

Warszawa, 23.01.2024



Projekt zespołowy - komunikacja magistralą APB z
układami apb_exe_unit poszczególnych członków
zespołu projektowego

Grzegorz Fabisiak, Alicja Misterka, Rafał Najda

Cel projektu

Celem projektu jest połączenie modułów jednostek wykonawczych z projektów indywidualnych poszczególnych członków zespołu projektowego wspólną magistralą AMBA APB. Jednostki mają współpracować z kontrolerem abp_master.

Wstęp

Magistrala APB (*Advanced Peripheral Bus*) jest kluczowym elementem w architekturze systemów mikroprocesorowych. Stanowi efektywną komunikację pomiędzy mikroprocesorem a pozostałymi peryferiami systemu komputerowego.

Model magistrali APB tworzony podczas projektu zespołowego jest stworzony z XXX modułów:

- *Memory model* odpowiadający za czytanie danych z pliku i zapisywanie ich.
- *APB master* odpowiadający za inicjowanie wymiany danych i kontrolowanie dostępu do magistrali.
- *APB slave* odpowiadający za reagowanie na sygnały wysyłane przez mastera i realizowanie operacji zgodnie z żądaniem; w przypadku projektu zespołowego każdy ze slave'ów jest jednostką wykonawczą wykonaną przez studentów podczas projektów indywidualnych.
- *Arbiter* odpowiadający za komunikację pomiędzy slave'ami (przychodzącymi) i masterami (wychodzącymi).
- *Bus* odpowiadający za komunikację kontrolera z slave'ami, a dokładnie za przesył poleceń oraz danych do slave'ów.

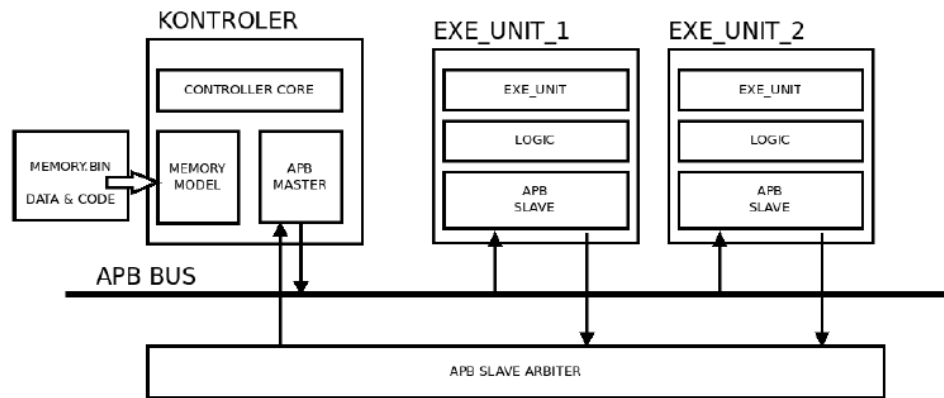
Komunikacja pomiędzy *slave* oraz *master* odbywa się za pomocą poszczególnych sygnałów przychodzących oraz wychodzących:

- *PLCK*
Sygnał zegarowy wchodzący zarówno do każdego ze *slave* jak i *mastera*.
- *PRESETn*
Sygnał resetujący aktywowany niskim stanem logicznym (opadającym zboczem zegara), wchodzącym do każdego ze *slave* jak i do *mastera*.
- *PSELx*
Sygnał służący do wyboru *slave'a*, który ma działać w danym momencie. Posiada 3 bity po jednym na każdy *slave* aby sterować nimi niezależnie.
- *PADDR*
Sygnał odpowiadający za przechowywanie adresu pamięci używanego do określenia z którą komórką pamięci (lub rejestrem) będzie komunikować się magistrala. Wychodzący z *mastera*, odbierany przez każdego ze *slave'a*.
- *PWDATA*
Sygnał odpowiadający za przenoszenie danych wysyłanych przez kontroler do slave'ów.
- *PRDATA*
Sygnał odpowiadający za przenoszenie danych wysyłanych przez *slave'y*.
- *PWRITE*
Sygnał określający czy dana wymiana danych na magistrali APB jest operacją zapisu, czy też nie. Wychodząca z *mastera*

- *PREADY*
Sygnał mówiący o aktualnym stanie *slave'ów* (czy są zajęte czy gotowe na wykonywanie kolejnych poleceń).
- *PSLVERR*
Sygnał służący do wystawiania flag błędów. W naszym przypadku wystawia on stan statusów (*o_status*) w modułach *exe_unit*.

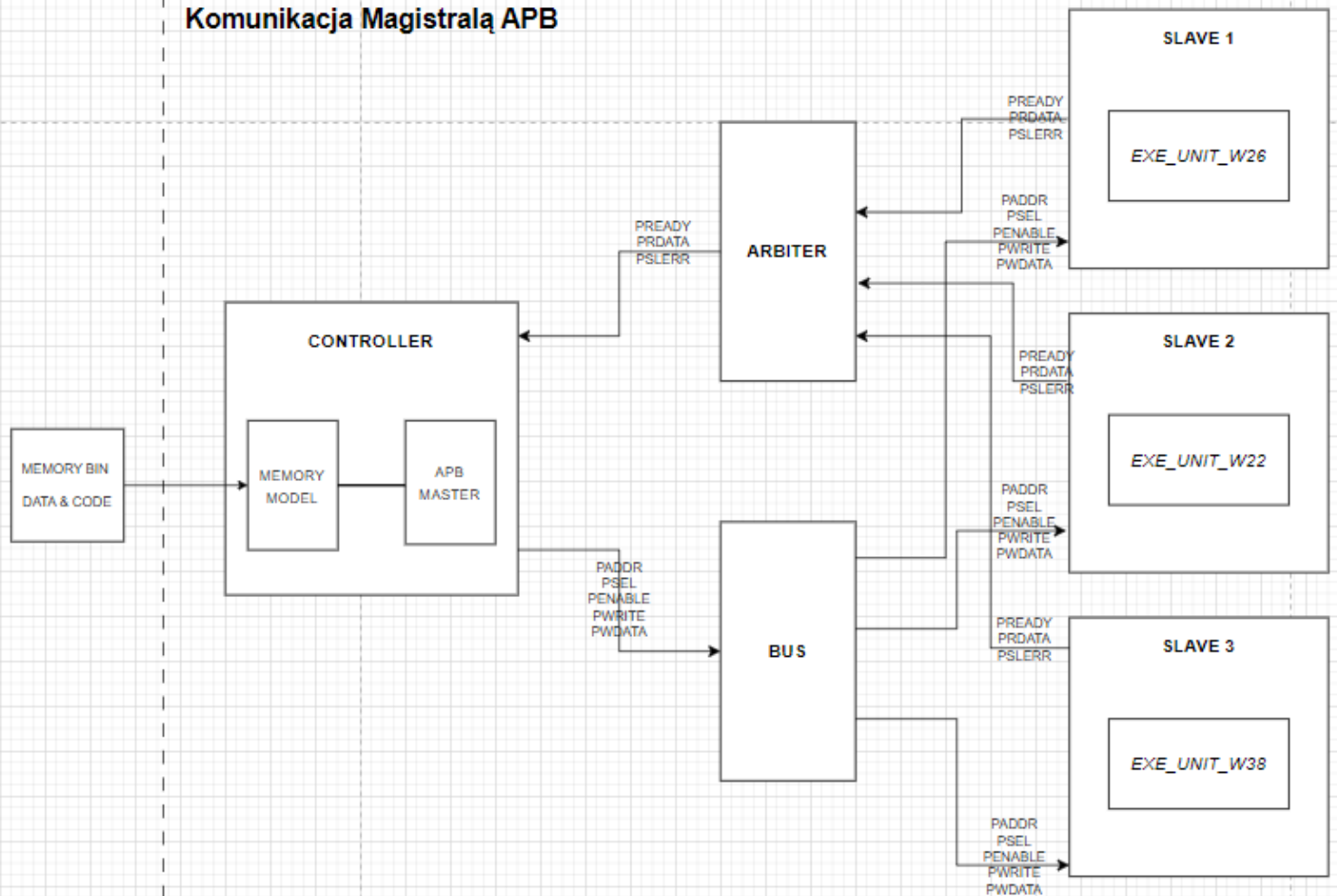
Realizacja

1. schemat projektu



Schemat blokowy dołączony do zadania

Komunikacja Magistralą APB



1. Sposób implementacji magistrali

Przy otrzymaniu instrukcji do wykonania zadania, otrzymaliśmy także gotowy moduł *APB_master* oraz *memory_model* który użyliśmy do stworzenia działającej magistrali. Zanim przeszliśmy do właściwej implementacji, uzgodniliśmy wartości parametrów:

SEL_WIDTH = 3 - jeden bit dla każdego ze *slave'ów*.

ADDR_WIDTH = 2 - ze względu na dwubitowość operacji wykonywanych przez *slave'y*.

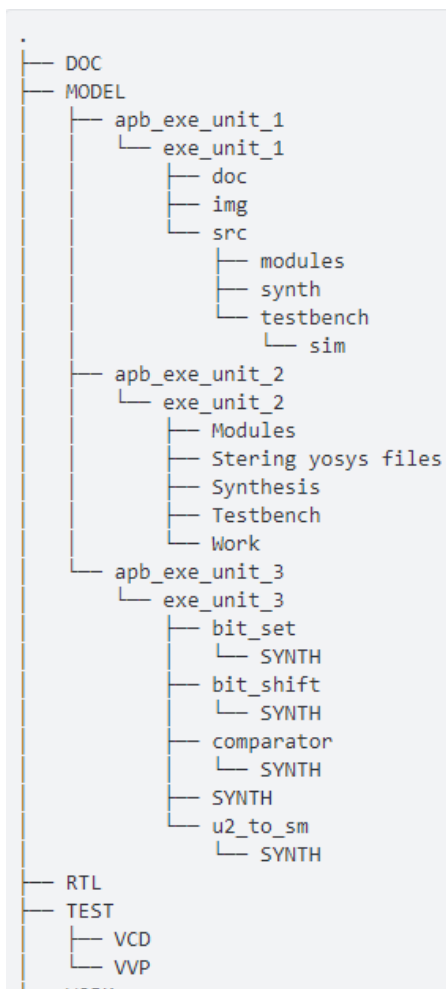
DATA_WIDTH = 32 - tak jak było w oryginalnym module master

SEL_BIT = 3 - jeden bit dla każdego ze *slave'ów*.

Stworzyliśmy także dodatkowe pliki wymagane w naszej implementacji do stworzenia magistrali:

- Model - magistrala APB (połączenie wszystkich podmodułów w jedną całość)
- Controller - implementacja połączeń między MEMORY MODEL oraz APB MASTER
- Slave - implementacja jednostek *exe_unit*
- Arbiter, bus - implementacja sposobu przesyłu danych pomiędzy controllerem a *slave'ami*

2. Wskazanie plików, konwencja oznaczeń



Na grafice przedstawimy podział projektu (plików) na poszczególne podkatalogi:

- *DOC*, w którym znajdują się dokumentację projektu i jednostek ALU wykonanych podczas projektów indywidualnych.

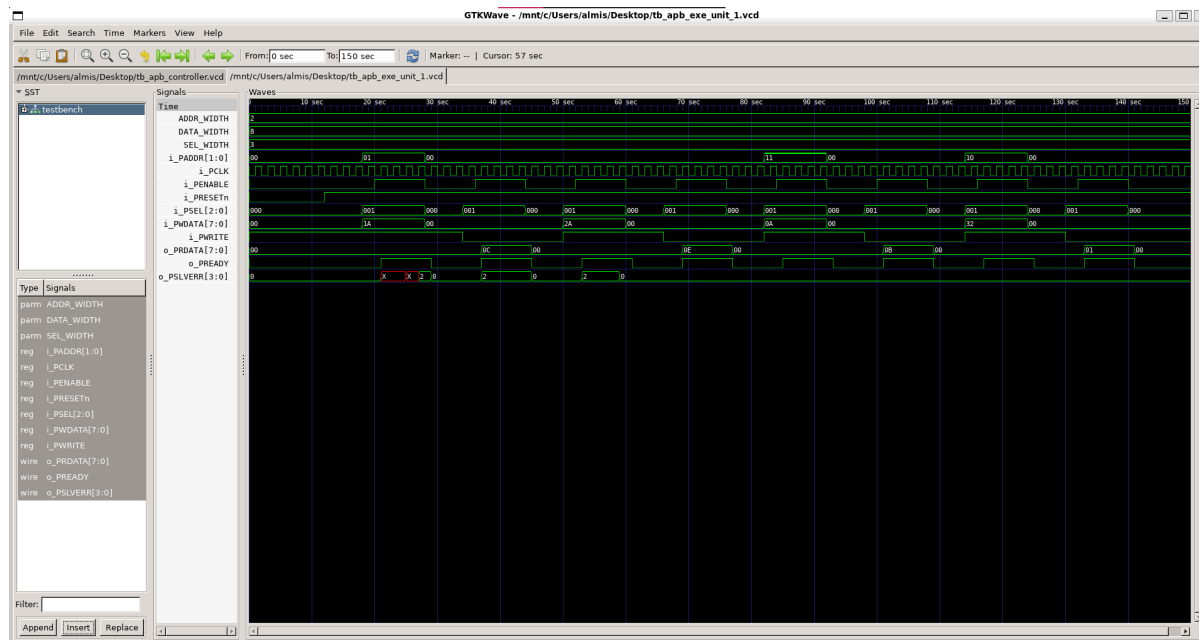
- *MODEL* w którym znajdują się podkatalogi poszczególnych układów *exe_unit* oraz inne potrzebne moduły do wykonania magistrali APB: bus, arbiter, controller, slave, master, memory model oraz ogólny moduł łączący wszystko w całość.

- *TEST* w którym znajdują się pliki typu testbench.sv.

- *RTL* w którym znajdują się pliki wynikowe syntezy logicznej

- *WORK* zawiera skrypty syntezy, symulacji oraz wyniki przebiegów.

3. Wyniki symulacji testbench Magistrali APB



gtkwave - symulacja testbench jednego ze slave'ów

4. Podział obowiązków podczas projektu:

Kolejność została podana alfabetycznie, niezależnie od liczby wykonanych commitów. Ponadto, większość projektu została wykonana “stacjonarnie”, przez co commity w większości były wykonywane na jednym komputerze.

Imię i nazwisko	nr indeksu	Zakres wykonywanych obowiązków:
Grzegorz Fabisiak	325000	szkielet modułów, praca nad <i>bus</i> oraz <i>master</i> , pliki typu <i>testbench</i> , dokumentacja
Alicja Misterka	325045	szkielet modułów, praca nad <i>controller</i> oraz <i>slave</i> , pliki typu <i>testbench</i> dokumentacja
Rafał Najda	325046	szkielet modułów, praca nad <i>arbiter</i> , <i>memory</i> , pliki typu <i>testbench</i> dokumentacja