# Specyfikacja spi\_exe\_unit\_3

Imię: Szymon

Nazwisko: Klimaszewski

Numer albumu: 310717

# Opis

## Sygnały wejściowe

i\_rst

resetujący modułu spi\_exe\_unit.

i\_sclk

zegar modułu spi\_exe\_unit, pochodzący z mastera.

i\_mosi

dane od mastera.

i\_cs

steruje komunikacją z slave z masterem.

Sygnały wyjściowe

o\_miso

dane wyjściowe, pochodzące ze slave.

### Sygnały pomocnicze

- **s\_transfer** pozwalający na pracę shifter
- s\_bit sygnał przechowujący pojedynczy bit, który jest podawany na wyjście slave
- **s\_bit\_next** sygnał przechowujący kolejną wartość s\_bit
- **reg\_argA** rejestr sygnału i\_argA modułu exe\_unit , przechowujący wartość zmiennej, na której moduł exe\_unit będzie wykonywał operacje.
- **reg\_argB** rejestr sygnału i\_argB modułu exe\_unit, przechowujący wartość zmiennej, na której moduł exe\_unit będzie wykonywał operacje.
- **reg\_oper** rejestr sygnału i\_oper modułu exe\_unit, przechowujący numer operacji, którą ten moduł ma wykonać
- **reg\_results** rejestr sygnału o\_results modułu exe\_unit, przechowujący wynik operacji wykonanej przez ten moduł
- **reg\_flags** rejestr flag modułu exe\_unit, w którym przechowywane są wartości poszczególnych flag zwracanych przez ten moduł
- **s\_argA\_next** sygnał przechowujący kolejną wartość sygnału s\_argA
- **s\_argB\_next** sygnał przechowujący kolejną wartość sygnału s\_argB
- **s\_oper\_next** sygnał przechowujący kolejną wartość sygnału s\_oper
- **s\_results\_next** sygnał przechowujący kolejną wartość sygnału s\_results
- **s\_flags\_next** sygnał przechowujący kolejną wartość sygnału s\_flags
- s\_argA sygnał przechowująca wartość, która ma zostać zapisana do rejestru reg argA
- s\_argB sygnał przechowująca wartość, która ma zostać zapisana do rejestru reg argB
- **s\_oper** sygnał przechowująca wartość, która ma zostać zapisana do rejestru reg oper
- s\_results przechowujący wartość, która ma zostać zapisana do rejestru reg\_results
- **s\_flags** sygnał przechowująca wartość, która ma zostać zapisana do rejestru reg\_flags
- **s\_wyniki** wyjście równoległe shiftera
- **s\_watchdog\_we** zezwolenie na wpis
- **s\_wrt** sygnał na wpisanie równolegle/szeregowe shift'era
- **s\_data** do zapisywania na niego po kolei rejestrów wyników i argumentów
- **s\_state** sygnał przechowujący wartość mówiącą o aktualnym stanie slave
- **s\_state\_next** sygnał przechowujący wartość kolejnego stanu slave.

## Parametry

NUM

pomocniczy, liczba bitów na których pracuje moduł watchdog.

BITS

liczba bitów, na których pracuje slave.

### Stany

1

slave jest gotowy do otrzymania danych z mastera. W module watchdog wpisywana jest odpowiednia ilość cykli pobudzenia.

2

dane z zmiennej s\_wyniki są przypisywane do zmiennych pomocniczych i następnie do rejestrów.

3

wartości sygnałów wyjściowych exe\_unit przypisywane są do zmiennych pomocniczych, aby mogły potem trafić do rejestrów.

4

wartości z rejestrów są zapisywane równolegle na głównym rejestrze slave.

### Instancjonowane moduły

#### shifter

odpowiedzialny za przesuwanie bitów otrzymanych z master'a. Slave otrzymuje kolejno pojedyncze bity od master'a i dzięki temu modułowi są one pojedynczo wsuwane i przypisywane do sygnału pomocniczego s\_wyniki. Odgrywa rolę rejestru naszego slave'a

#### watchdog

pobudza slave do pracy na 20-bitowym ciągu. Watchdog ma 20 cykli, dzięki którym możliwe jest zapisanie danych do odpowiednich rejestrów i wykonanie na nich odpowiednich operacji.

#### exe\_unit

wykonuje operacje na danych otrzymanych z master'a. Jego sygnały wejściowe to i\_argA, i\_argB, i\_oper natomiast sygnały wyjściowe to o\_result, o\_flags. Szerszy opis modułu znajduje się w dokumentacji projektu 1.

## Algorytm pracy spi\_exe\_unit\_3

w pierwszym stanie pracy zapisywana jest odpowiednia ilość cykli do modułu watchdog sterującym tą jednostką. Dodatkowo ustawiane są odpowiednie wartości sygnałów zezwalające do transfer danych oraz zmianę stanu do stanu numer 2.

w kolejnym stanie pracy slave'a blokowany jest zapis nowej ilości cykli do modułu watchdog.

Dodatkowo do zmiennych pomocniczych s\_argA\_next, s\_argB\_next, s\_oper\_next, s\_results\_next, s\_flags\_next, przypisywane są dane otrzymane z master'a. Stan ten jest powtarzany do momenty, gdy watchdog zgłosi koniec pracy poprzez sygnał s\_inter.

w kolejnym stanie pracy slave'a, wyniki z exe\_unit przypisywane są do odpowiednich zmiennych pomocniczych s\_flags\_next, s\_results.

ostatnim stanem pracy slave'a jest stan, w którym otrzymane wartości wyników oraz danych wejściowych są przypisywane do wyjścia równoległego i cała informacja jest zwracana do jednostki master.