Specyfikacja spi\_exe\_unit\_2

Imię: Ivan

Nazwisko: Lisovyi

Numer albumu: 317361

# Opis sygnałów

## wejścia

**i\_rst**

Sygnał resetu modułu spi\_exe\_unit

**i\_sclk**

Sygnał zegara modułu spi\_exe\_unit, generowany z mastera.

**i\_mosi**

Wejście slave

**i\_cs**

Sygnał odpowiadjący czy slave powinien komunikować się z master’em

## wyjścia

**o\_miso**

Wyjście slave

## Parametry

**NUM**

Parametr pomocniczy, który służy do określenia liczby bitów, na których pracuje moduł watchdog

**BITS**

Parametr, służący do określenia liczby bitów danych, na których ma pracować slave

## Automat stanów

**1**

Stan oznaczający gotowość slave do otrzymania danych od mastera. Dodatkowo w tym stanie do modułu watchdog wpisywana jest odpowiednia ilość cykli pobudzenia

**2**

Stan, w którym dane z zmiennej s\_wyniki są przypisywane do odpowiednich zmiennych pomocniczych, które następnie przypisywane są do rejestrów.

**3**

Stan, w którym przypisywane są wartości wyjść modułu exe\_unit do odpowiednich zmiennych pomocniczych w celu późniejszego zapisania ich do rejestrów.

**4**

Stan, w którym wartości zapisane w rejestrach są zapisane równolegle na głównym rejestrze slave.

## Opis sygnałów pomocniczych

s\_transfer – sygnał 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 – sygnał przechowująca 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 równelegle shiftera

s\_watchdog\_we – zezwolenie na wpis

s\_wrt – sygnał na wpisanie równolegle/szeregowe shift’era

s\_data – sygnał pomocniczy 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.

## Instancjonowane moduły

### Shifter

Moduł jest odpowiedzialny za przesuwanie bitów otrzymanych z master’a. Slave otzymuje kolejno pojedyncze bity od master’a i dzięki temu modułowi są one pojedynczo wsuwane i przypisywane do sygnału pomocniczego s\_wyniki. W tym modulę znajduję się tak naprawdę rejestr naszego slave’a

### watchdog

Moduł, które zadaniem jest pobudzenie slave’a 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

Moduł, którego zdaniem jest wykonanie operacji na danych otrzymanych z master’a. Przyjmuje on odpowiednio argument i\_argA, i\_argB, i\_oper oraz zwraca wyniki w postaci o\_result, o\_flags. Jego szczegółowa specyfikacja jest opisana w dokumentacji projektu 1.

# Algorytm pracy spi\_exe\_unit\_2

W pierwszym stanie pracy spi\_exe\_unit\_2 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, przypisywane są do odpowiednich zmiennych pomocniczych s\_flags\_next, s\_results, wyniki obliczeń zwrócone przez moduł exe\_unit.

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.