

RAPORT

Lista członków zespołu

1. Gronowski Adrian
2. Klimaszewski Szymon
3. Krupski Konrad Tomasz
4. Lisovyi Ivan

Nazwy jednostek SPI

`spi_exe_unit_1`

Realizacja: Konrad Krupski

`spi_exe_unit_2`

Realizacja: Ivan Lisovyi

`spi_exe_unit_3`

Realizacja: Szymon Klimaszewski

`spi_exe_unit_4`

...

Decyzje projektowe

Komunikacja równoległa pomiędzy master – slave

Jako zespół projektowy zdecydowaliśmy, aby komunikacja między master a slave odbywała się w sposób równoległy. Działanie to miało na celu ograniczyć liczbę bitów potrzebnych do komunikacji pomiędzy układami. Jeżeli komunikacja miałaby się odbywać w sposób szeregowy to liczba bitów przechowywanych w rejestrze master'a oraz slave'a musiałaby wynosić 80. Dzięki realizacji komunikacji równoległej, ta liczba została ograniczona do 20.

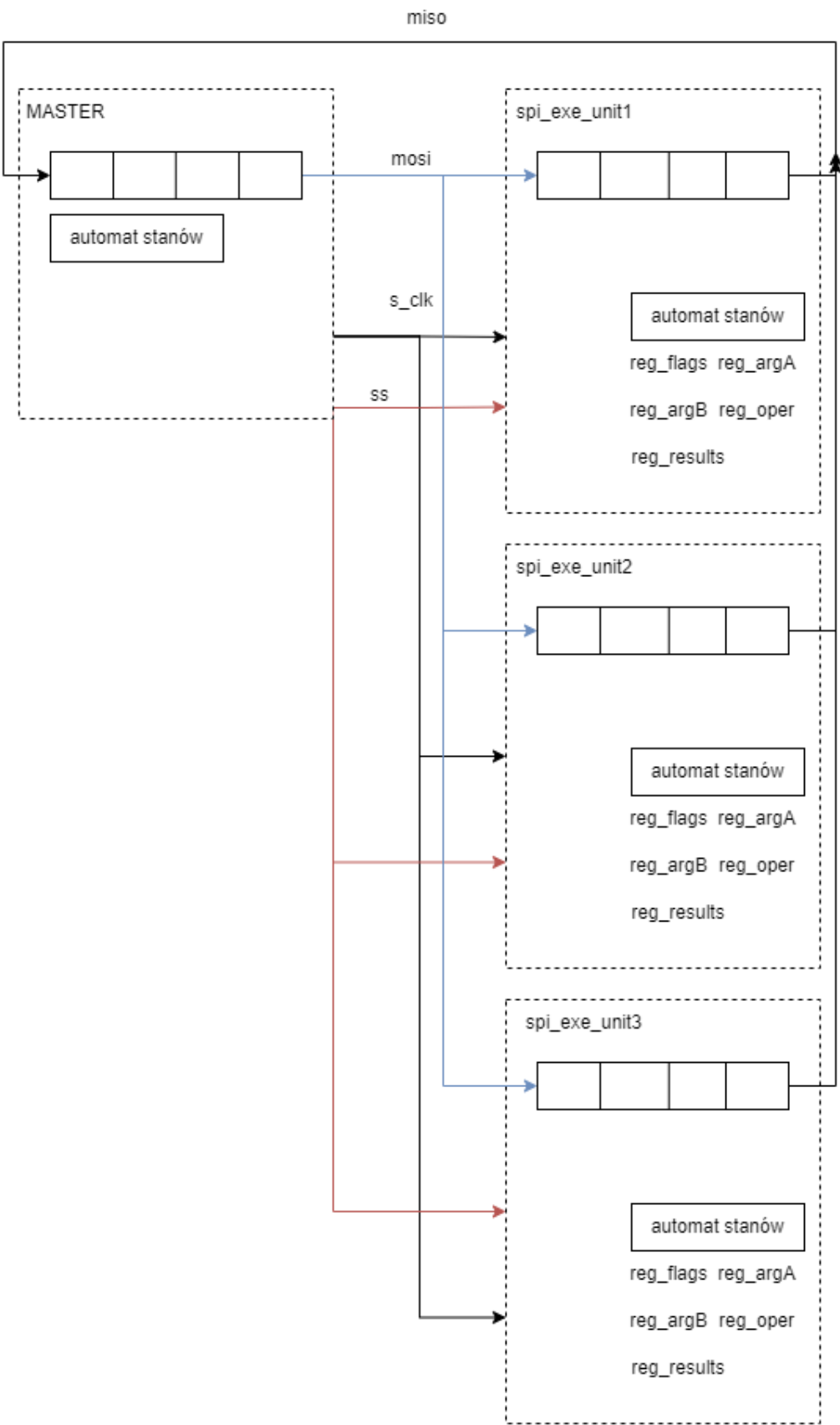
20-bitowe rejestry slave'a i master'a

Realizując projekt zdecydowaliśmy się na 20-bitowe rejestry slave'a i master'a. Wynika to z faktu, że zakładając pracę jednostki `exe_unit` na 4 bitach, potrzebujemy 20 bitów na zapisanie informacji o `i_argA`, `i_argB`, `i_oper` oraz `o_result`, `o_flags`.

Większa liczba stanów w jednostce master

Jako zespół projektowy zdecydowaliśmy się na zaimplementowanie dodatkowego stanu `STATE_LATE` w jednostce master. Stan ten ma na celu opóźnienie przesyłania informacji pomiędzy slave a master. Ponieważ slave zawsze potrzebuje dodatkowego cyklu zegara ze względu na przejście do stanu gdzie wsuwane są rejestry argumentów do `exe_unit'a`

Schemat blokowy



Przesuwanie bitów w rejestrach master'a i slave'a

Podczas projektu została podjęta decyzja o stronie, w którą dane są przesuwane w master oraz slave. W jednostkach master oraz slave zdecydowaliśmy się, że dane będą przesuwane bitowo w prawą stronę.

4-bitowe dane w jednostkach exe_unit

W celu realizacji wspólnej komunikacji mastera z jednostkami spi_exe_unit ustaliliśmy, żeby exe_unit każdego członka pracowało na danych 4-bitowych. Pozwoliło to na ustalenie ilu bitów mają być rejestry w masterze oraz slave'ie. Niemniej jednak w celu osiągnięcia pracy na większej ilości bitów należy odpowiednio skonfigurować exe_unit oraz poszczególne jednostki.

Specyfikacja Master'a

Wejścia

i_slave_select

i_clk – sygnał sterujący zegarem master'a

i_rst – sygnał, który odpowiada za przywrócenie master'a do stanu początkowego

i_data – sygnał, który jest wejściem równoległym

i_send – sygnał pozwalający na wysyłanie danych

i_miso – master in slave out. Sygnał, który przechowuje dane pochodzące od slave'a

Wyjścia

o_data

o_busy – sygnał mówiący o tym czy master aktualnie zajęty jest komunikacją z slave'em.

o_mosi - master out slave in. Sygnał, który przechowuje dane wysyłane z master'a do slave'a

o_sclk – sygnał, który steruje zegarem slave'a

o_ss – select slave. Sygnał przechowujący wartość, która mówi o tym, z którym slave'em chcemy przeprowadzić komunikację.

Parametry

BITS – wartość, mówiąca o ilości bitów, na której operuje master

SLAVES_NUMBER – wartość, mówiąca o ilości modułów slave w procesie komunikacji

STATES_NUM – wartość, mówiąca o ilości stanów automatu master'a

STATE_READY – stan, w którym master jest gotowy do rozpoczęcia pracy. W momencie, gdy dostaje on sygnał informujący o potrzebie przestania informacji przechodzi on do stanu następnego

STATE_LOAD – stan, w którym master informuje moduł watchdog o ilości cykli potrzebnych do wykonania oraz wybiera, z którym slave'em ma przebiegać komunikacja

STATE_HIGH – stan odpowiadający za generację sygnału zegarowego

STATE_LOW - stan odpowiadający za generację sygnału zegarowego

STATE_END – stan mówiący o końcu przesuwania danych w masterze

STATE_SS – stan w którym wzbudzany jest slave

STATE_LATE – stan wprowadzający opóźnienie w postaci pojedynczego taktu zegara

Sygnały pomocnicze

s_sout_en – pozwolenie na wpis

s_sout_wrt – wpis równoległy bądź szeregowy do shiftera

s_inter – sygnał mówiący o końcu pracy modułu watchdog

s_watchdog_we – sygnał, który przechowuje wartość wprowadzaną do modułu watchdog

s_bit – sygnał przechowujący wartość w postaci pojedynczego bitu, który jest podawany na wyjście master'a

s_bit_in – sygnał przechowujący wartość w postaci pojedynczego bitu, który jest podawany z wejście master'a z konkretnego slave'a

Instancje modułów

shifter

Moduł jest odpowiedzialny za przesuwanie bitów otrzymanych z slave'a. Master otrzymuje kolejno pojedyncze bity od slave'a i dzięki temu modułowi wartości są odpowiednio przesuwane w rejestrze master'a.

watchdog

Moduł, którego zadaniem jest pobudzenie master'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.