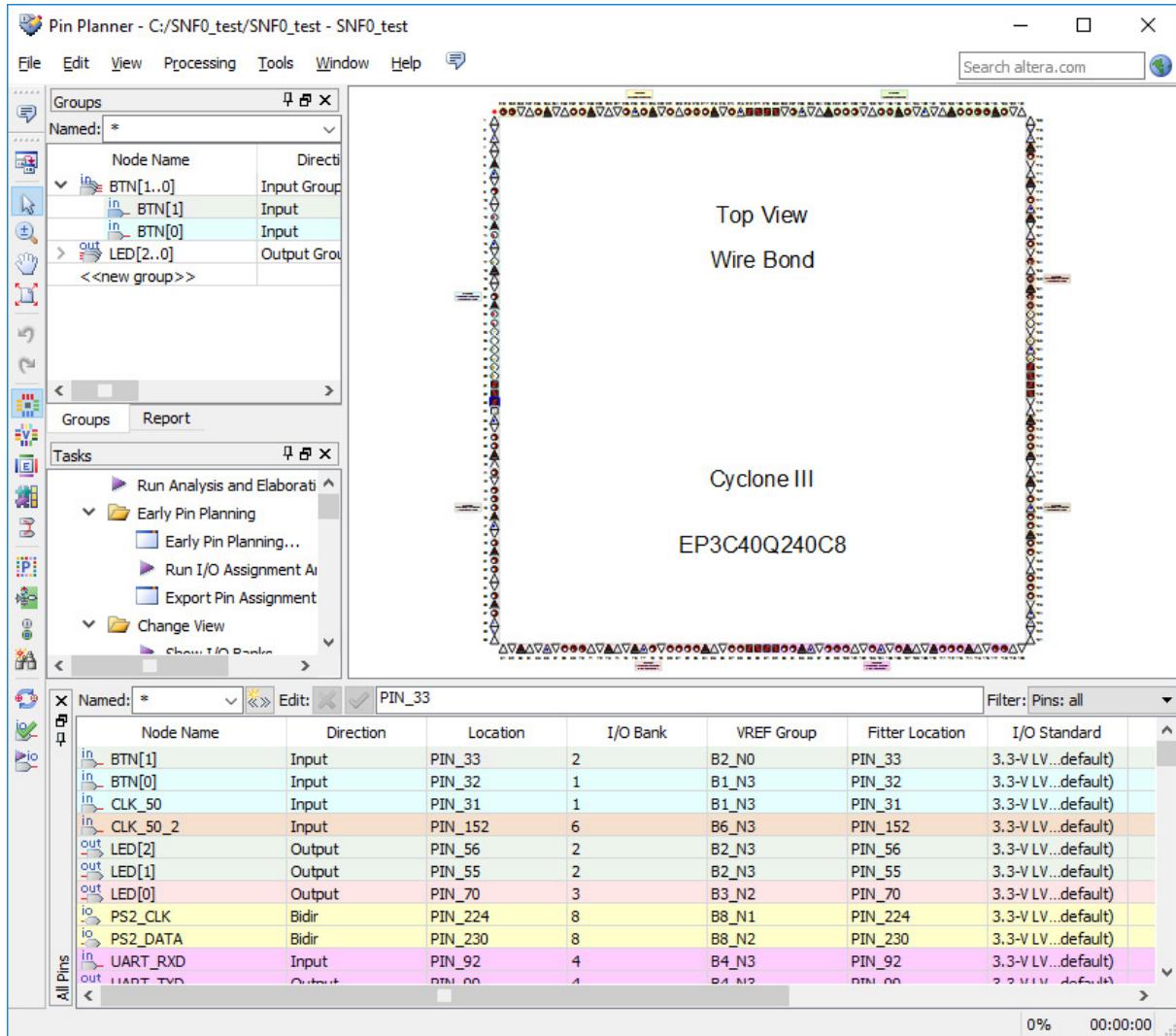


Rys. 8. Ustawienie standardu wejścia/wyjścia w konfiguracji układu FPGA

Przypisanie nazw portów wejścia/wyjścia i ich konfiguracji

Aby przypisać nazwy i funkcje portów do odpowiednich pinów układu FPGA, należy otworzyć narzędzie *Pin Planer* dostępne z menu *Assignments* programu Quartus II. W dolnej części okna znajduje się lista, w której można przypisać konfigurację potrzebnych wyprowadzeń.

Pełna lista wyprowadzeń i odpowiadających im funkcji znajduje się w tabeli Tab. 1.



Rys. 9. Okno konfiguracji wyprowadzeń układu FPGA

Ważne, aby przy każdym wyprowadzeniu wybrać standard napięcia zasilania wejścia/wyjścia **3.3-V LVTTL**.

Sygnały zegarowe

Nazwa sygnału	Kierunek	Wyprowadzenie układu FPGA	Opis
CLK_50		PIN_31	Wejście zegarowe 50MHz
CLK_50_2		PIN_152	Wejście zegarowe 50MHz

Tab. 1. Wyprowadzenia sygnałów zegarowych układu FPGA

Przyciski i diody LED

Nazwa sygnału	Kierunek	Wyprowadzenie układowu FPGA	Opis
BTN[0]	I	PIN_32	Przycisk 0
BTN[1]	I	PIN_33	Przycisk 1
LED[0]	O	PIN_70	Dioda LED 0
LED[1]	O	PIN_55	Dioda LED 1
LED[2]	O	PIN_56	Dioda LED 2

Porty wejścia/wyjścia ogólnego przeznaczenia

Nazwa sygnału	Kierunek	Wyprowadzenie układowu FPGA	Opis
GPIO[0]	I/O	PIN_203	Port wejścia/wyjścia ogólnego przeznaczenia 0
GPIO[1]	I/O	PIN_207	Port wejścia/wyjścia ogólnego przeznaczenia 1
GPIO[2]	I/O	PIN_200	Port wejścia/wyjścia ogólnego przeznaczenia 2
GPIO[3]	I/O	PIN_223	Port wejścia/wyjścia ogólnego przeznaczenia 3
GPI[0]	I	PIN_89	Port wejścia ogólnego przeznaczenia 0
GPI[1]	I	PIN_90	Port wejścia ogólnego przeznaczenia 1
GPI[2]	I	PIN_91	Port wejścia ogólnego przeznaczenia 2
GPI[3]	I	PIN_211	Port wejścia ogólnego przeznaczenia 3
GPI[4]	I	PIN_212	Port wejścia ogólnego przeznaczenia 4
GPI[5]	I	PIN_149	Port wejścia ogólnego przeznaczenia 5
GPI[6]	I	PIN_150	Port wejścia ogólnego przeznaczenia 6
GPI[7]	I	PIN_151	Port wejścia ogólnego przeznaczenia 7

Porty szeregowe

Nazwa sygnału	Kierunek	Wyprowadzenie układu FPGA	Opis
UART_RXD	I	PIN_92	Wejście portu szeregowego
UART_TXD	O	PIN_99	Wyjście portu szeregowego
PS2_CLK	I/O	PIN_224	Sygnal zegarowy portu PS2
PS2_DATA	I/O	PIN_230	Sygnal danych portu PS2

Wyjście VGA2

Nazwa sygnału	Kierunek	Wyprowadzenie układu FPGA	Opis
VGA2_R	O	PIN_76	Sygnal koloru czerwonego portu VGA2
VGA2_G	O	PIN_78	Sygnal koloru zielonego portu VGA2
VGA2_B	O	PIN_80	Sygnal koloru niebieskiego portu VGA2
VGA2_VSync	O	PIN_63	Sygnal synchronizacji pionowej portu VGA2
VGA2_HSync	O	PIN_73	Sygnal synchronizacji poziomej portu VGA2

Wyjście VGA1

Nazwa sygnału	Kierunek	Wyprowadzenie układu FPGA	Opis
VGA1_R[0]	O	PIN_188	Kolor czerwony portu VGA1, bit 0
VGA1_R[1]	O	PIN_187	Kolor czerwony portu VGA1, bit 1
VGA1_R[2]	O	PIN_186	Kolor czerwony portu VGA1, bit 2
VGA1_R[3]	O	PIN_185	Kolor czerwony portu VGA1, bit 3
VGA1_R[4]	O	PIN_184	Kolor czerwony portu VGA1, bit 4
VGA1_R[5]	O	PIN_139	Kolor czerwony portu VGA1, bit 5
VGA1_R[6]	O	PIN_142	Kolor czerwony portu VGA1, bit 6
VGA1_R[7]	O	PIN_143	Kolor czerwony portu VGA1, bit 7
VGA1_G[0]	O	PIN_144	Kolor zielony portu VGA1, bit 0

VGA1_G[1]	O	PIN_145	Kolor zielony portu VGA1, bit 1
VGA1_G[2]	O	PIN_146	Kolor zielony portu VGA1, bit 2
VGA1_G[3]	O	PIN_214	Kolor zielony portu VGA1, bit 3
VGA1_G[4]	O	PIN_216	Kolor zielony portu VGA1, bit 4
VGA1_G[5]	O	PIN_218	Kolor zielony portu VGA1, bit 5
VGA1_G[6]	O	PIN_159	Kolor zielony portu VGA1, bit 6
VGA1_G[7]	O	PIN_160	Kolor zielony portu VGA1, bit 7
VGA1_B[0]	O	PIN_161	Kolor niebieski portu VGA1, bit 0
VGA1_B[1]	O	PIN_162	Kolor niebieski portu VGA1, bit 1
VGA1_B[2]	O	PIN_176	Kolor niebieski portu VGA1, bit 2
VGA1_B[3]	O	PIN_177	Kolor niebieski portu VGA1, bit 3
VGA1_B[4]	O	PIN_164	Kolor niebieski portu VGA1, bit 4
VGA1_B[5]	O	PIN_166	Kolor niebieski portu VGA1, bit 5
VGA1_B[6]	O	PIN_169	Kolor niebieski portu VGA1, bit 6
VGA1_B[7]	O	PIN_173	Kolor niebieski portu VGA1, bit 7
VGA1_PIXEL_CLK	I	PIN_209	Sygnal zegara pikseli bufora obrazu
VGA1_TE	I	PIN_210	Wejście sygnału <i>Tearing Effect</i> bufora obrazu
VGA1_CS_n	O	PIN_196	Aktywacja bufora obrazu (<i>Chip select</i>)
VGA1_DC_n	O	PIN_183	Sygnal wyboru danych lub polecenie
VGA1_E_n	O	PIN_128	Sygnal aktywacji w trybie 6800 bufora obrazu
VGA1_RESET_n	O	PIN_195	Reset bufora obrazu
VGA1_RW_n	O	PIN_133	Wybór trybu odczytu lub zapisu

Sygnały pamięci SRAM

Nazwa sygnału	Kierunek	Wyprowadzenie układu FPGA	Opis
SRAM_DQ[0]	I/O	PIN_226	Port danych 0
SRAM_DQ[1]	I/O	PIN_221	Port danych 1
SRAM_DQ[2]	I/O	PIN_219	Port danych 2
SRAM_DQ[3]	I/O	PIN_217	Port danych 3
SRAM_DQ[4]	I/O	PIN_202	Port danych 4

SRAM_DQ[5]	I/O	PIN_201	Port danych 5
SRAM_DQ[6]	I/O	PIN_197	Port danych 6
SRAM_DQ[7]	I/O	PIN_194	Port danych 7
SRAM_DQ[8]	I/O	PIN_171	Port danych 8
SRAM_DQ[9]	I/O	PIN_137	Port danych 9
SRAM_DQ[10]	I/O	PIN_135	Port danych 10
SRAM_DQ[11]	I/O	PIN_134	Port danych 11
SRAM_DQ[12]	I/O	PIN_132	Port danych 12
SRAM_DQ[13]	I/O	PIN_131	Port danych 13
SRAM_DQ[14]	I/O	PIN_127	Port danych 14
SRAM_DQ[15]	I/O	PIN_126	Port danych 15
SRAM_DQ[16]	I/O	PIN_98	Port danych 16
SRAM_DQ[17]	I/O	PIN_94	Port danych 17
SRAM_DQ[18]	I/O	PIN_93	Port danych 18
SRAM_DQ[19]	I/O	PIN_88	Port danych 19
SRAM_DQ[20]	I/O	PIN_87	Port danych 20
SRAM_DQ[21]	I/O	PIN_82	Port danych 21
SRAM_DQ[22]	I/O	PIN_81	Port danych 22
SRAM_DQ[23]	I/O	PIN_57	Port danych 23
SRAM_DQ[24]	I/O	PIN_52	Port danych 24
SRAM_DQ[25]	I/O	PIN_51	Port danych 25
SRAM_DQ[26]	I/O	PIN_49	Port danych 26
SRAM_DQ[27]	I/O	PIN_45	Port danych 27
SRAM_DQ[28]	I/O	PIN_43	Port danych 28
SRAM_DQ[29]	I/O	PIN_41	Port danych 29
SRAM_DQ[30]	I/O	PIN_38	Port danych 30
SRAM_DQ[31]	I/O	PIN_37	Port danych 31
SRAM_PAR[0]	I/O	PIN_231	Bit parzystości pierwszego bajtu danych
SRAM_PAR[1]	I/O	PIN_103	Bit parzystości drugiego bajtu danych
SRAM_PAR[2]	I/O	PIN_100	Bit parzystości trzeciego bajtu danych
SRAM_PAR[3]	I/O	PIN_95	Bit parzystości czwartego bajtu danych
SRAM_ADDR[0]	O	PIN_21	Adres, bit 0

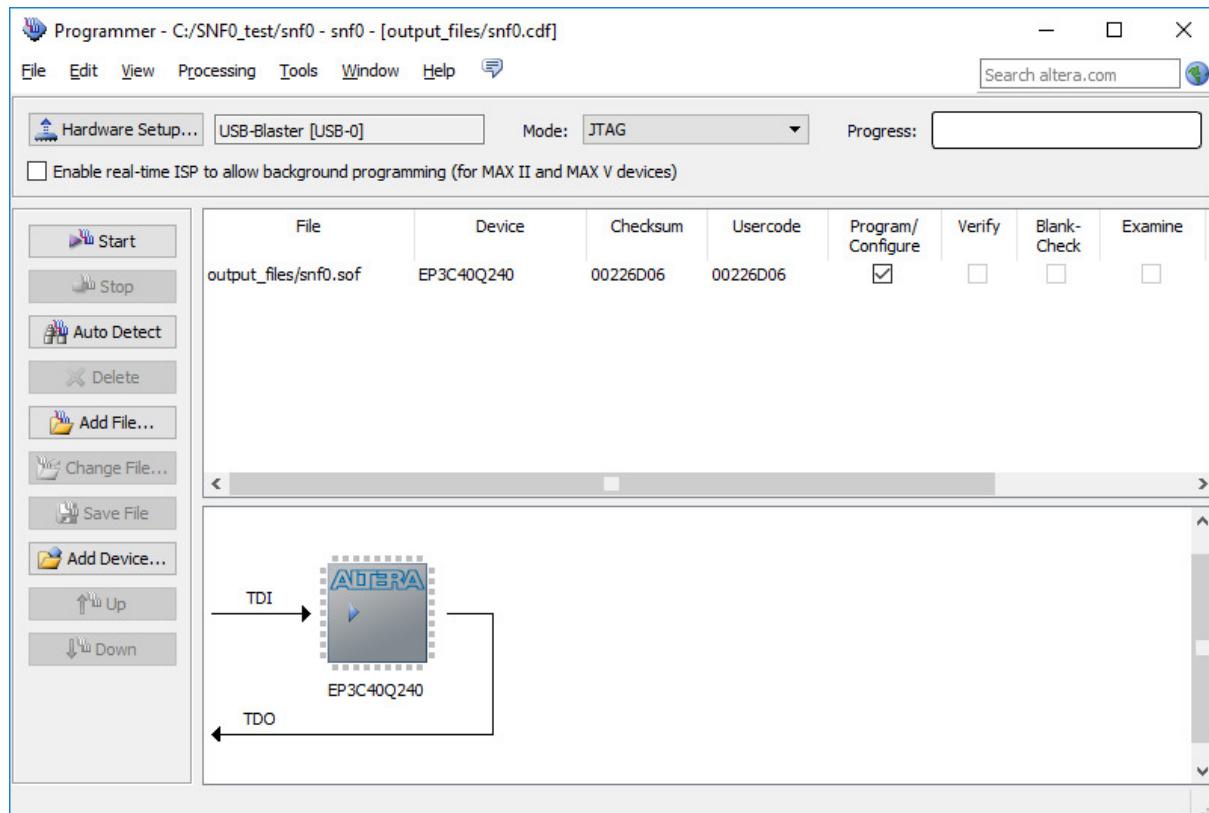
SRAM_ADDR[1]	O	PIN_22	Adres, bit 1
SRAM_ADDR[2]	O	PIN_50	Adres, bit 2
SRAM_ADDR[3]	O	PIN_46	Adres, bit 3
SRAM_ADDR[4]	O	PIN_44	Adres, bit 4
SRAM_ADDR[5]	O	PIN_39	Adres, bit 5
SRAM_ADDR[6]	O	PIN_18	Adres, bit 6
SRAM_ADDR[7]	O	PIN_13	Adres, bit 7
SRAM_ADDR[8]	O	PIN_9	Adres, bit 8
SRAM_ADDR[9]	O	PIN_6	Adres, bit 9
SRAM_ADDR[10]	O	PIN_240	Adres, bit 10
SRAM_ADDR[11]	O	PIN_239	Adres, bit 11
SRAM_ADDR[12]	O	PIN_236	Adres, bit 12
SRAM_ADDR[13]	O	PIN_235	Adres, bit 13
SRAM_ADDR[14]	O	PIN_232	Adres, bit 14
SRAM_ADDR[15]	O	PIN_118	Adres, bit 15
SRAM_ADDR[16]	O	PIN_117	Adres, bit 16
SRAM_ADDR[17]	O	PIN_107	Adres, bit 17
SRAM_ADDR[18]	O	PIN_106	Adres, bit 18
SRAM_CLK	O	PIN_69	Sygnal zegarowy pamięci SRAM
SRAM_MODE	O	PIN_68	Wybór sekwencji <i>Burst Mode</i>
SRAM_ADV_n	O	PIN_114	Sygnal kontroli adresu <i>Burst Mode</i>
SRAM_ADSC_n	O	PIN_112	Address Status Processor (<i>Burst Mode</i>)
SRAM_ADSP_n	O	PIN_113	Address Status Cache Controller (<i>Burst Mode</i>)
SRAM_BWE_n	O	PIN_110	Aktywacja zapisu danych
SRAM_OE_n	O	PIN_111	Aktywacja odczytu z pamięci
SRAM_CE2_n	O	PIN_84	Sygnal <i>Chip select 2</i>
SRAM_CE_n	O	PIN_83	Sygnal <i>Chip select</i>
SRAM_ZZ	O	PIN_189	Tryb oszczędzania energii

W repozytorium projektu, w folderze *HDL/SNFO_empty* znajduje się skonfigurowany pusty projekt, który można wykorzystać, jako bazę do rozbudowy.

3.5. Programowanie układu FPGA

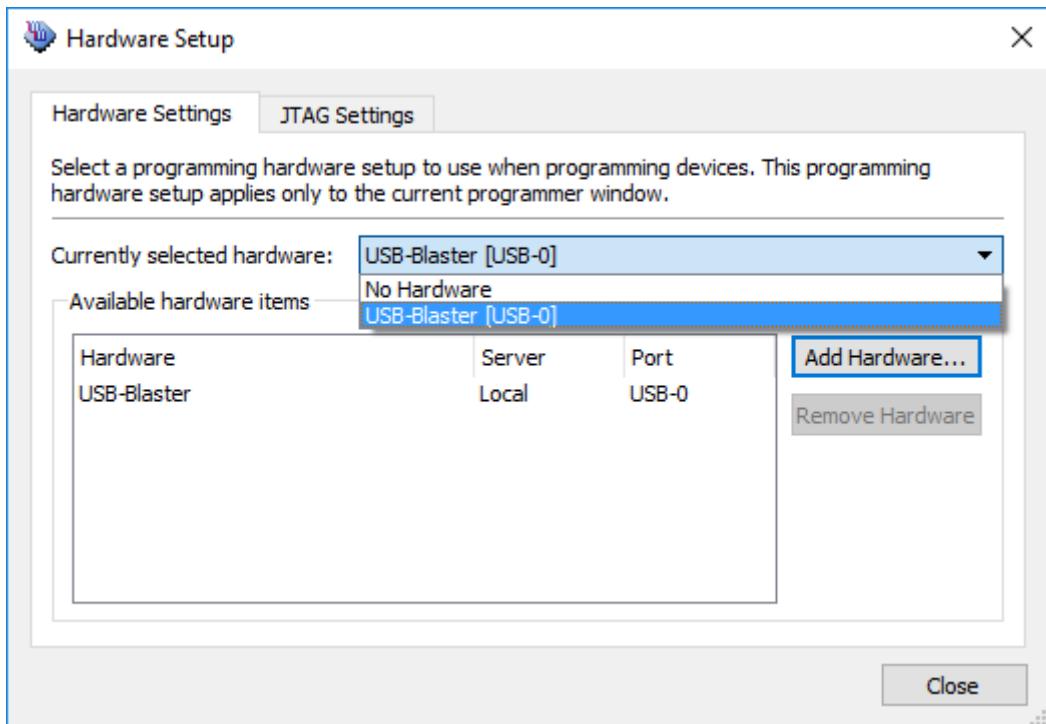
Aby zaprogramować układ FPGA, przy użyciu oprogramowania Altera Quartus II, należy najpierw podpiąć programator zgodny *USB-Blaster* (o ile jest taka potrzeba, zainstalować sterowniki).

Następnie, z menu *Tools*, wybrać opcję *Programmer*. Pokaż się okno programowania.



Rys. 10. Okno programowania układu FPGA

Jeżeli w polu obok przycisku *Hardware setup* nie wyświetli się nazwa programatora *USB-Blaster*, należy wybrać go z menu, wywoływanego poprzez naciśnięcie tego przycisku.



Rys. 11. Wybór programatora

Po wybraniu odpowiedniego pliku z programem, należy nacisnąć przycisk Start. Układ FPGA powinien zostać zaprogramowany.

4. Opis przykładowego procesora graficznego

4.1. Opis procesora graficznego i przygotowanie do jego uruchomienia

Specjalizowany procesor graficzny wyświetlający prostą grafikę 3D został stworzony, aby zaprezentować możliwości układu FPGA i portów zewnętrznych.

Aby uruchomić ten przykład, należy przygotować monitor VGA i klawiaturę PS2, zgodnie z opisem z podrozdziału 3.1.

Częścią składową projektu specjalizowanego procesora graficznego są moduły napisane w języku VHDL, obsługujące wyjście VGA oraz porty PS2 i port szeregowy UART. Można je wykorzystać do tworzenia nowych projektów opartych o moduł SNF-0. Szczegółowy opis tych jednostek projektowych znajduje się w dokumentacji technicznej projektu inżynierskiego.

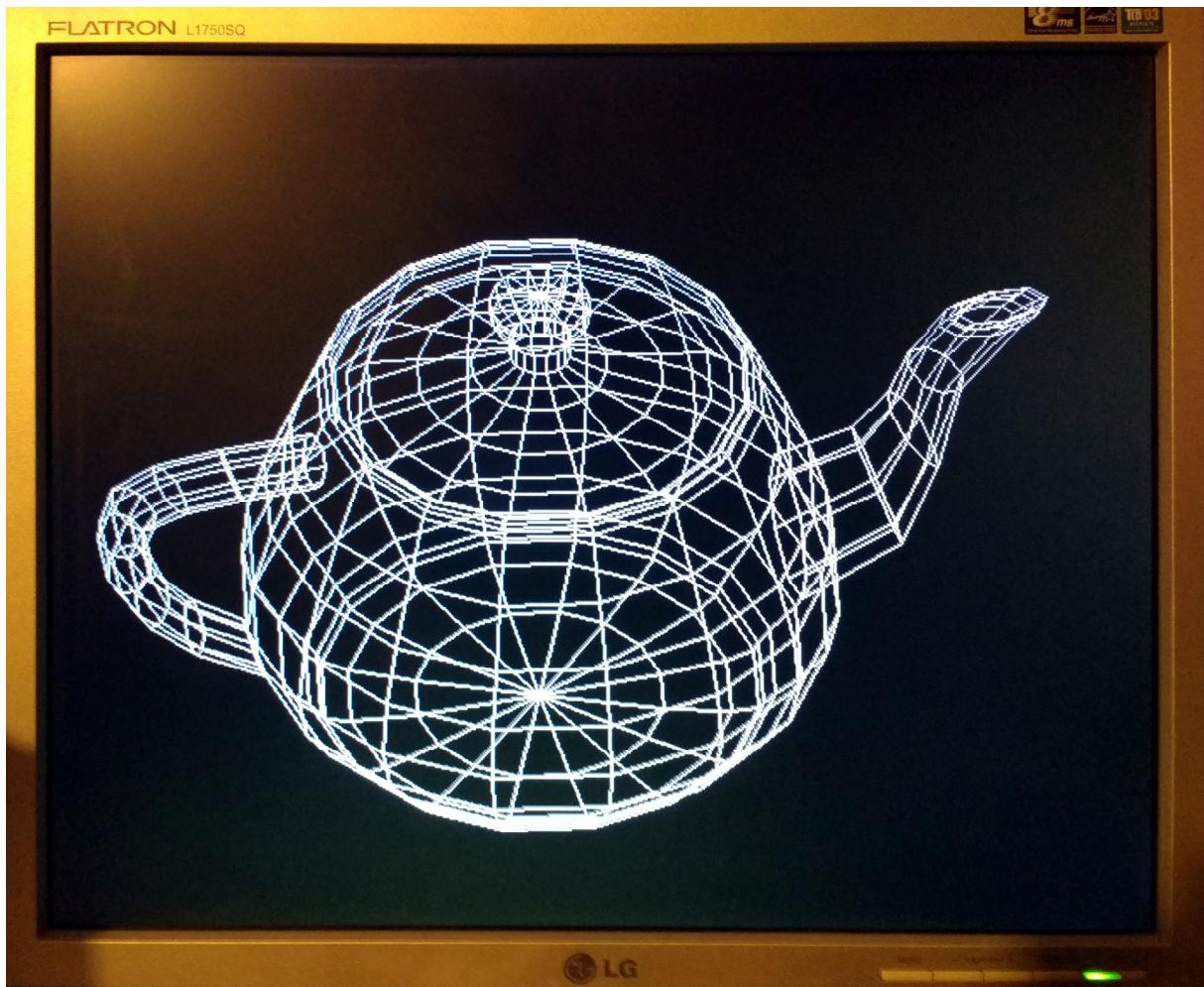
4.2. Kompilacja i uruchomienie projektu na module SNF-0

Projekt jest dostępny w katalogu *HDL/SNF0_3D_processor* w repozytorium projektu inżynierskiego dostępnego pod adresem:

https://github.com/papciakk/FPGA_Board (dostęp 2.11.2016)

Po zimportowaniu projektu do programu Altera Quartus II, należy go skompilować za pomocą opcji *Start compilation* w menu *Processing*. Po udanej komplikacji projekt powinien być gotowy do zaprogramowania układu FPGA, zgodnie z instrukcją podaną w podrozdziale 3.5.

Jeżeli do modułu został podpięty monitor i klawiatura PS2, możliwe będzie przetestowanie działania programu.



Rys. 12. Działanie przykładowego specjalizowanego procesora graficznego

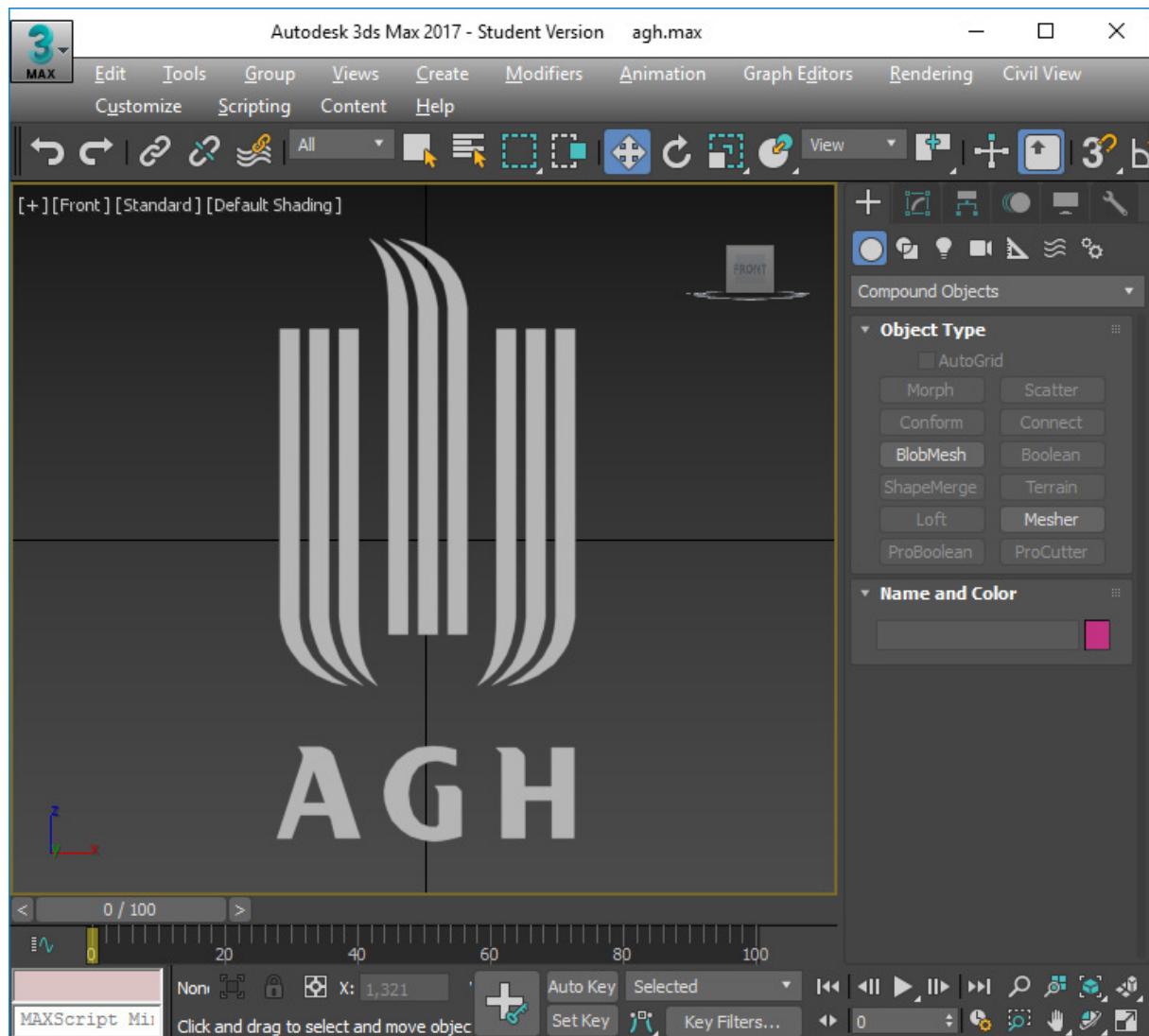
4.3. Sterowanie przy pomocy klawiatury PS2

Za pomocą klawiatury PS2, można sterować obrotem i skalowaniem wczytanej bryły 3D.

- W – zwiększenie kąta obrotu względem osi Y
- S – zmniejszenie kąta obrotu względem osi Y
- A – zwiększenie kąta obrotu względem osi X
- D – zmniejszenie kąta obrotu względem osi X
- Q – zwiększenie kąta obrotu względem osi Z
- E – zmniejszenie kąta obrotu względem osi Z
- Z – zwiększenie skali obiektu
- X – zmniejszenie skali obiektu

5. Instrukcja przygotowania własnych modeli do wyświetlenia przy pomocy procesora graficznego

5.1. Przygotowanie modelu w programie Autodesk 3DS Max

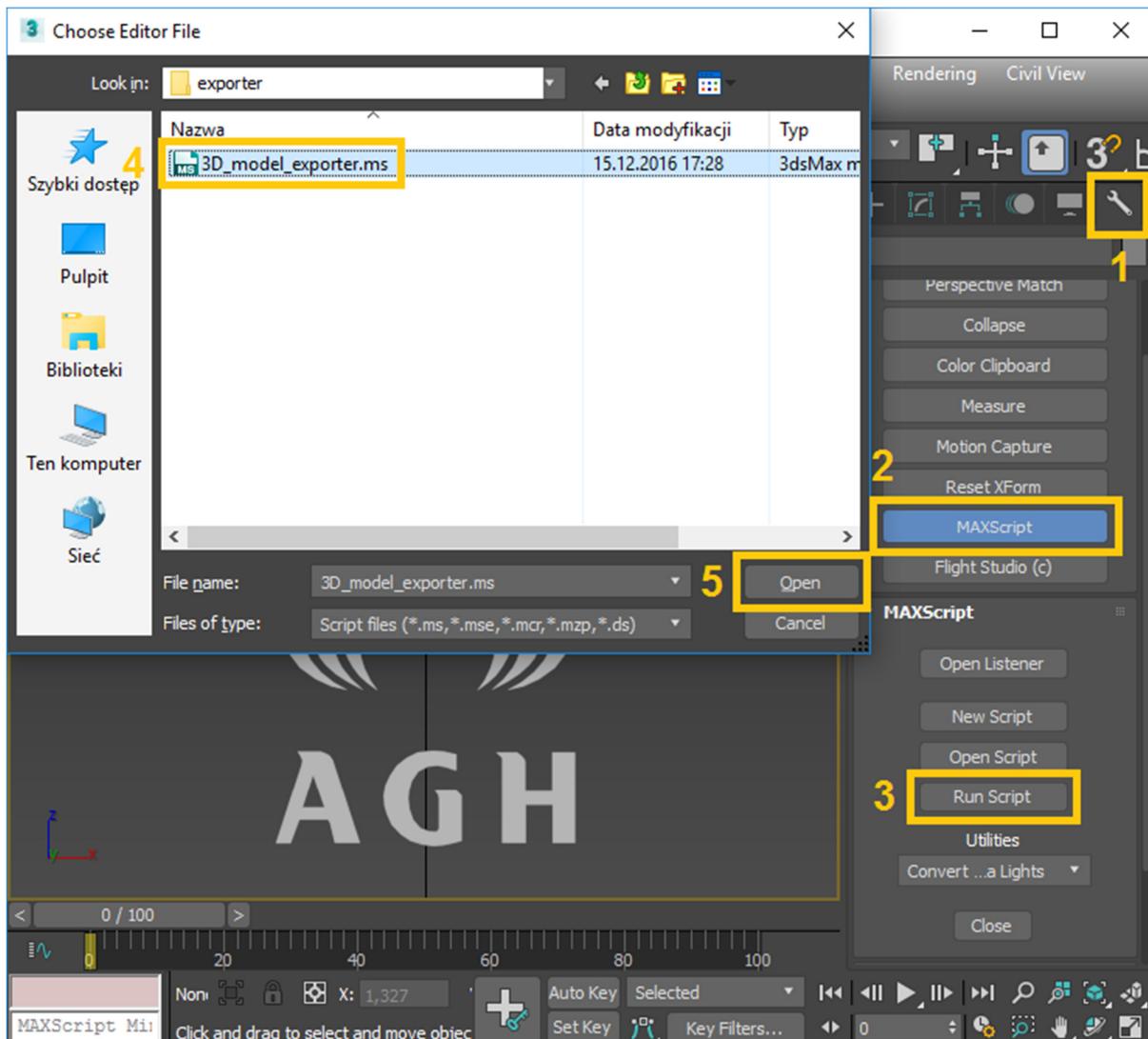


Rys. 13. Scena w programie 3DS Max. Ilość krawędzi jest mniejsza niż 5500.

Do wykonania poniższych czynności potrzebne będzie oprogramowanie Autodesk 3DS Max, w wersji co najmniej 2010.

Pierwszym krokiem jest przygotowanie modelu 3D, który chcemy wyświetlić. W razie potrzeby należy zaimportować model przy pomocy odpowiedniego plug-inu, obsługującego wymagany format. Jeżeli model składa się z więcej niż jednego obiektu, należy je ze sobą połączyć (np. przy narzędziu *collapse* w bocznym menu *utilities*). Ważne jest, aby model posiadał nie więcej niż **5500 krawędzi**. Ograniczenie to jest związane z pojemnością pamięci RAM układu FPGA.

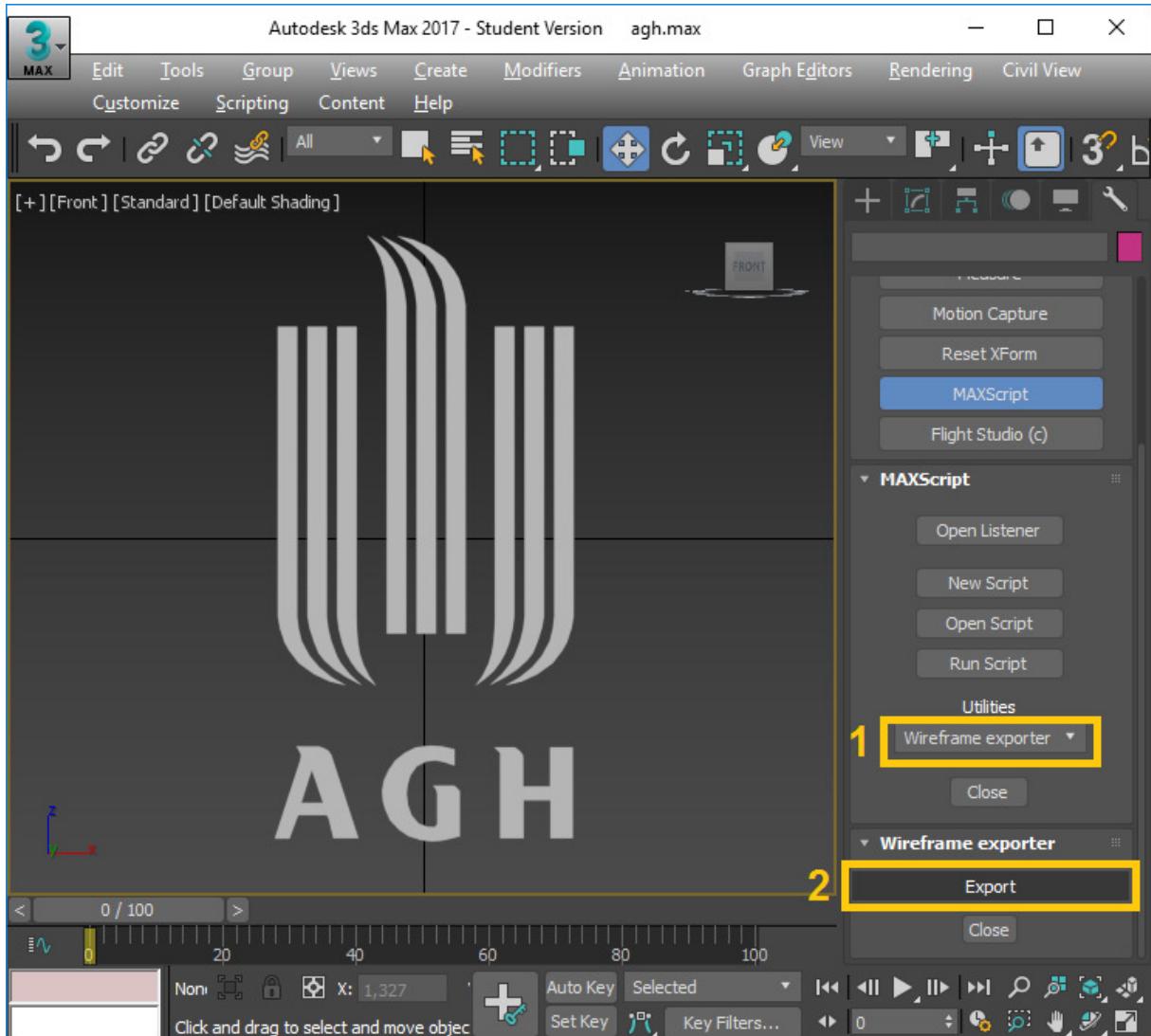
5.2. Eksport modelu przy użyciu skryptu



Rys. 14. Wczytywanie skryptu eksportu modelu w programie 3DS Max

Kiedy scena do wyświetlenia na procesorze graficznym jest gotowa, należy wczytać skrypt eksportu *3D_model_exporter.ms*. Znajduje się on w repozytorium projektu w folderze *software/exporter*. Aby to zrobić należy najpierw otworzyć panel języka skryptowego *MAXScript*. Należy kliknąć w opcję *Utilities* w menu bocznym i wybrać opcję *MAXScript*. Następnie w otworzonym panelu trzeba kliknąć w przycisk *Run script* i wybrać eksporter *3D_model_exporter.ms*. Kolejne kroki całej operacji zostały przedstawione na rysunku 14.

Po właściwym wczytaniu skryptu, należy wybrać go z listy rozwijanej *Utilities* (figuruje w niej jako *Wireframe exporter*). Pojawi się przycisk *Export scene objects*. Po jego naciśnięciu, wyświetli się dialog wyboru ścieżki zapisu pliku z modelem 3D. Zostaną do niego wyeksportowane modele znajdujące się w aktualnej scenie projektu 3DS Max. Proces przedstawiono na rysunku 15.



Rys. 15. Eksport modelu w programie 3DS Max do wykorzystania na procesorze graficznym

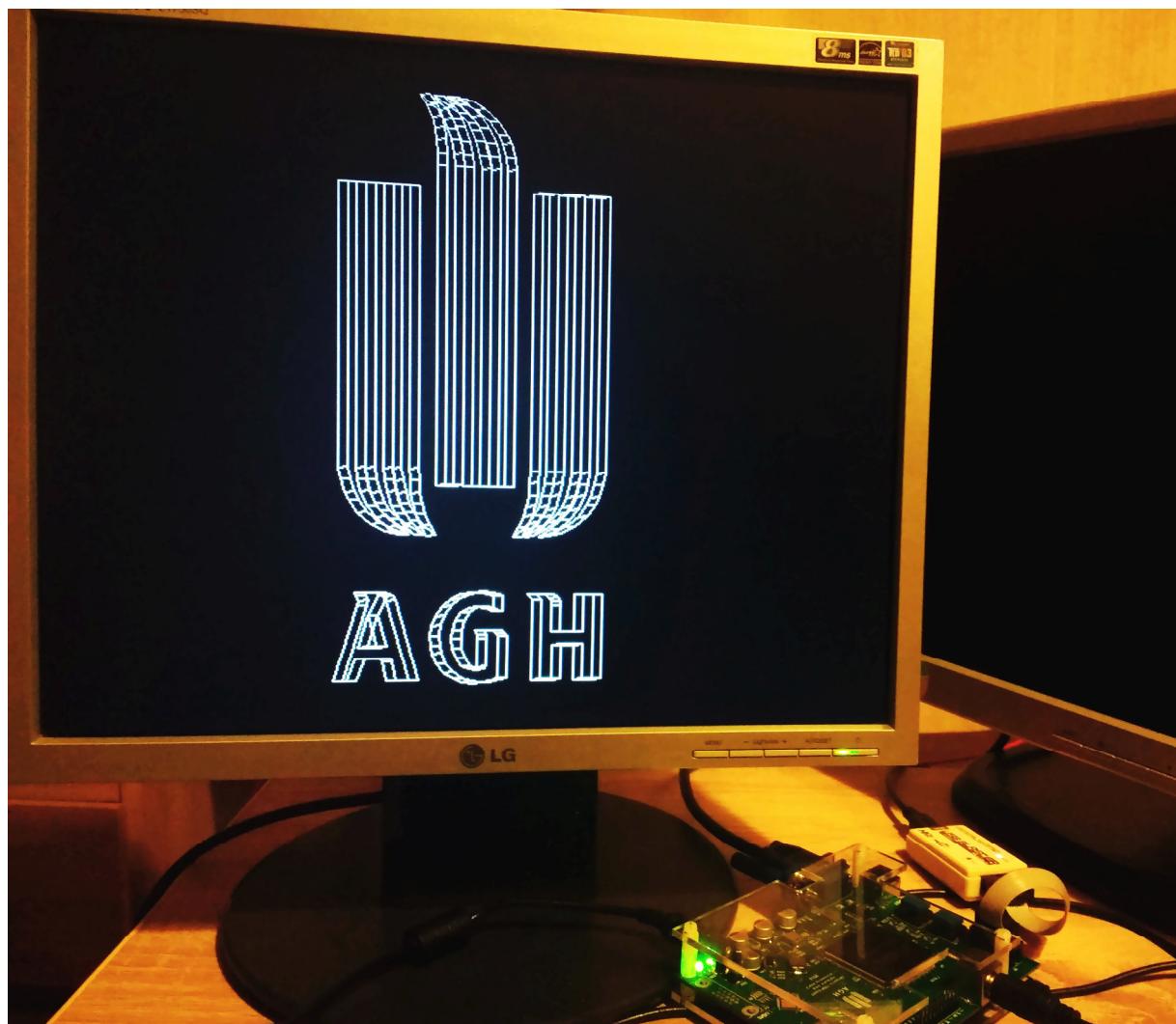
5.3. Przygotowanie do wyświetlenia modelu na module SNF-0

Po wyeksportowaniu modelu z programu 3DS Max przy użyciu dołączonego skryptu, należy skopiować go do folderu projektu procesora graficznego, do katalogu *src* i nazwać *model.vhd*. Poniżej przedstawiono fragment wygenerowanego pliku *vhd*, który zawiera dane modelu 3D zapisane z postaci szesnastkowej.

Następnie należy otworzyć projekt i go skompilować. Po zaprogramowaniu układu FPGA, na monitorze powinna pojawić się wyeksportowana scena. Rysunek 16 przedstawia wynik działania programu z wyeksportowanym modelem 3D.

```
use work.typedefs.all;

package model is
    constant MODEL_EDGES_NUMBER : integer := 656;
    constant MODEL_DATA : mesh_data(655 downto 0) := (
        X"0DBF01F4F7C70D9601F4F67C",
        X"0D9601F4F67C0D96FE0CF67C",
        X"0D96FE0CF67C0DBFF0CF7C7",
        X"0DBFFE0CF7C70DBF01F4F7C7",
        ...
    );
```



Rys. 16. Moduł SNF-0 wyświetlający wyeksportowany model 3D

5. Spis rysunków

Rys. 1. Elementy zestawu uruchomieniowego SNF-0	4
Rys. 2. Porty i układy modułu uruchomieniowego SNF-0	5
Rys. 3. Schemat blokowy modułu.....	7
Rys. 4. Menedżer urządzeń po pierwszym podłączeniu programatora USB-Blaster.....	11
Rys. 5. Wybór lokalizacji sterownika USB-Blaster.....	12
Rys. 6. Wybór układu FPGA podczas tworzenia nowego projektu w programie Altera Quartus II.....	13
Rys. 7. Ustawienie funkcji wyprowadzenia nCEO w opcjach układu FPGA.....	14
Rys. 8. Ustawienie standardu wejścia/wyjścia w konfiguracji układu FPGA	15
Rys. 9. Okno konfiguracji wyprowadzeń układu FPGA.....	16
Rys. 10. Okno programowania układu FPGA.....	22
Rys. 11. Wybór programatora	23
Rys. 12. Działanie przykładowego specjalizowanego procesora graficznego	25
Rys. 13. Scena w programie 3DS Max. Ilość krawędzi jest mniejsza niż 5500.....	26
Rys. 14. Wczytywanie skryptu eksportu modelu w programie 3DS Max.....	27
Rys. 15. Eksport modelu w programie 3DS Max do wykorzystania na procesorze graficznym	28
Rys. 16. Moduł SNF-0 wyświetlający wyeksportowany model 3D	29