

# Projekt indywidualny 1 - wariant 12

Projekt jednostki wykonawczej **exe\_unit\_w12**  
operującej na liczbach w kodzie **U2**

Semestr zima 2023/24

## Cel projektu

Celem projektu jest zaimplementowanie jednostki wykonawczej **exe\_unit\_w12** realizującej operacje arytmetyczne, logiczne i inne na liczbach całkowitych zapisanych w kodzie **U2**.

W skład realizacji projektu wchodzi:

- implementacja modelu **exe\_unit\_w12** w języku *SystemVerilog*
- synteza logiczna **exe\_unit\_w12** przy użyciu programu *YOSYS*
- realizacja modułu **testbench** i weryfikacja poprawności działania **exe\_unit\_w12** na drodze symulacji logicznej w *ICARUS VERILOG* zarówno modelu (przed synteza logiczną) jak i modułu uzyskanego po syntezie logicznej; wyniki symulacji układów powinny być identyczne
- napisanie specyfikacji zrealizowanego układu wraz z raportem zawierającym wyniki realizacji projektu

## Realizowane operacje

Jednostka **exe\_unit\_w12** ma realizować następujące operacje na danych wejściowych (operacje i punktacja za poprawną implementację, syntezę i weryfikację na drodze symulacji logicznej):

- dodawanie argumentu A i B (3 pkt)
- porównanie argumentów  $A \geq B$ ; jeśli warunek jest spełniony to wynikiem jest liczba większa od 0, w przeciwnym wypadku wynikiem jest 0 (3 pkt)
- ustawienie bitu w argumencie A na wartość 1; numer bitu jest określony w argumencie B; zgłoszenie błędu jeśli wartość w B jest ujemna lub przekracza liczbę bitów w A (3 pkt)

- konwersja liczby z kodu **U2** na kod **ZNAK-MODUŁ**; jeśli konwersja nie może zostać poprawnie wykonana to ma zostać zgłoszony błąd a wynik konwersji jest nieokreślony (3 pkt)

## Wejścia i wyjścia

Rodzaj wykonywanej operacji jest określony przez **n-bitowe** wejście sterujące **i\_oper**. Dane **m-bitowe**, na których wykonywane są operacje są pobierane z dwóch wejść **i\_argA** i **i\_argB**. Wynik jest podawany na **m-bitowe** wyjście **o\_result**.

- **i\_oper** - **n-bitowe** wejście sterujące (kombinacyjne) określające rodzaj operacji do wykonania na argumentach A i B
- **i\_argA** - **m-bitowe** wejście argumentu A
- **i\_argB** - **m-bitowe** wejście argumentu B
- **i\_clk** - wejście zegara (aktywne zbocze narastające)
- **i\_rsn** - wejście resetu synchronicznego (wartość logiczna 0 ustawia wszystkie rejestry modułu **exe\_unit\_w12** na wartość 0)
- **o\_result** - wyjście synchroniczne (z rejestru) wyniku operacji, zmiana wyjścia następuje na zboczu narastającym zegara **i\_clk**

Jednostka **exe\_unit\_w12** ma dodatkowe synchroniczne (z rejestru) wyjście statusu **o\_status** (zmiana wyjścia następuje na zboczu narastającym zegara **i\_clk**), którego kolejno bity określają znaczniki:

- **ERROR** - operacja nie została wykonana; wartość **o\_result** jest nieokreślona (3 pkt)
- **ZERO** - wszystkie bity wyniku są ustawione na 0 (3 pkt)
- **NEG** - wynik jest liczbą ujemną (3 pkt)
- **EVEN** - w wyniku jest parzysta liczba jedynek (3 pkt)

## Wymagania i punktacja

- Synteza logiczna całego modułu **exe\_unit\_w12** wraz ze wszystkimi plikami musi przebiegać poprawnie i nie może zawierać zatrzasków.
- Działanie układu (wyniki symulacji logicznej) modelu układu przed i po syntezie musi być identyczne
- Wszystkie pliki projektu i dokumenty muszą być umieszczone w repozytorium [gitlab-stud.elka.pw.edu.pl](https://gitlab-stud.elka.pw.edu.pl) w odpowiednich katalogach - warunek konieczny

- W repozytorium musi być widoczny systematyczny rozwój i praca nad projektem w postaci historii poszczególnych operacji commit na projekcie - warunek konieczny
- Specyfikacji zrealizowanego układu wraz z raportem zawierającym wyniki realizacji projektu zawierającą:
  - listę wejść i wyjść, parametrów jednostki oraz zakresy ich wartości
  - listę realizowanych funkcji wraz z ich argumentami oraz kodem jaki należy podać na wejście **i\_oper** aby wykonać zadaną funkcję
  - schemat blokowy struktury jednostki
  - specyfikacje sygnałów określających zaimplementowane flagi wraz z przykładowymi sytuacjami kiedy dana flaga przyjmie wartość 0 i 1
  - przykładowe użycie modułu pokazującego jak go instancjonować
  - listę wszystkich plików .sv wraz z informacją jakie moduły są w nich zdefiniowane
  - raport z syntezy logicznej zawierający liczbę przerzutników, zatrzasków i bramek logicznych użytych do realizacji modelu po syntezie logicznej; w raporcie należy zawrzeć tabelę z wymienionymi elementami i ich liczbą
  - raport z przeprowadzonej symulacji układów zawierający przykładowe przebiegi sygnałów dla wszystkich zrealizowanych operacji

Liczba punktów do zdobycia - 30 pkt w tym:

- 12 pkt - za implementację w języku *System Verilog* (4 pkt), synteze (4 pkt) i weryfikację (4 pkt) wszystkich operacji
- 12 pkt - za implementację w języku *System Verilog* (4 pkt), synteze (4 pkt) i weryfikację (4 pkt) wszystkich znaczników
- 6 pkt - specyfikacja (2 pkt) zrealizowanego modułu **exe\_unit\_w12** wraz z raportem z wynikami syntezy (2 pkt) i symulacji (2 pkt)