**APB Master VIP**

**Design Specification**

**Revision 0.1**

# REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Author | Date | Description of changes |
| 0.1 | ThinhLe | 03/09/2020 | Update content 8. User Manual |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# TABLE OF CONTENTS

# Overview

# Block Diagram

# Class Diagram

# Type Description

|  |  |
| --- | --- |
| APB\_WRITE | Generate write transfer on APB bus |
| APB\_READ | Generate read transfer on APB bus |
| APB\_WRITE\_PROT | Write protection for all register |

# Class Description

## vip\_apb\_master\_transaction

## vip\_apb\_master\_sequencer

## vip\_apb\_master\_driver

**Execution process**



Figure. vip\_apb\_master\_driver execution process

APB

**\_intf.paddr     <= \_item.\_trans\_addr;**

**\_intf.pprot     <= \_item.\_trans\_prot;**

**\_intf.psel      <= '1;**

**\_intf.penable   <= '0;**

**\_intf.pwrite    <=(\_item.\_trans == APB\_WRITE);**

**//     \_intf.pwdata    <= \_item.\_trans\_wdata;**

**\_intf.pstrb     <= \_item.\_trans\_strb;**

foreach (pstrb[i])

if (\_intf.\_pstrb[i] == ‘1) begin

\_intf.\_pwdata[(8i + 7):(8i)] <= \_item.pwdata[(8i + 7):(8i)]

end

**@(posedge \_intf.pclk);**

**\_intf.penable  <= '1;**

**@(posedge \_intf.pclk);**

**while (!\_intf.pready) @(posedge \_intf.pclk);**

**\_item.\_trans\_rdata  = \_intf.prdata  ;**

**\_item.\_trans\_resp   = \_intf.pslverr ;**

## vip\_apb\_master\_agent

## vip\_apb\_master\_write

## vip\_apb\_master\_read

## vip\_apb\_write\_prot\_en

# Message list

# Discussion

## Oct.04.2020

1/ Chỉ cần 1 transaction duy nhất cho cả đọc và ghi. transaction này sẽ chứa cả bit điều khiển write\_protect\_en

Cấu trúc transaction như sau (đây chính là cái mà anh nói em cần phải định nghĩa rõ đầu tiên thì mới hình dung được việc điều khiển và xử lý APB interface ra sao:

rand logic pwrite; //Determine the transfer type: (0) READ or (1) WRITE

rand logic [31:0] paddr;

rand logic [31:0] pwdata;

rand logic [3:0] pstrb;

rand logic [2:0] pprot;

logic [31:0] prdata;

logic pslverr;

rand logic write\_protect\_en;

rand logic apbSeqEn; -> Bit điều khiển on/off transaction. Nếu bit này bằng 1 thì transaction mới được lái trên APB interface, nếu bằng 0 thì bở qua transaction, không dùng

rand logic apbConEn; -> Cho phép transfer liên tục với transfer trước đó (không kéo Psel xuống, chỉ lái Penabe)

rand int apbDelay; -> thiết lập delay giữa các APB transaction, phải delay bằng giá trị được thiết lập ở biến này rồi mới gửi transfer trên APB interface

2/ Các checker hãy tích hợp vào monitor, cái này không cần để riêng bên ngoài, và cần làm rõ checker sẽ kiểm tra và báo các lỗi gì (liệt kê cụ thể)?



3/ Giữa monitor và Scoreboard chỉ cần 1 kết nối TLM vì trong giao thức APB, tại một thời điểm chỉ đọc hoặc ghi,