

BÀI 9: XÂY DỰNG HỆ THỐNG SOPC VỚI BỘ DMA ĐƠN GIẢN

1. MỤC ĐÍCH

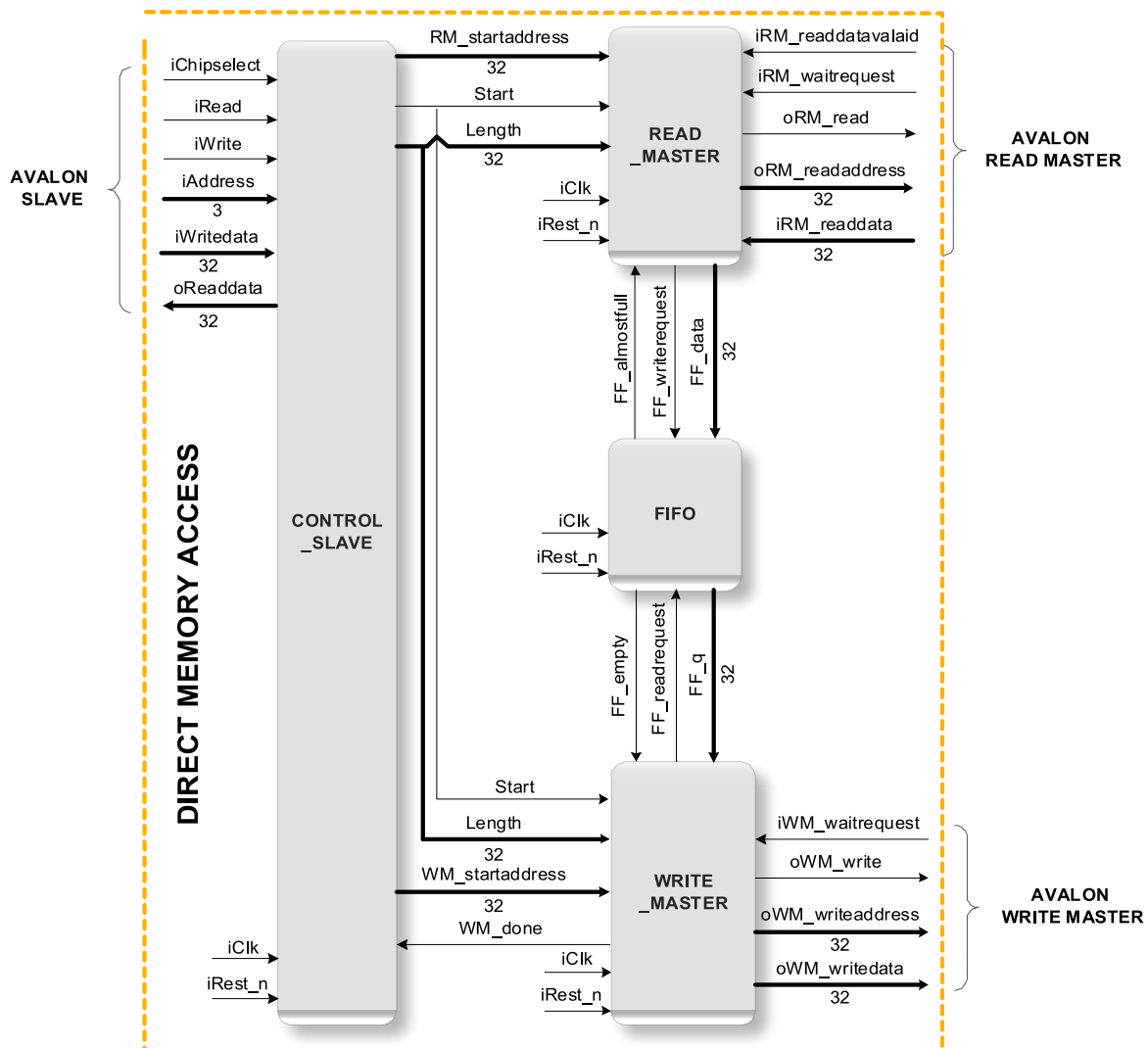
Thông qua bài thực hành này, sinh viên sẽ hiểu rõ:

- Cách xây dựng sơ đồ khối và kiến trúc tổng quan của một DMA đơn giản tự thiết kế.
- Cách thiết kế khối MasterRead trong DMA.

2. PHÂN LÝ THUYẾT

2.1 Giới thiệu DMA đơn giản

- DMA gồm 4 khối chính: Control_Slave, Read_Master, Write_Master và Fifo được mô tả như Hình 2.1.



Hình 2.1 – Sơ đồ khối của DMA Controller

- Sơ đồ chân của DMA được:

STT	Tên tín hiệu	Ngõ vào/ra	Số bit dữ liệu	Mô tả
1	iClk	In	1	Cấp xung clock 50 MHz cho DMA hoạt động
2	iReset_n	In	1	iReset_n = 0: reset lại DMA

3	iChipselect	In	1	iChipselect = 1: cho phép truy xuất vào các thanh ghi trạng thái và điều khiển của DMA
4	iRead	In	1	iRead = 1: cho phép đọc giá trị của các thanh ghi trong DMA
5	iWrite	In	1	iWrite = 1: cho phép ghi giá trị vào các thanh ghi của DMA
6	iAddress	In	3	Chọn địa chỉ của thanh ghi cần đọc/ghi
7	iWritedata	In	32	Truyền giá trị cần ghi vào thanh ghi từ Avalon bus
8	oReaddata	Out	32	Xuất giá trị cần đọc từ thanh ghi ra Avalon bus
9	iRM_readdatavalid	In	1	iRM_readdatavalid = 1: báo hiệu dữ liệu vào hợp lệ
10	iRM_waitrequest	In	1	iRM_waitrequest = 1: yêu cầu Read_Master chờ
11	oRM_read	Out	1	oRM_read = 1: yêu cầu đọc dữ liệu từ vùng địa chỉ oRM_readaddress
12	oRM_readaddress	Out	32	Truyền địa chỉ của vùng nhớ cần đọc giá trị vào
13	iRM_readdata	In	32	Truyền giá trị từ Avalon bus vào Read_Master
14	iWM_waitrequest	In	1	iWM_waitrequest = 1: yêu cầu Write_Master chờ
15	oWM_write	Out	1	oWM_write = 1: yêu cầu ghi dữ liệu vào vùng địa chỉ oWM_writeaddress
16	oWM_writeaddress	Out	32	Truyền địa chỉ của vùng nhớ cần ghi giá trị vào
17	oWM_writedata	Out	32	Truyền giá trị từ Write_Master vào Avalon bus

Hình 2.2 – Mô tả chân tín hiệu của DMA Controller

- Nguyên tắc hoạt động của DMA
 - ✓ DMA được cấu hình các giá trị ban đầu từ CPU NIOS II thông qua khối , CONTROL_SLAVE. Sau khi các thông số được thiết lập xong, CONTROL_SLAVE gửi tín hiệu bắt đầu hoạt động đến khối READ_MASTER và WRITE_MASTER.
 - ✓ Khối READ_MASTER kiểm tra xem FIFO có gần đầy hay chưa (tín hiệu FF_almostfull = 1). Nếu FIFO chưa gần đầy, READ_MASTER gửi các tín hiệu ra Avalon bus để yêu cầu đọc dữ liệu từ vùng nhớ có địa chỉ đầu lưu trong thanh ghi oRM_readaddress. Dữ liệu trả về sẽ được READ_MASTER truyền đến FIFO. READ_MASTER ngừng yêu cầu đọc dữ liệu nếu FIFO báo gần đầy.
 - ✓ Khối WRITE_MASTER kiểm tra FIFO có rỗng hay không. Nếu FIFO không rỗng (tín hiệu FF_empty ≠ 0), WRITE_MASTER đọc dữ liệu từ FIFO và truyền sang Avalon bus để ghi vào vùng nhớ có địa chỉ bắt đầu lưu trong thanh ghi oWM_writeaddress. Nếu quá trình DMA hoàn tất, WRITE_MASTER xác lập tín hiệu oWM_done để báo về CONTROL_SLAVE cũng như CPU NIOS II.

2.2 Chi tiết các khối trong DMA

2.2.1 Khối CONTROL_SLAVE

- Khối CONTROL_SLAVE nhận thông tin cấu hình từ CPU NIOS II và thiết lập các thông số cấu hình đó cho khối READ_MASTER và khối WRITE_MASTER.
- Sơ đồ chân của khối CONTROL_SLAVE

STT	Tên tín hiệu	Ngõ vào/ra	Số bit dữ liệu	Mô tả
1	iClk	In	1	Mục 2.1 – Hình 2.2
2	iReset_n	In	1	Mục 2.1 – Hình 2.2
3	iChipselect	In	1	Mục 2.1 – Hình 2.2
4	iRead	In	1	Mục 2.1 – Hình 2.2
5	iWrite	In	1	Mục 2.1 – Hình 2.2
6	iAddress	In	3	Mục 2.1 – Hình 2.2
7	iWritedata	In	32	Mục 2.1 – Hình 2.2
8	oReaddata	Out	32	Mục 2.1 – Hình 2.2
9	RM_startaddress	In	1	Truyền địa chỉ bắt đầu của vùng nhớ cần đọc đến READ_MASTER
10	Start	In	1	Start = 1: bắt đầu thực hiện DMA
11	Length	Out	1	Truyền số lượng dữ liệu cần thực hiện DMA (theo bytes) đến READ_MASTER và WRITE_MASTER
12	WM_startaddress	Out	32	Truyền địa chỉ bắt đầu của vùng nhớ cần ghi đến WRITE_MASTER
13	WM_done	In	32	Truyền từ WRITE_MASTER đến CONTROL_SLAVE để báo hiệu quá trình DMA hoàn tất

Hình 2.3 – Mô tả chân tín hiệu của module CONTROL_SLAVE

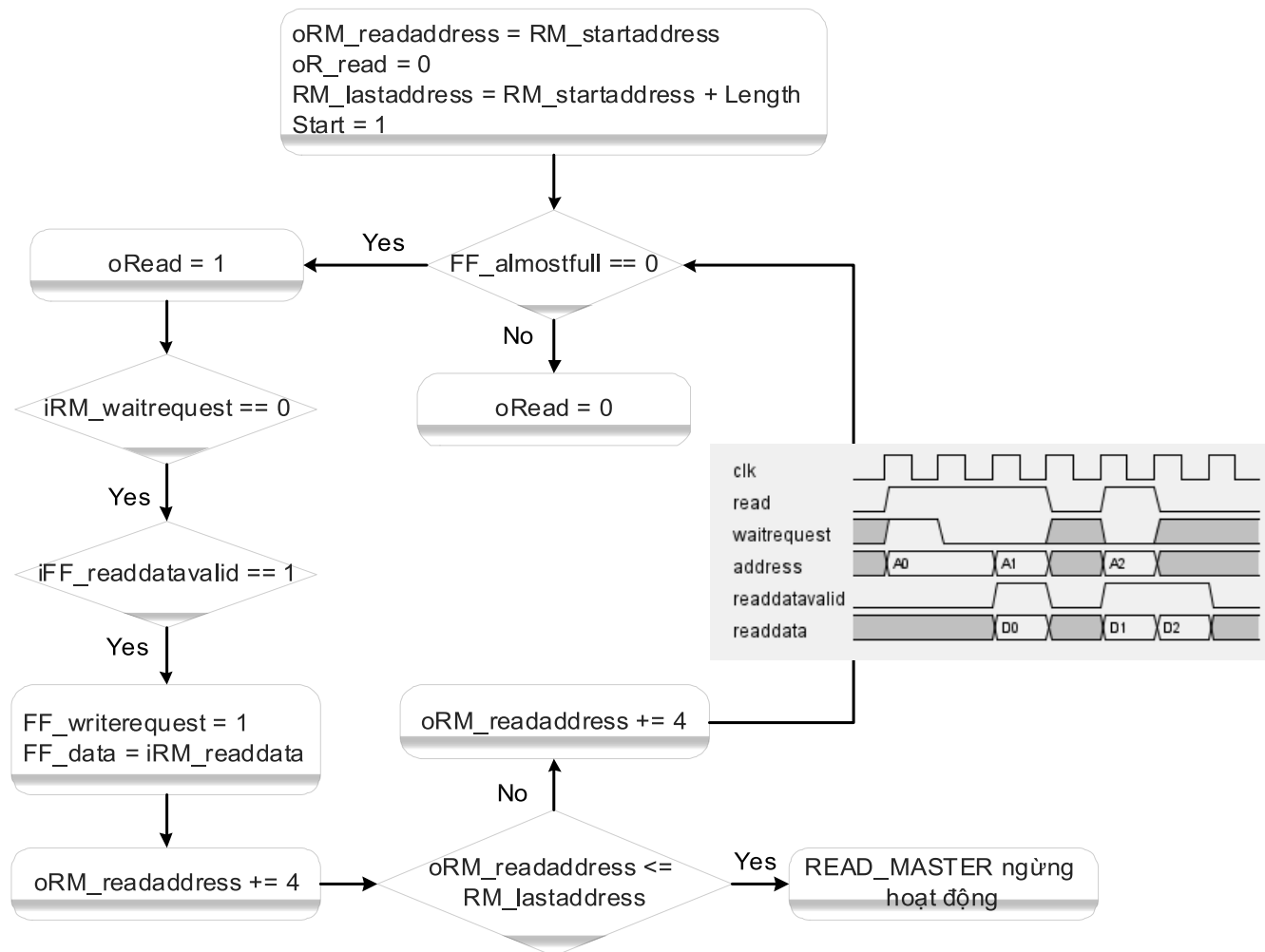
- Khối CONTROL_SLAVE gồm các thanh ghi cấu hình được mô tả ở Hình . Bit GO = 1 yêu cầu DMA hoạt động. Bit BUSY = 1 trong quá trình DMA hoạt động. Bit DONE = 1 báo hiệu quá trình DMA đã hoàn tất.

Offset	Tên thanh ghi	Đọc/Ghi	31 ... 4	3	2	1	0
0	readaddress	R/W	Địa chỉ bắt đầu của vùng nhớ cần đọc ra				
1	writeaddress	R/W	Địa chỉ bắt đầu của vùng nhớ cần ghi vào				
2	length	R/W	Số lượng bytes cần thực hiện DMA				
3	---	---	N/A				
4	control	R/W			GO		
5	status	R/W					BUSY DONE
6	---	---	N/A				
7	---	---	N/A				

Hình 2.4 – Mô tả các thanh ghi trong module CONTROL_SLAVE

2.2.2 Khối READ_MASTER

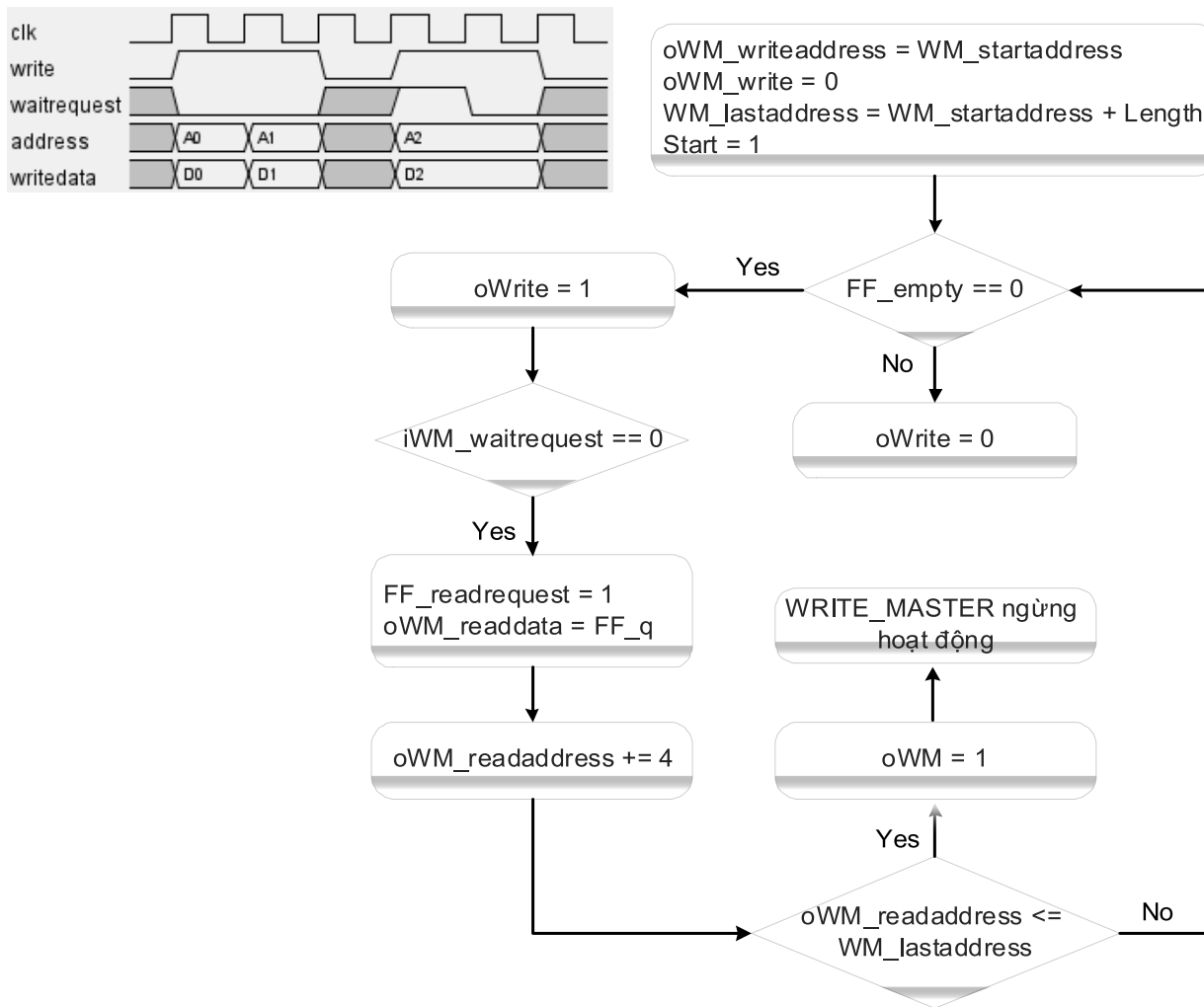
- Khối READ_MASTER yêu cầu dữ liệu từ Avalon bus và ghi vào FIFO. Nguyên tắc hoạt động của READ_MASTER được mô tả như Hình 2.5.



Hình 2.5 – Lưu đồ giải thuật mô tả hoạt động của module READ_MASTER

2.2.3 Khối WRITE_MASTER

- Khối WRITE_MASTER yêu cầu dữ liệu từ FIFO và ghi vào Avalon bus. Nguyên tắc hoạt động của WRITE_MASTER được trình bày ở Hình 2.6.



Hình 2.6 – Lưu đồ giải thuật mô tả hoạt động của WRITE_MASTER

2.2.4 Khối FIFO

- Khối FIFO làm bộ đệm dữ liệu cho việc truyền nhận giữa 2 khối READ_MASTER và WRITE_MASTER. Kích thước khối FIFO được đặt là 256 words.

BÁO CÁO THỰC HÀNH **BÀI 9: XÂY DỰNG HỆ THỐNG SOPC** **SỬ DỤNG DMA TỰ THIẾT KẾ**

Sinh viên:

Lớp: Nhóm:

Phần chuẩn bị trước ở nhà

*** Sinh viên nộp cho giảng viên trước khi vào thực hành**

BÁO CÁO THỰC HÀNH

BÀI 9: XÂY DỰNG HỆ THỐNG SOPC

SỬ DỤNG DMA TỰ THIẾT KẾ

Sinh viên:

Lớp: **Nhóm:**

Phần thực hành trên lớp

*** Sinh viên nộp cho giảng viên sau buổi thực hành**

Bài 1: Tiến hành thiết kế khối `READ_MASTER` để tạo ra được DMA hoàn chỉnh. Sau đó, kết nối DMA vào hệ thống SoPC để thực hiện việc đọc dữ liệu từ On-chip memory 4 KB và ghi vào SRAM trên board DE2. So sánh kết quả mô phỏng dùng Modelsim và kết quả thực nghiệm dùng SignalTap.