

Specyfikacja spi_exe_unit_1

Imię: Konrad

Nazwisko: Krupski

Numer albumu: 310729

Opis sygnałów

wejścia

i_rst

Sygnał resetu modułu spi_exe_unit

i_sclk

Sygnał zegara modułu spi_exe_unit, który zapewnia odpowiednie sterowania slave

i_mosi

MASTER OUT SLAVE IN. Jednobitowy, sygnał z danymi pochodzącymi od mastera

i_cs

CHIP SELECT.

wyjścia

o_miso

Jednobitowy sygnał, którym przesyłane są dane wyjściowe z slave

Parametry

NUM

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

N

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

BITS

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

STATES_NUM

Parametr lokalny, służący do określenia liczby stanów automatu zastosowanego w spi_exe_unit

STATE_READY

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

STATE_LOAD

Stan, w którym dane z zmiennej s_wyniki są przypisywane do odpowiednich zmiennych pomocniczych, które następnie przypisywane są do rejestrów. Dodatkowo w tym stanie blokowany jest zapis nowej ilości cykli do układu watchdog, co pozwala na jego odpowiednią pracę.

STATE_FLAGS

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

STATE_SEND

Stan, w którym wartości zapisane w rejestrach są podawane na równoległe wyjście slave'a.

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 reg_argA

s_argB_next - sygnał przechowujący kolejną wartość sygnału reg_argB

s_oper_next - sygnał przechowujący kolejną wartość sygnału reg_oper

s_results_next - sygnał przechowujący kolejną wartość sygnału s_results

s_flags_next - sygnał przechowujący kolejną wartość sygnału s_flags

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 – sygnał przechowujący ciąg 20 bitów otrzymanych z mastera, które są przypisywane do odpowiednich sygnałów pomocniczych. Wyjście równoległe z shifter’a.

s_watchdog_we – zezwolenie na wpis do modułu watchdog

s_wrt – sygnał pozwalający do wpis równoległy/szeregowy do shifter’a

s_data – sygnał służący do zapisywania wyników rejestrów 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 otrzymuje kolejno pojedyncze bity od master’a i dzięki temu modułowi są one pojedynczo wsuwane i przypisywane do sygnału pomocniczego s_wyniki.

watchdog

Moduł, którego 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 zadaniem 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_1

W pierwszym stanie pracy spi_exe_unit_1 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.

Dane podane do testowania

Lp	I_argA	I_argB	I_oper	I_result	I_flags
.					
1	0100	0001	0000	0101	0000
2	1010	0101	0010	1111	0001
3	1100	0011	0001	0000	0000
4	1000	0000	0111	0011	0000
5	0010	0010	0011	1000	0110
6	0010	1000	0100	0000	0000
7	1010	0000	0101	1111	0001
8	0001	0101	0110	0101	0000
9	1101	0010	1011	1011	0100
10	1101	0010	1000	1010	0000

Dane uzyskane przy pracy spi_exe_unit zostały przetworzone poprawnie.