

Projekt zespołowy SCK

Projekt wykonawczy modułu z komunikacją za pomocą protokołu komunikacji szeregowej z magistralą wewnętrzną

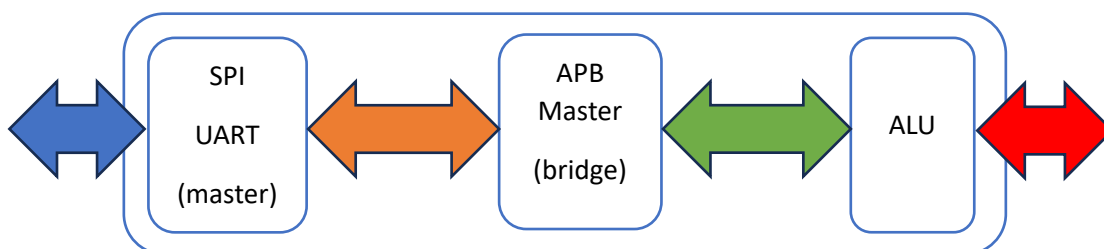
Semestr 2023Z

Cel projektu

Celem projektu jest stworzenie układu cyfrowego, który komunikuje się ze światem zewnętrznym za pomocą wybranego protokołu komunikacji szeregowej. Dobór protokołu komunikacji jest opisany w dalszej części dokumentu.

Studenci, w ramach zespołów, będą musieli stworzyć system, w którym należy dorobić kontroler (mostek) magistrali APB oraz dodatkowy moduł komunikacyjny będący jednostką podlegającą kontrolerowi APB (z możliwością rozszerzenia peryferyjnych modułów podłączonych do kontrolera APB).

Zadanie



Ten prosty schemat blokowy pokazuje zamysł systemu. Należy zaimplementować dodatkowy moduł komunikacji SPI lub UART (opisane dalej, razem z sposobem wyboru), moduł kontrolera magistrali APB (dokument opisujący działanie magistrali APB w załączniku do projektu grupowego), który będzie zarządzać modułem UART w celu wysłania danych otrzymanych z spreparowanej jednostki ALU.

Jednostka ALU ma powstać z **połączenia operacji spośród wszystkich indywidualnych projektów**. Jeżeli indywidualne projekty robione były na różne systemy kodowania liczb binarnych, to te operacje mają działać dalej dla tych samych kodowań. Powstaną w ten sposób układy obsługujące operacje na różnych kodach. Należy do tak powstałego ALU dodać dodatkowy port:

1. **o_op_rdy** – wyjście jednobitowe, sygnalizujące kontrolerowi APB, że ma dla niego prawidłowe dane do wysłania

Dodatkowo, ALU ma zostać przystosowane do działania domyślnie na 8 bitach, jednakże ma zostać zaprojektowana z wykorzystaniem parametrów generycznych.

Sam układ ma działać w następujący sposób:

1. Układ (kontroler APB) ma czekać w stanie oczekiwania na informację z ALU, że wykonana została jakaś operacja, że dane są poprawne i nie zgłaszają błędów. (operacja w domyśle zadana spoza przedstawionego układu, powiedzmy z dalszej części układu cyfrowego z innego miejsca – w tym przypadku z testbencha)
2. Gdy taki sygnał pojawia się na wejściu, czyli gdy linia **o_op_rdy** wychodząca z ALU i wchodząca do uproszczonego mostka APB będzie w stanie wysokim, a samo ALU nie będzie zgłaszać błędów, kontroler APB ma rozpocząć **operację wpisywania** do kontrolera UART/SPI (protokół komunikacji do wyboru w tabeli nr 1 wg przedstawionego w niej wzoru) po magistrali danych **PWDATA** (opisanej w załączniku odnośnie do magistrali APB).
3. Gdy kontroler komunikacji odbierze dane, kontroler APB przechodzi w stan czytania i czyta magistralę **PRDATA** (opisaną w załączniku odnośnie do magistrali APB), czekając aż na linii kontroler komunikacji poinformuje o zakończeniu transferu i gotowości do przesłania kolejnych danych.
4. Gdy to się stanie, kontroler APB ma za zadanie poinformować układ na zewnątrz, że do ALU można wysłać nowe dane i nakazać wykonanie nowej operacji za pomocą wyjścia **o_waiting**, jak również na wyjściu **o_transfer_done**, że procedura wysłania zakończyła się powodzeniem
5. Jeżeli nastąpił jakikolwiek błąd, wówczas ustawiane jest tylko wyjście **o_waiting**.

Pozostała funkcjonalność ALU ma pozostać niezmieniona w stosunku do zadań indywidualnych.

Kontroler AMBA APB

Zgodnie z załącznikiem o magistrali APB, kontroler (mostek) magistrali APB ma za zadanie obsługiwać w pełni funkcjonalnie procedury *write transfers with wait states* oraz *read transfers with wait states*. Różnica między transferami z czekaniem i bez czekania jest taka, że kontroler APB czeka na ustawienie linii PREADY przez wybrane urządzenie podrzędne w stan wysoki zanim przejdzie do kolejnego etapu (na przykład z *write* do *read transfer*). Dokładny opis w dokumentacji APB.

Kontroler APB ma posiadać następujące linie ze standardu:

1. **PCLK** – (wejście) zegar
2. **PRESET** – (wejście) reset synchroniczny
3. **PADDR** – (wyjście) 2 bitową magistralę adresową – pozwalającą w ten sposób na ewentualne rozszerzenie projektu o dodatkowe moduły komunikacyjne (pomimo, iż używać będzie na razie tylko jednego peryferyjnego modułu komunikacji szeregowej)
4. **PSELx** – (wyjścia) pojedynczy bit dla każdego możliwego urządzenia (2 bitowa magistrala adresowa, czyli maksymalnie 4 urządzenia)
5. **PENABLE** – (wyjście) sygnalizacja dla modułu podrzędnego, że może przyjąć dane, sygnalizacja rozpoczęcia właściwej części transferu
6. **PWRITE** – (wyjście) Sygnalizacja kierunku obsługi magistrali **PWDATA** i **PRDATA**, zgodnie z opisem APB
7. **PWDATA** – (wyjście) m-bitowa magistrala danych z kontrolera do urządzeń podrzędnych
8. **PREADY** – (wejście) sygnalizacja z urządzenia podrzędnego
9. **PRDATA** – (wejście) 1-bitowa magistrala danych z urządzeń podrzędnych do kontrolera, sygnalizacja poprawnie wykonanego transferu

Pozostałe, nieopisane linie z tabeli 2-1 z załączonego dokumentu o APB nie obowiązują, co za tym idzie, nie obowiązują dotyczące ich procedury (na przykład, priorytet).

Dodatkowo, układ kontrolera APB ma posiadać następujące porty do komunikacji z ALU i systemem zewnętrznym:

1. **i_data** – m-bitowe wejście danych z ALU, o szerokości dopasowanej do wyjścia ALU
2. **i_data_ready** – 1-bitowe wejście, po którym ALU informuje o tym, że dane są gotowe do odbioru i wysłania, (jeżeli nastąpił błąd obliczeń w ALU, nie rozpoczynamy komunikacji, nie wysyłamy danych)
3. **i_alu_error** – 1-bitowe wejście sprawdzające, czy w ALU wystąpił błąd. Jeżeli wystąpił, transfer nie rozpoczyna się
4. **i_protocol_sel** – 2 bitowe wejście wyboru protokołu komunikacji, do którego ma zostać wysłany rozkaz przesłania danych, wybierany z testbench'a
5. **o_waiting** – 1-bitowe wyjście sygnalizujące poza cały moduł projektu, że jest on gotowy na przyjęcie nowych danych do ALU i wysłanie, ustawiany w stan wysoki, gdy układ może przyjąć dane, odbierany w testbench'u
6. **o_transfer_done** – 1 bitowe wyjście ustawiane razem z **o_waiting** tylko, jeżeli nastąpił transfer poprawnych danych z ALU

Kontrolery protokołu komunikacji szeregowej

Wybór komunikacji szeregowej następuje poprzez zsumowanie wszystkich cyfr indeksów wszystkich członków zespołów i wykonaniu operacji modulo 2. Następnie wybieramy moduł kontrolera komunikacji z tabeli nr 1

Tabela nr 1

Wynik modulo 2 = 0	Wynik modulo 2 = 1
UART	SPI mode 3
Tylko wyjście TX z protokołu komunikacji	Tylko wyjścia MOSI i SCLK z protokołu komunikacji
Pozostałe wejścia i wyjścia zgodnie z liniami ze standardu opisanymi dla kontrolera AMBA APB	
Dodatkowo: 1 bit parzystości oraz 1 bit stop	Dodatkowo: -

Ocena

Projekt zespołowy oceniany jest na **30 punktów**. Nie jest on tak dokładnie podzielony jak w przypadku projektów indywidualnych. W przypadku tego projektu oceniany będzie:

1. Równomierne zaangażowanie i zespołowa praca
2. Implementacje rozwiązań napotkanych problemów
3. Prowadzenie wspólnego repozytorium
4. Model, jakość kodu, spójność stylu kodowania modeli
5. Modułowość, czytelność, łatwość w zrozumieniu zasady działania
6. Stworzona dokumentacja projektu

Zapraszam na konsultacje online. W styczniu 2024 w każdy czwartek i niektóre wtorki będę dostępny dla Państwa fizycznie w pokoju 363 na WEiTl.