

# Pierwsze kroki z FPGA (11)

## Sprzętowy generator funkcji sinus oraz cosinus w oparciu o NCO

Założeniem projektu było zbudowanie cyfrowego generatora kwadraturowych funkcji sinusoidalnych sinus i cosinus o częstotliwości regulowanej w zakresie 1 Hz do 10 kHz za pomocą przycisków i stałej amplitudzie. Ustawiona częstotliwość miała być wyświetlana na wyświetlaczu 7-segmentowym.

Projekt prezentowany w artykule w założeniu powinien służyć jako przestrajany generator funkcyjny:

- Do zmiany częstotliwości wykorzystane są dwa przyciski, których przytrzymanie powoduje coraz szybszą zmianę wartości częstotliwości.
- Aktualna częstotliwość jest pokazywana na wyświetlaczu 7-segmentowym.
- Wyjścia analogowe są kwadraturowe (sin/cos), konwerter cyfrowo analogowy oparty jest o sprzętowe generowanie sygnałów PWM w układzie FPGA, które następnie filtrowane są przez prosty, 1-biegunowy filtr RC.

Projekt został zrealizowany na platformie sprzętowej maXimator firmy KAMAMI razem z nakładką rozszerzającą, zawierającą przyciski oraz wyświetlacz 7-segmentowy. Płyta ta zawiera układ FPGA Altera MAX10 10M08DAF256C8GES, którego parametry istotne do tego projektu to:

Projekt powstał w Katedrze Elektroniki Wydziału Informatyki, Elektroniki i Telekomunikacji AGH, pod kierunkiem dr inż. Pawła Rajdy i dr inż. Jerzego Kasperka.

- Zintegrowana pętla sprzężenia fazowego, pozwalająca na wygenerowanie odpowiednio wysokiej częstotliwości sterującej dla reszty układu.
- Duża częstotliwość drzewa zegarowego (aż do ok. 400 MHz).
- Mała pojemność pinów I/O (poniżej 7 pF), co pozwoliło na uzyskanie dużej częstotliwości PWM i łatwości filtrowania sygnału.
- Bezpłatne oprogramowanie narzędziowe Quartus, które zawiera moduł NCO (*Numerical Controlled Oscillator*). Opis układu przygotowano w języku VHDL, z modułem głównym (*top*) przygotowanym zarówno jako plik VHDL, jak i schemat

blokowy programu Quartus. Kod VHDL przygotowany w edytorze SigasiStudio, a przetestowany symulatorem Model SIM.

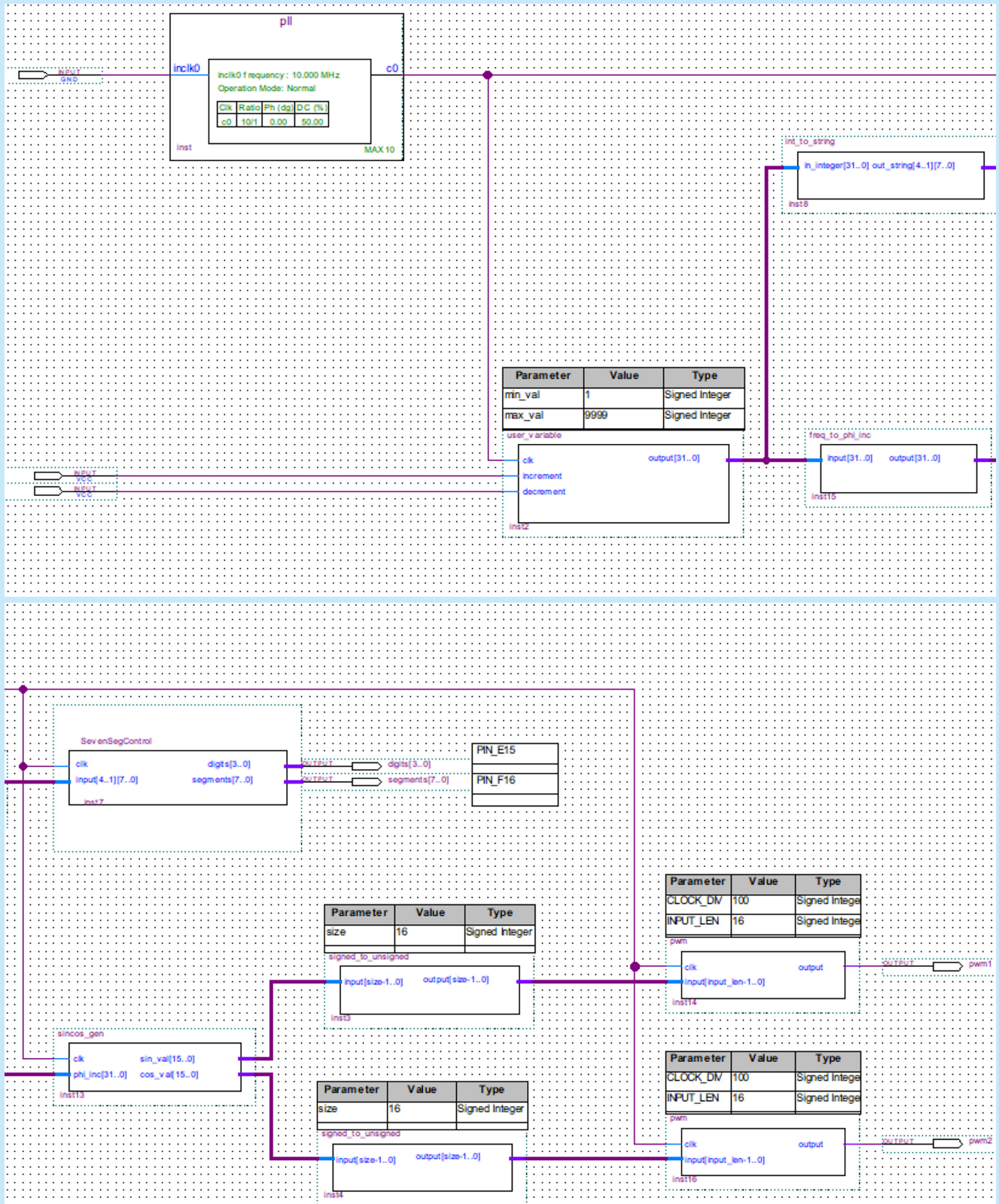
Warte wspomnienia jest ograniczenie modułu NCO. W darmowej wersji programu Quartus do poprawnego funkcjonowania wymaga on stale przyłączonego programatora.

### Schemat blokowy

Niżej omówiono bloki funkcjonalne w kolejności ich występowania na schemacie blokowym pokazanym na rysunku 1.

**PLL** PLL wbudowany w układ FPGA służy do wygenerowania zegara podstawowego o częstotliwości 100 MHz. Następnie, moduły przez generowanie sygnałów *enable* ustalają swoje wewnętrzne taktowanie.

Port	Kierunek	Typ	Znaczenie
inclk0	IN	std_logic	Zegar 10 MHz (z oscylatora)
c0	OUT	std_logic	Zegar 100 MHz



Rysunek 1. Schemat blokowy implementowanego projektu

**USER\_VARIABLE** Moduł służy do przechowywania oraz zmieniania częstotliwości wyjściowej. Odczytuje stan przycisków i modyfikuje zapamiętaną wartość.

Port	Kierunek	Typ	Znaczenie
min_val	generic	integer	Minimalna wartość zmiennej wyjściowej
max_val	generic	integer	Maksymalna wartość zmiennej wyjściowej
Clk	IN	std_logic	Zegar 100 MHz
increment	IN	std_logic	Stan linii przycisku odpowiadającego za zwiększanie częstotliwości
decrement	IN	std_logic	Stan linii przycisku odpowiadającego za zmniejszanie częstotliwości
Output	OUT	integer range (min_val to max_val)	Ustalona wartość częstotliwości

**INT\_TO\_STRING** Moduł przeprowadza operację zmiany wartości liczby na odpowiadający jej string w reprezentacji dziesiętnej. Został odziedziczony z pomocą tablicy LUT.

Port	Kierunek	Typ	Znaczenie
in_integer	IN	integer range (0 to 9999)	Wartość wejściowa
out_string	OUT	string(4 downto 1)	Reprezentacja dziesiętna wejścia

**SEVEN\_SEG\_CONTROL** Moduł jest sterownikiem wyświetlacza 7-segmentowego. Przeprowadza dekodowanie cyfr na kod wyświetlacza oraz multipleksuje wyświetlanie cyfr (z częstotliwością 1 kHz).

Port	Kierunek	Typ	Znaczenie
Clk	IN	std_logic	Zegar wejściowy 100 MHz
Input	IN	string(4 downto 1)	Tablica znaków do wyświetlenia (długość 4)
Digits	OUT	std_logic_vector (3 downto 0)	Wyjścia wspólnych katod do zapalania poszczególnych cyfr wyświetlacza
Segments	OUT	std_logic_vector (7 downto 0)	Anody cyfr wyświetlacza (segmenty od A do G + kropka)

**FREQ\_TO\_PHI\_INC** Zadaniem tego bloku jest zamiana wartości częstotliwości na wartość zmiany fazy przekazywaną do modułu NCO. Moduł przeprowadza operację  $output \leq (input * 214748) / 500$ .

Port	Kierunek	Typ	Znaczenie
Input	IN	natural	Częstotliwość w Hz
Output	OUT	natural	Obliczona wartość zmiany fazy do modułu NCO

**SIN\_COS\_GEN** Plik ten opakowuje IP core NCO w celu jego łatwiejszej obsługi.

Port	Kierunek	Typ	Znaczenie
Clk	IN	std_logic	Zegar wejściowy 100 MHz
phi_inc	IN	natural	Wartość zmiany fazy
sin_val	OUT	signed(15 downto 0)	Wartość wyjścia sinusoidalnego, w zapisie U2 (signed)
cos_val	OUT	signed(15 downto 0)	Wartość wyjścia kosinusoidalnego, w zapisie U2 (signed)

**SIGNED\_TO\_UNSIGNED** Blok ten odpowiada za zamianę liczby w zapisie U2 na zmienną bez znaku (przesunięcie o offset), którego wymaga moduł PWM. Przeprowadza operację  $output \leq val + 2 ** size - 1$ .

Port	Kierunek	Typ	Znaczenie
Size	generic	integer	Liczba bitów zmiennej wejściowej
Input	IN	signed(size - 1 downto 0)	Zmienna ze znakiem
Output	OUT	unsigned(size - 1 downto 0)	Zmienna bez znaku przesunięta o odpowiednią wartość

**PWM** Ostatni blok służy do modulacji fali wyjściowej w takt generowanych wartości z modułu NCO. Można ustawić go według potrzeb (częstotliwość wyjściowa, rozdzielczość, skalowanie wejścia). Aktualne ustawienia: częstotliwość sygnału 1 MHz, rozdzielczość 1%, 16-bitowe wejście.

Wartość zmiennej *input* jest zatrzymywana podczas każdego okresu fali wyjściowej – maksymalne odświeżanie to 1 MHz.

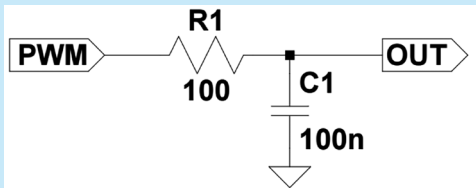
Port	Kierunek	Typ	Znaczenie
CLOCK_DIV	generic	natural	Wartość dzielnika częstotliwości = częstotliwość zegara / częstotliwość wyjściowa
INPUT_LEN	generic	natural	Liczba bitów zmiennej wejściowej
Clk	IN	std_logic	Zegar wejściowy
Input	IN	unsigned(INPUT_LEN - 1 downto 0)	Wartość wejścia do modulatora
Output	OUT	std_logic	Pin wyjściowy PWM

### Filtr dolnoprzepustowy

Aby z układu cyfrowego otrzymać analogowy sygnał wyjściowy należy użyć konwertera C/A. W tym projekcie tę rolę pełni moduł PWM oraz filtr dolnoprzepustowy RC (**rysunek 2**). Wartości dobrane do projektu to  $R=100 \Omega$  i  $C=100 nF$ , co daje częstotliwość ok. 16 kHz przy 3 dB spadku. Charakterystykę przejściową filtra pokazano na **rysunku 3**. Układ bardzo silnie tłumi częstotliwości pracy PWM (40...70 dB), przepuszczając pożądane częstotliwości (do 10 kHz). Układ dodatkowy wpina się w złącze przeznaczone dla konwertera USB/UART. Jest to miejsce na płytce, gdzie obok pinów jest najlepiej dostępna masa (poza zajętej przez górną płytkę rozszerzeń). Widok zmontowanego filtra pokazano na **fotografii 4**.

### Pomiary

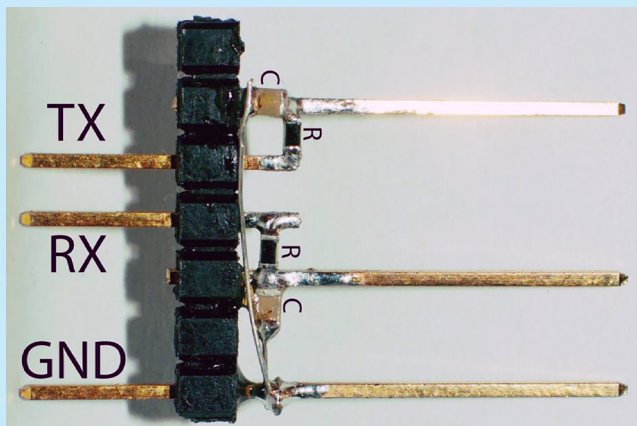
Po zmontowaniu układ został przetestowany oscyloskopem (Siglent SDS 1022). Otrzymane przebiegi wyjściowe dla różnych częstotliwości sygnałów pokazano na **rysunkach 5...8**. Na **rysunku 9**



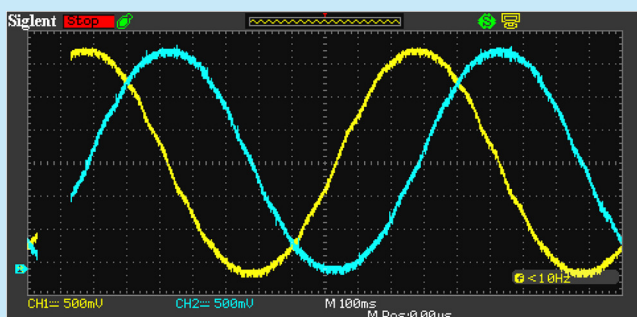
Rysunek 2. Schemat FDP użytego w projekcie



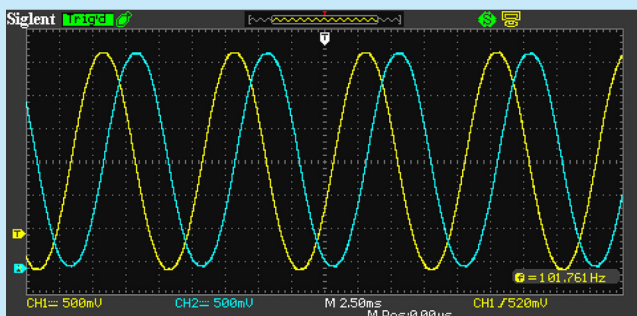
Rysunek 3. Charakterystyka użytego FDP



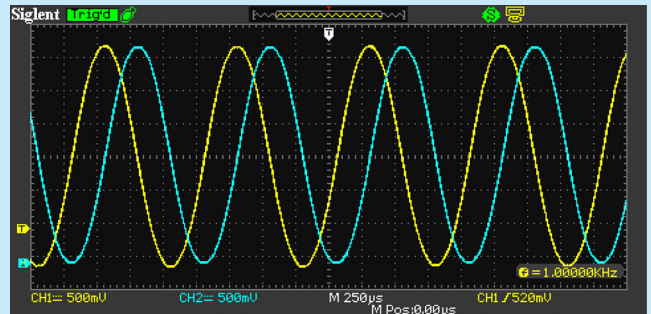
Fotografia 4. Wygląd zmontowanego FDP



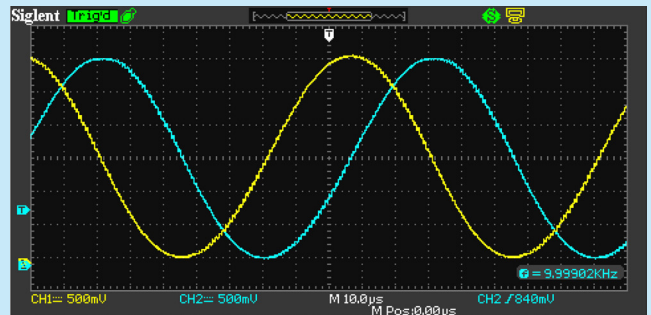
Rysunek 5. Przebieg wyjściowy o częstotliwości 1 Hz



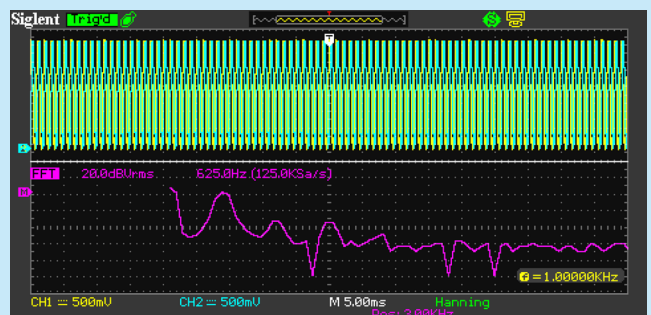
Rysunek 6. Przebieg wyjściowy o częstotliwości 100 Hz



Rysunek 7. Przebieg wyjściowy o częstotliwości 1 kHz



Rysunek 8. Przebieg wyjściowy o częstotliwości 10 kHz



Rysunek 9. FFT sygnału wyjściowego

pokazano FFT sygnału, z której wynika, że druga i trzecia harmoniczna są na poziomie -30 dB, reszta nie wykrywalna (poniżej poziomu szumów).

### Podsumowanie

Projekt został zaimplementowany oraz przetestowany. Spełnił wszystkie wymagania projektowe. W domowym laboratorium taki układ może służyć za prosty generator do testowania innych układów, jak np. przetworniki ADC.

Za pomocą tego projektu została przetestowana funkcjonalność IP core NCO, jak również wyświetlacz 7-segmentowy i przyciski. Problemem implementacyjnym była niewielka liczba pinów wprowadzonych na złącza w zestawie maXimator, w szczególności przy używaniu wyświetlacza 7-segmentowego.

Warta zauważenia jest względna dokładność zegarów na płycie oraz w oscyloskopie – błąd częstotliwości nie przekroczył 1%.

Obecny projekt podczas syntezy zajmuje około 60% zasobów układu FPGA. Dużą część zajmują układy NCO, które zawierają duże LUT-y do szybkiego obliczania wartości.

Grzegorz Gajoch