

**Wydział Fizyki, Astronomii i Informatyki Stosowanej**  
**Uniwersytetu Mikołaja Kopernika**  
**Pracownia układów programowalnych**

**Ćwiczenie 7**

**Implementacja wbudowanych bloków funkcjonalnych DCM i PLL w strukturze programowalnej SPARTAN-6 (ATLYS, CMT, DCM, PLL, DLL, DFS, DDS, FPGA Editor)**

### Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z zasobami logicznymi jakie oferuje układ programowalny typu FPGA Spartan-6 (XC6SLX45-CSG324C) a w szczególności poznanie wbudowanych bloków funkcjonalnych CMT (Clock Management Tiles) i oferowanych przez nie bloków DCM (Digital Clock Managers) oraz pętli PLL (Phase Locked Loop), stosowanych w procesie przetwarzania sygnałów zegarowych. Dodatkowo, wykonując ćwiczenie możliwe będzie zaznajomienie się z możliwościami edycji struktury FPGA przy użyciu narzędzia „FPGA Editor”. W ćwiczeniu należy zbudować projekt dzielnika częstotliwości wykorzystując opisane bloki funkcjonalne i zasoby logiczne układu. Kolejne stopnie podziału częstotliwości należy zaobserwować na diodach LED dostępnych w zestawie Atlys™ Spartan-6 FPGA Development Board.

### Zagadnienia do przygotowania

Wbudowane bloki funkcjonalne CMT, DCM oraz PLL – budowa, zasada działania, możliwe konfiguracje, zastosowania, warunki właściwego wykorzystania [1].

Synteza częstotliwości, pętla DLL i PLL, Bezpośrednia synteza cyfrowa (DDS).

Zasoby układów programowalnych FPGA rodziny Spartan-6 [2].

Projekt dzielnika częstotliwości o zadanym stopniu podziału (VHDL).

Zapoznanie się z dokumentacją zestawu Atlys™ Spartan-6 FPGA Development Board [3].

### Przebieg ćwiczenia

1. Uruchomić środowisko ISE, wybrać odpowiedni układ oraz dodać do projektu nowe źródło „VHDL Module”. W celu zmapowania bloku DCM można skorzystać z „Language Template”. W ścieżce: „VHDL → Device Primitive Instantiation → FPGA → Clock Components → Spartan Digital Clock Manager” znajduje się szablon wykorzystania elementu „DCM\_SP”<sup>1</sup>. Do wyboru mamy trzy zasadnicze funkcje bloków DCM: funkcję cyfrowego syntezerza częstotliwości, cyfrowego przesuwnika fazy i układu DLL.
2. Wykorzystując kaskadowo połączony układ DLL (bloku funkcjonalnego DCM), z dzielnikiem częstotliwości o zadanym stopniu podziału, obniżyć częstotliwość sygnału zegarowego w taki sposób aby otrzymać na wyjściu sygnał o częstotliwości 1Hz. Opisać projekt w VHDL'u. Jako sygnał wejściowy wykorzystać generator dostępny w zestawie o częstotliwości 100 MHz. Odpowiednio zmapować blok „DCM” wraz z sygnałem sprzężenia zwrotnego oraz odpowiednimi buforami (zgodnie z dokumentacją). Wymagane komponenty można znaleźć w „Language Template”. Stopień podziału częstotliwości bloku DLL ustawić na 10. W opisie należy dodać bibliotekę UNISIM.
3. Wyjście „CLKDV” doprowadzić do wejścia dzielnika częstotliwości opisanego funkcjonalnie w VHDL'u, który powinien posiadać stopień podziału ustawiony tak, aby

---

1 Poszczególne etapy właściwego wyboru szablonu bądź modułów IP mogą ulegać nieznacznym zmianom w zależności od wersji używanego środowiska projektowego.

częstotliwość wyjściowa wynosiła około 1 Hz. Sygnał wyjściowy doprowadzić do wybranej z dostępnych w zestawie diod LED. Na podstawie dołączonego do ćwiczenia pliku **AtlysGeneral.ucf**, utworzyć i nadać odpowiednie numery wyprowadzeń w pliku \*.ucf [4]. Przeprowadzić syntezę, implementację i przesłać mapę bitową (plik \*.bit) do zestawu Atlys. Zaobserwować stan diod LED.

4. Dokonać edycji struktury przy użyciu „FPGA Editor'a”, wybierając w oknie procesów „*Implement Design → Place & Route → View/Edit Routed Design (FPGA Editor)*”. Zmienić stopień podziału częstotliwości przez blok DCM. Zaobserwować sposób implementacji pozostałych elementów. Ponownie zmierzyć częstotliwość migania diody LED.
5. Wyprowadzić wyjście „LOCKED” bloku DCM na dostępną diodę LED oraz zbudować układ bramkujący na sygnale zegarowym w pętli sprzężenia zwrotnego, który umożliwi zewnętrznym przyciskiem zablokowanie sygnału zegarowego podawanego na wejście „CLKFB”. Zaobserwować stan logiczny na wyjściu „LOCKED” w zależności od obecności sygnału zegarowego na wejściu „CLKFB” (stanu na przycisku blokującym).
6. Zmodyfikować projekt w taki sposób, aby otrzymać częstotliwość załączania diody równą 2Hz i 0.5Hz (Metodyka tworzenia projektu: dowolna bądź ustalona z opiekunem ćwiczenia np. można w tym przypadku użyć dwóch modułów DCM połączonych kaskadowo i skorzystać z funkcji syntezy częstotliwości). Wybór odpowiedniej częstotliwości powinien być dokonany przy użyciu dowolnie wybranych przełączników. Należy przy tym pamiętać o zastosowaniu układu eliminującego drgania mechaniczne przełączników.
7. (punkt obowiązkowy dla 60–cio godzinnych grup ćwiczeniowych) W analogiczny sposób jak zostało to przedstawione w punkcie pierwszym, zrealizować samodzielnie projekt pętli PLL. W tym celu należy dodać do projektu nowe źródło „VHDL Module”. Aby zmapować blok PLL można skorzystać z „*Language Template*”. Znajduje się on w następującej ścieżce: „*VHDL → Device Primitive Instantion → FPGA → Clock Components → Base Phase Locked Loop*” znajduje się szablon wykorzystania elementu „*PLL\_BASE*”<sup>2</sup>.

## Literatura

[1] Spartan-6 FPGA Clocking Resources: **ug382.pdf**.

[2] Spartan-6 Family Overview: **ds160.pdf**.

[3] Atlys™ Spartan-6 FPGA Development Board: **Atlys\_rm\_v2.pdf, Atlys\_c2\_sch.pdf**.

[4] Plik: **AtlysGeneral.ucf**.

**UWAGA**, pozycje literaturowe dostępne są w postaci elektronicznej.

---

<sup>2</sup> Poszczególne etapy właściwego wyboru szablonu bądź modułów IP mogą ulegać nieznacznym zmianom w zależności od wersji używanego środowiska projektowego.