

LAB#1 Generator sygnałów testujących i prosty układ kombinacyjny.

1. Utwórz entity, które modeluje generator sygnałów: S1, S2, S3 CLK. Sygnały te zadane są poniżej za pomocą różnych notacji.

Wymagane parametry sygnałów:

- Sygnał zegarowy CLK: przebieg zaczyna się od stanu '0' i ma okres $T=40$ ns;

W przypadku sygnału CLK raz zasymuluj układ z sygnałem CLK jako BIT, po czym jako zmień typ sygnału CLK na STD_LOGIC i powtórz symulację.

O czym należy pamiętać w przypadku użycia typu STD_LOGIC?

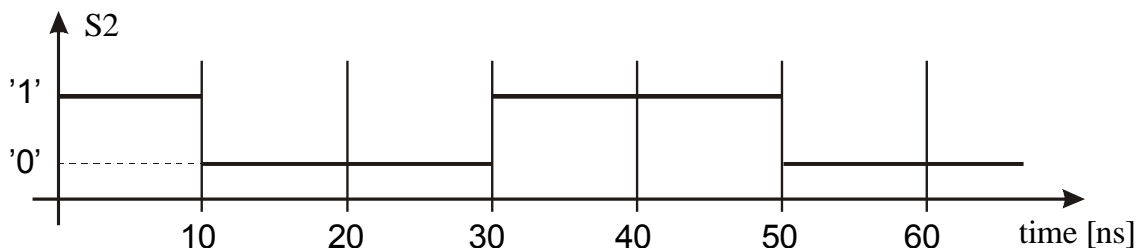
W czasie symulacji może pojawić się problem wynikający z określenia CLK jako wyjścia (OUT). Sygnał wyjściowy (OUT) nie może pojawić się po prawej stronie przypisania np. $CLK \leq \text{not } CLK$.

Możliwe rozwiązania to:

- zadeklarowanie portu CLK jako "inout"
- wprowadzenie dodatkowego sygnału wewnętrznego np.

```
architecture AAA of Generator is
    CLK1 : std_logic;
begin
    .....
    CLK_1 <= .....? .....
    CLK <= .....? .....
    .....
end architecture AAA;
```

- Sygnał S1 mający przebieg '1' @ 0 ns, '0' @ 10 ns, '1' @ 20 ns i dalej
- Sygnał S2 mający przebieg zadany graficznie:



- Sygnał S3 jest sygnałem dwubitowym (STD_LOGIC_VECTOR) i ma przebieg: "01" @ 0 ns, "00" @ 10 ns, "11" @ 20 ns.
- Sygnał S4 (typu INTEGER) ma przebieg: 2 @ 10 ns, 5 @ 40 ns

2. Wygeneruj te same przebiegi co w pkt 1 używając komend “force”. Umieść je w pliku komend (*.do). W tym celu należy utworzyć dodatkową “architecture” w tym samym entity. Nowo utworzona architecture może być pusta (begin end). Pozwoli to uruchomić symulację bez uruchamiania architektury utworzonej w pkt 1.
3. Zaprojektuj układ kombinacyjny porównujący dwie liczby 2-bitowe (A i B). Układ powinien mieć dwa wyjścia: “EQ” and “GE”. Wyjście “EQ” jest w stanie wysokim ‘1’ gdy $A=B$, a “GE” jest w stanie wysokim gdy $A \geq B$. Użyj tabel Karnough do minimalizacji tych funkcji. Naszkicuj schemat obu funkcji po minimalizacji (z użyciem podstawowych bramek logicznych).

Utwórz entity z 2 architecture:

- “Behavioral” – zawiera opis behawioralny w postaci 2 równań logicznych bez minimalizacji (boolean equations)
- „Structural” – nadal opis za pomocą funkcji logicznych, ale wprowadzając sygnały wewnętrzne i funkcje logiczne zgodnie z uzyskanym schematem (tym razem nie używamy jeszcze komponentów). Sygnały we, wy i wewnętrzne oznacz tak jak na schemacie.

Zasymuluj obie architektury i sprawdź poprawność modeli. Użyj odpowiednio komendy force aby przetestować wszystkie 16 kombinacji. Sygnały testujące to sygnały okresowe o okresach T, 2T, 4T, 8T odpowiednio dla każdego wejścia. Przyjmij $T=50$ ns.

Przygotuj osobny generator sygnałów testujących wykorzystując doświadczenie z pkt 1. Połączenie generatora z układem testowanym przedstawione zostanie na laboratorium i/lub wykładzie.

Zaliczenie laboratorium na podstawie przedstawionego schematu z pkt 2 wraz z tabelami Karnough (odręczenie na kartce) oraz wyników symulacji na komputerze (zaprezentować prowadzącemu).