

## THIẾT KẾ VÀ THI CÔNG BỘ TRUYỀN NHẬN THEO GIAO THỨC I2C DESIGN AND VERIFICATION OF AN I2C CONTROLLER

Dương Minh Trí, Nguyễn Hoàng Kỳ Anh, Nguyễn Chí Phúc,  
Nguyễn Đình Cao Tú, Trần Minh Trí, Huỳnh Hoàng Hà, Đỗ Duy Tân  
Trường Đại học Sư phạm Kỹ thuật TP.HCM, Việt Nam

Ngày toà soạn nhận bài 4/5/2021, ngày phản biện đánh giá 4/6/2021, ngày chấp nhận đăng 25/6/2021.

### TÓM TẮT

*I2C (Inter-Integrated Circuit) là một chuẩn truyền dữ liệu nối tiếp đồng bộ được sử dụng rộng rãi để kết nối nhiều IC (Integrated Circuit) với nhau hay kết nối các IC và các ngoại vi với tốc độ trung bình thấp trong các hệ thống số. Điểm mạnh của chuẩn I2C là khả năng kết nối giữa vi xử lý trung tâm và nhiều ngoại vi với phần cứng đơn giản. Trong nội dung bài báo này, chúng tôi trình bày quá trình thiết kế một module I2C giao tiếp qua chuẩn bus AMBA APB (Advanced Microcontroller Bus Architecture - Advanced Peripheral Bus). Cụ thể, module I2C được thiết kế bằng ngôn ngữ mô tả phần cứng Verilog, có thể cấu hình là Master hoặc Slave và hỗ trợ nhiều tốc độ truyền dữ liệu khác nhau. Nhiều kết quả đánh giá qua mô phỏng được trình bày để xác thực chất lượng của module I2C được thiết kế.*

**Từ khóa:** chuẩn I2C; FPGA; giao tiếp số; AMBA; mô phỏng.

### ABSTRACT

*I2C (Inter-Integrated Circuit) is a synchronous serial data transmission standard widely used to connect multiple ICs (Integrated Circuits) together or to connect ICs and peripherals with low-to-medium speed in digital systems. The main advantage of the I2C standard is the ability to connect the central processor and multiple peripherals with simple hardware. In this paper, we present the design process of an I2C module communicating via AMBA APB (Advanced Microcontroller Bus Architecture - Advanced Peripheral Bus) bus standard. Specifically, the I2C module is designed using Verilog hardware description language, can be configured as a Master or a Slave, and supports various data transmission rates. Extensive simulation results are presented to verify the performance of the designed I2C module.*

**Keywords:** I2C standard; FPGA; digital interface; AMBA; testbench.

### 1. GIỚI THIỆU

Inter-Integrated Circuit (I2C) là một chuẩn bus nối tiếp được phát triển bởi hãng sản xuất linh kiện điện tử Philips. Do tính ưu việt và đơn giản, I2C đã được chuẩn hóa và sử dụng rộng rãi trong các module truyền thông nối tiếp của vi mạch tích hợp ngày nay. Bus I2C được sử dụng cho rất nhiều loại IC khác nhau như các loại vi điều khiển 8051, PIC, AVR, ARM, chip như RAM tĩnh (static RAM), EEPROM, bộ chuyển đổi tương tự-số (ADC), số-tương tự (DAC), IC điều khiển LCD...

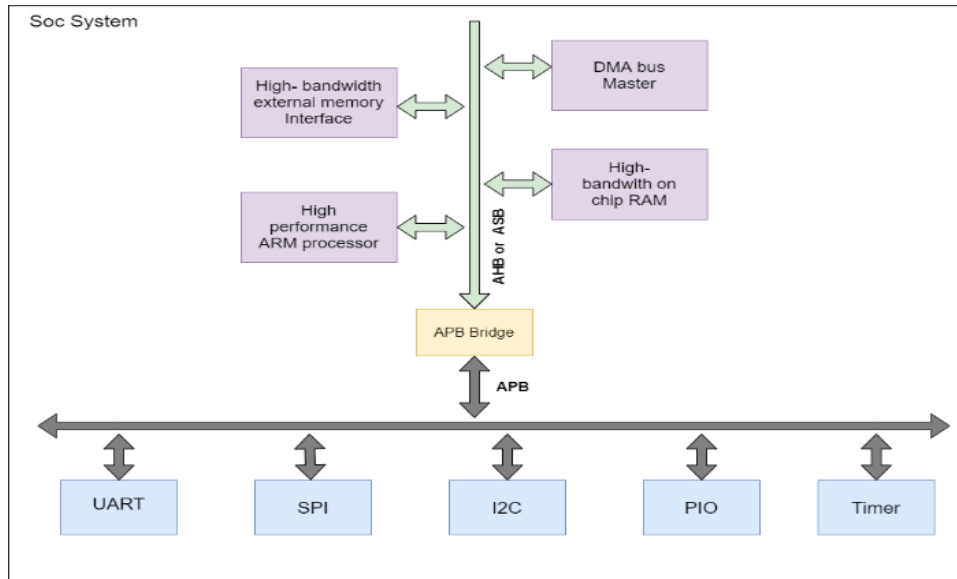
Thiết kế bus I2C sử dụng ngôn ngữ mô tả phần cứng Verilog [1-2] đã được trình bày

trong nhiều bài báo [3-8]. Trong bài báo [3], các tác giả đã thực hiện thiết kế chuyển đổi từ giao tiếp SPI sang I2C nhằm mục đích chuyển đổi giao tiếp cho thiết bị gửi có tốc độ truyền nhận cao sang thiết bị nhận có tốc độ truyền nhận thấp. Trong bài báo [4] và [5], các tác giả đã thiết kế bus I2C có khả năng đáp ứng được nhiều Master trên cùng bus bằng việc sử dụng máy trạng thái (FSM) để thực hiện quyết định ưu tiên giữa các Master điều khiển I2C khi hoạt động cùng một thời điểm. Các tác giả trong bài báo [6] và [7] đã trình bày việc triển khai bộ điều khiển I2C trên FPGA để kết nối ADC thông qua bus I2C nhằm mục đích đồng bộ hóa dữ

liệu giữa ADC và FPGA. Thêm vào đó, bài báo số [8] đã trình bày cách thiết kế và thi công chuẩn I2C dựa trên giao thức Wishbone. Tuy nhiên, các thiết kế trong [3-8] chưa trình bày chi tiết thiết kế hoặc bao gồm các thiết kế đơn giản, chưa có nhiều cấu hình và hỗ trợ nhiều tốc độ truyền-nhận.

Để cải thiện các hạn chế nêu trên, bài báo này trình bày một thiết kế module I2C sử

dụng giao tiếp bus AMBA APB (Advanced Microcontroller Bus Architecture-Advanced Peripheral Bus) của ARM [9] có thể cấu hình là Master hoặc Slave, đồng thời hỗ trợ nhiều tốc độ truyền dữ liệu khác nhau. Hình 1 minh họa hệ thống System-On-Chip (SoC) sử dụng bus AMBA và module I2C. Thiết kế được đánh giá và mô phỏng trên ISE Design Suite 14.7, QuestaSim 10.2 và Quartus II 13.0.



Hình 1. Hệ thống SoC sử dụng AMBA bus và module I2C.

## 2. THIẾT KẾ CHI TIẾT HỆ THỐNG

Module thực hiện truyền nhận dựa trên tín hiệu cấu hình, các tệp thanh ghi và các bộ đệm FIFO có nhiệm vụ lưu các dữ liệu cấu hình và thực hiện đồng bộ dữ liệu, cuối cùng là module giao tiếp với bus I2C. Trong bài báo này, chúng tôi thực hiện thiết kế module giao tiếp I2C có khả năng giao tiếp tối đa 128 thiết bị và cấu hình được 4 tốc độ truyền. Sơ đồ khối của một module I2C được thể hiện trong Hình 2.

### 2.1 Module APB Interface

Module APB Interface có chức năng giao tiếp với bus APB đồng thời điều khiển ghi/đọc dữ liệu. Khối APB Interface gồm Register File (tệp thanh ghi) chứa 10 thanh ghi có độ rộng 8 bit, các thanh ghi được cấu hình bởi tín hiệu điều khiển từ bus APB, tạo ra các tín hiệu điều khiển cho các khối còn lại của module I2C hoạt động và khối điều

khiển ngắt tạo ra tín hiệu ngắt cho khối I2C (`i2c_if`) được tạo ra từ tín hiệu tổng hợp của ngắt truyền, ngắt nhận và ngắt tràn.

### 2.2 Module Clock Divider

Module Clock Divider tạo tín hiệu xung clock với tần số mong muốn cung cấp cho bus I2C. Khối Clock Divider hoạt động dựa theo tín hiệu kích hoạt `i_clock_en` và tín hiệu bất hoạt `i_clock_stop` được tạo ra từ khối I2C Interface.

Tần số của đường SCL do Master tạo ra được lựa chọn dựa vào tín hiệu `apb_csel` được lấy từ các thanh ghi cấu hình.

### 2.3 Bộ đệm TX FIFO và RX FIFO

TX FIFO là bộ đệm cho quá trình truyền dữ liệu, RX FIFO là bộ đệm cho quá trình đọc dữ liệu, đảm bảo dữ liệu không bị mất do sự bất đồng bộ giữa 2 miền xung clock giữa vi xử lý và giao tiếp I2C. Các khối FIFO



chi tiết và chức năng của các thanh ghi trên được miêu tả qua Bảng 2.

### 3. ĐÁNH GIÁ CHẤT LƯỢNG CỦA THIẾT KẾ

#### 3.1 Tài nguyên logic được sử dụng

*Bảng 1. Chức năng, địa chỉ và kiểu truy cập của các thanh ghi*

PAD DR [7:0]	Kiểu truy cập	Tên thanh ghi	Chức năng
0x1C	Đọc/ Ghi	CONTROL	Thanh ghi điều khiển
0x20	Đọc	STATUS/ENABLE	Thanh ghi trạng thái
0x24	Đọc/ Ghi	ADDRESS	Thanh ghi địa chỉ
0x28	Đọc/ Ghi	COMMAND	Thanh ghi lệnh
0x2C	Đọc/ Ghi	DATA	Thanh ghi dữ liệu
0x30	Đọc/ Ghi	INTERRUPT ENABLE	Thanh ghi cho phép ngắt
0x34	Đọc	RAW INTERRUPT ENABLE	Thanh ghi ngắt
0x38	Đọc	INTERRUPT FLAG REGISTER	Thanh ghi cờ ngắt
0x3c	Đọc/ Ghi	CLOCK SELECT	Thanh ghi chứa giá trị tần số xung clock được chọn
0x40	Đọc/ Ghi	TRANSMIT ADDRESS	Thanh ghi chứa địa chỉ Slave muốn truyền đi

Thiết kế với đầy đủ các chức năng được tổng hợp trên FPGA Spartan 6 XC6SLX9-FT(G)256 của hãng Xilinx [10] có tài nguyên logic được sử dụng được tóm tắt trong Bảng 3. Tần số hoạt động tối đa của mạch được tổng hợp trên FPGA Spartan 6 XC6SLX9-FT(G)256 là 124,245 MHz (xem bảng 4).

Đồng thời thiết kế cũng được tổng hợp trên Quartus II theo cấu hình board Cylone II [11] để so sánh tài nguyên logic cần sử dụng giữa 2 board FPGA (Bảng 5). Tần số tối đa của thiết kế trên board Cylone II là 150.35MHz (Bảng 6).

#### 3.2 Kết quả đánh giá qua mô phỏng

##### 3.2.1 Mô hình mô phỏng và tóm tắt Testcase

Thiết kế được mô phỏng trên QuestaSim 10.2 với mô hình cho 2 module cùng thiết kế nhưng được cấu hình khác nhau (1 Master và 1

Slave) để kiểm tra hoạt động của thiết kế. Mô hình mô phỏng được thể hiện trong Hình 3.

*Bảng 2. Các thanh ghi ứng với địa chỉ PADDR và kiểu truy cập*

Tên Thanh Ghi	Miêu Tả
CONTROL	Ký hiệu: CTR CTR[0]: Xóa nội dung FIFO truyền CTR[1]: Xóa nội dung FIFO nhận CTR[2]: Cấu hình hoạt động thiết bị '0' – Slave , '1' – Master
STATUS/ENABLE	Ký hiệu: SE SE[0]: Cho phép I2C hoạt động SE[1]: TX FIFO chưa đầy SE[2]: RX FIFO chưa trống SE[3]: Khối I2C đang làm việc
ADDRESS	Ký hiệu: ADDR ADDR[6:0]: Địa chỉ của thiết bị khi hoạt động là Slave
COMMAND	Ký hiệu: COM COM[0]: Điều khiển hướng truyền dữ liệu '0' – Master -> Slave '1' – Master <- Slave COM[1]: Khởi tạo/kết thúc giao tiếp '0' – Stop '1' – Start COM[2]: Khởi tạo giao tiếp mới
DATA	Ký hiệu: DATA DATA[7:0]: Dữ liệu truyền từ APB xuống I2C hoặc từ I2C lên APB
INTERRUPT ENABLE	Ký hiệu: IE IE[0]: Cho phép ngắt Overrun IE[1]: Cho phép ngắt truyền IE[2]: Cho phép ngắt nhận
RAW INTERRUPT ENABLE	Ký hiệu: IR IR[0]: Ngắt Overrun IR[1]: Ngắt truyền IR[2]: Ngắt nhận
INTERRUPT FLAG REGISTER	Ký hiệu: IF IF [0]: Cờ ngắt Overrun IF [1]: Cờ ngắt truyền IF [2]: Cờ ngắt nhận
CLOCK SELECT	Ký hiệu: CSEL CSEL[1:0]: Chọn tốc độ giao tiếp I2C 00: 100kbps 01: 400kbps 10: 1Mbps 11: 3.4Mbps
TRANSMIT ADDRESS	Ký hiệu: TADDR TADDR[6:0]: Địa chỉ Slave mà Master cần giao tiếp

**Bảng 3.** Tài nguyên cần sử dụng của thiết kế trên chip FPGA Spartan 6.

Tài nguyên logic được sử dụng (ước tính)			
Tài nguyên logic	Đã dùng	Tài nguyên	Sử dụng
Slice thanh ghi	245	11440	2%
Slice LUT	577	5720	10%
Số cặp LUT-FF được dùng toàn bộ	184	638	28%
IOB được liên kết	32	186	17%
BUFG/BUFGCTRLs	1	16	6%

**Bảng 4.** Tần số tối đa của thiết kế trên chip FPGA Spartan 6.

Tóm lược về định thời	
Thời gian tối thiểu	8.049ns
Tần số tối đa	124.245MHz
Thời gian ngõ vào tối thiểu trước xung clock	9.063ns
Thời gian yêu cầu ngõ ra tối đa sau xung clock	13.341ns

**Bảng 5.** Tài nguyên cần sử dụng của thiết kế trên chip FPGA Cyclone II.

Tài nguyên logic được sử dụng			
Tài nguyên logic	Đã dùng	Tài nguyên	Sử dụng
Thành phần logic	651	4608	14%
Tổ hợp chức năng	615	4608	13%
Thanh ghi	286	4608	6%
Chân	32	89	36%
Số bit nhớ	256	1119808	<1%

**Bảng 6.** Tần số tối đa của thiết kế trên chip FPGA Cyclone II.

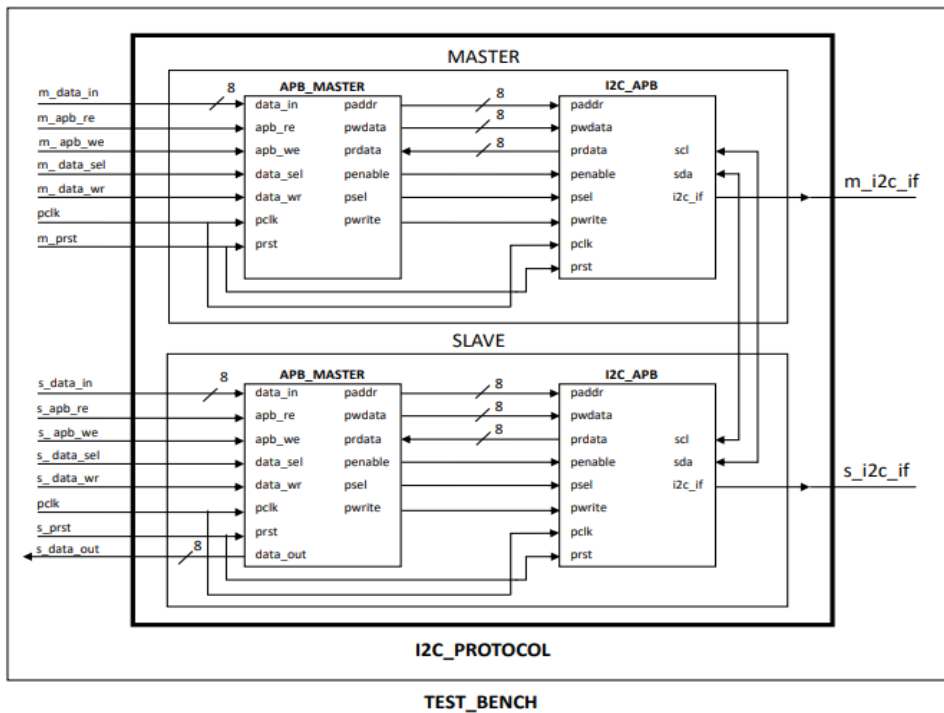
Tóm lược về định thời		
Fmax	Tần số tối đa hạn chế	Tên xung clock
150.35 MHz	150.35 MHz	pclk

Trong các hệ thống số sử dụng AMBA Bus (tương tự như Hình 1), APB Bridge đóng vai trò tương tác với các bus tốc độ cao. Các module ngoại vi muốn giao tiếp được với bus phải có mạch giao tiếp, mạch này

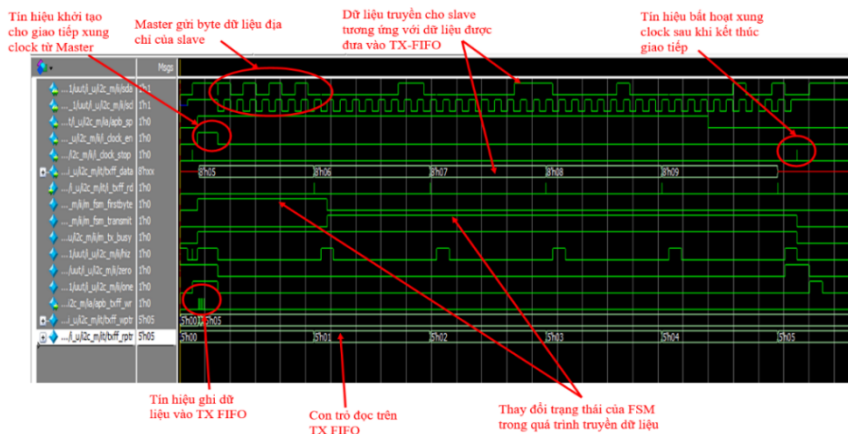
thường còn được gọi APB Slave, mạch điều khiển bus từ phía APB Bridge còn gọi là APB Master. Trong mô hình mô phỏng trên Hình 3, module I2C\_APB chính là module I2C, trong đó module APB Master đóng vai trò như APB Bridge để giao tiếp dựa trên bus APB và được thiết kế dựa trên mô hình trạng thái hữu hạn được cung cấp bởi tài liệu định nghĩa chuẩn AMBA của ARM [7].

**Bảng 7.** Bảng tóm tắt các testcase

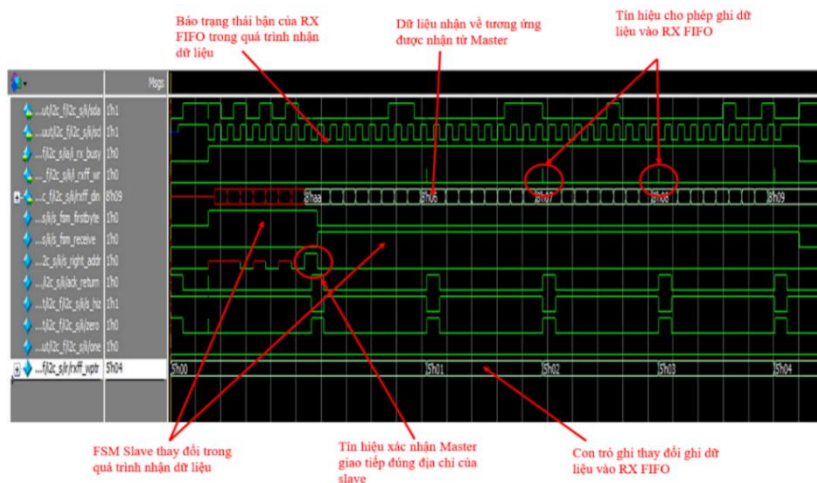
Test case	Cấu hình thanh ghi	
	Master	Slave
1 (Hình 4,5) Master truyền Slave nhận	+ CTR (111): cấu hình Master, xóa FIFO truyền, nhận. + TADDR (1010101): địa chỉ slave cần giao tiếp. + IF (111): cho phép ngắt Truyền, Nhận, Overrun. + CSEL (00): cấu hình tốc độ giao tiếp 100kbps. + COM (010): Phát tín hiệu Start, Master truyền Slave nhận. Kết thúc giao tiếp cấu hình COM (000). + SE: cho phép giao tiếp I2C. + DATA: 8'h06 -> 8'h09	+ CTR (011): cấu hình Slave, xóa FIFO truyền, nhận. + ADDR (1010101): địa chỉ Slave. + IF (111): cho phép ngắt Truyền, Nhận, Overrun. + SE: cho phép giao tiếp I2C.
2 (Hình 6,7) Master nhận Slave truyền	+ CTR (111): cấu hình Master, xóa FIFO truyền, nhận. + TADDR (1010101): địa chỉ slave cần giao tiếp. + IF (111): cho phép ngắt Truyền, Nhận, Overrun. + CSEL (11): cấu hình tốc độ giao tiếp 3,4Mbps. + COM (011): Phát tín hiệu Start, Master nhận Slave truyền. Kết thúc giao tiếp cấu hình COM (001). + SE: cho phép giao tiếp I2C.	+ CTR (011): cấu hình Slave, xóa FIFO truyền, nhận. + ADDR (1010101): địa chỉ Slave. + IF (111): cho phép ngắt Truyền, Nhận, Overrun. + SE: cho phép giao tiếp I2C. + DATA: 8'7f -> 8'h82.



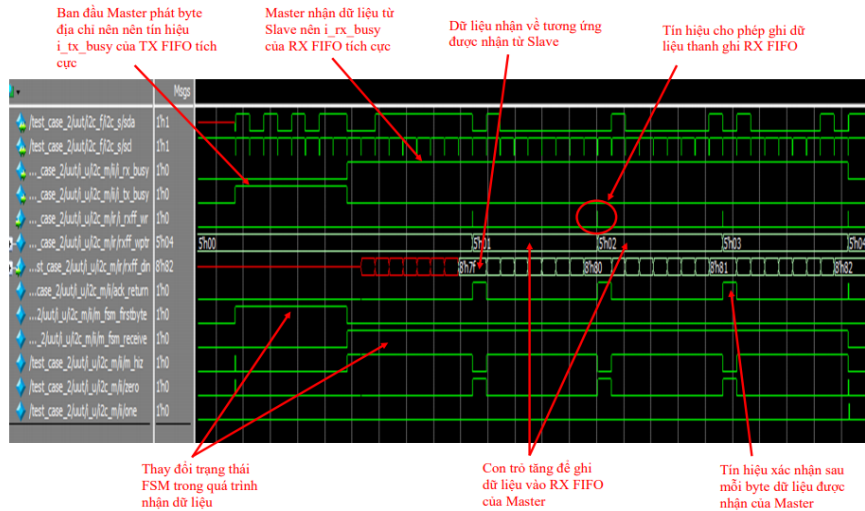
Hình 3. Mô hình mô phỏng kiểm tra thiết kế



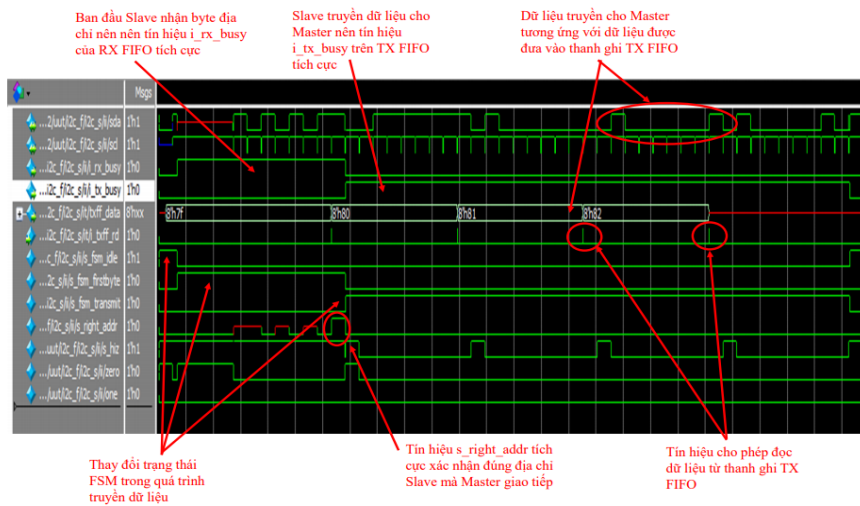
Hình 4. Mô phỏng đường dữ liệu SDA, clock SCL và các tín hiệu nội Master truyền (Testcase 1).



Hình 5. Mô phỏng đường dữ liệu SDA, clock SCL và tín hiệu nội của Slave nhận (Testcase 1).



Hình 6. Mô phỏng đường dữ liệu SDA, clock SCL và các tín hiệu nội Master nhận (Testcase 2).



Hình 7. Mô phỏng đường dữ liệu SDA, clock SCL và tín hiệu nội của Slave truyền (Testcase 2).

Thiết kế được mô phỏng qua 2 Testcase được liệt kê trong Bảng 7, bao hàm 2 tốc độ truyền 100 kbps và 3,4 Mbps. Dữ liệu cấu hình và dữ liệu truyền nhận được đọc và ghi qua các chu kỳ giao tiếp bus APB. Lần lượt các Testcase được thực hiện mô phỏng và kiểm tra. Dựa vào đó để có thể kết luận thiết kế hoạt động tốt trên mô phỏng, đảm bảo việc truyền nhận I2C ở nhiều tốc độ và cấu hình khác nhau, khả năng giao tiếp theo chuẩn APB.

### 3.2.2 Phân tích kết quả mô phỏng

Trong phần này, chúng tôi lần lượt trình bày các kết quả đánh giá qua mô phỏng tương ứng với Testcase 1 và 2.

#### 3.2.2.1 Testcase 1

##### • Master truyền

Hình 4 trình bày kết quả mô phỏng cho trường hợp master truyền. Xung clock Master được tạo và kết thúc bằng tín hiệu i\_clock\_en và i\_clock\_stop. Con trỏ trên TX FIFO txff\_rptr tăng tương ứng đọc dữ liệu từ thanh ghi txff\_data là 8'h06, 8'h07, 8'h08, 8'h09 để gửi dữ liệu. Máy trạng thái FSM bắt đầu quá trình giao tiếp chuyển từ trạng thái IDLE sang trạng thái FIRSTBYTE và gửi byte địa chỉ mà Slave cần giao tiếp, sau khi nhận bit ACK xác nhận truyền đúng địa chỉ, FSM chuyển trạng thái sang TRANSMIT để phát các bit dữ liệu. Các tín hiệu hiz, zero và one quyết định Master hay Slave sẽ lái tín

hiệu trên đường dữ liệu SDA, đảm bảo việc truyền nhận ở cả Master và Slave.

• **Slave nhận**

Hình 5 trình bày dạng sóng mô phỏng ứng với trường hợp slave nhận. Slave ở trạng thái nhận dữ liệu nên bit rx\_busy được tích cực. Sau byte địa chỉ đầu tiên, Slave kiểm tra địa chỉ và xác nhận đúng bật bit s\_right\_add và bắt đầu quá trình truyền nhận. Các dữ liệu nhận trên RX FIFO tương ứng dữ liệu được gửi, con trỏ ghi rxff\_wptr tăng. Sau mỗi byte truyền, khi nhận đủ byte, ack\_return được bật lên kéo đường SDA xuống 0 để xác nhận truyền đủ bit. Đối với FSM phía Slave, bắt đầu giao tiếp chuyển từ trạng thái IDLE sang trạng thái FIRSTBYTE để so sánh byte địa chỉ, nếu chính xác sẽ chuyển sang trạng thái RECEIVE để bắt đầu nhận dữ liệu.

**Bảng 8.** Bảng đánh giá kết quả mô phỏng.

Testcase	Đánh giá
1	- Đảm bảo thiết kế hoạt động chính xác khi được cấu hình là Master truyền hay Slave nhận. - Byte địa chỉ đầu tiên được Master gửi đúng với địa chỉ cấu hình trên Slave (1010101) và bit điều khiển truyền nhận chính xác R/W = 0 (Master gửi dữ liệu). - Các byte dữ liệu được đưa xuống Master được truyền chính xác. - TX FIFO và RX FIFO đảm bảo dữ liệu được truyền đi và nhận về và hoạt động đúng với thiết kế đề ra. - Tốc độ SCL tương ứng là 100 kbps.
2	- Đảm bảo thiết kế hoạt động chính xác khi được cấu hình là Master nhận hay Slave truyền. - Byte địa chỉ đầu tiên được Master gửi đúng với địa chỉ cấu hình trên Slave (1010101) và bit điều khiển truyền nhận chính xác R/W = 1 (Master nhận dữ liệu). - Các byte dữ liệu được đưa xuống Slave được truyền chính xác. - TX FIFO và RX FIFO đảm bảo dữ liệu được truyền đi và nhận về và hoạt động đúng với thiết kế đề ra. - Tốc độ SCL tương ứng là 3,4 Mbps.

3.2.2.2 Testcase 2

• **Master nhận**

Hình 6 trình bày dạng sóng mô phỏng cho trường hợp master nhận. Xung clock được tạo từ Master tương tự như Testcase 1. Khi Master ở trạng thái Receive, ban đầu sẽ phát byte địa chỉ giao tiếp với Slave và nhận dữ liệu từ Slave do đó tín hiệu i\_rx\_busy được tích cực. Sau mỗi 8 bit, tín hiệu i\_rxff\_wr được bật đồng thời tăng giá trị con trỏ ghi rxff\_wptr để ghi dữ liệu vào thanh ghi RX FIFO. Dữ liệu nhận tương ứng là 8'h7f, 8'h80, 8'h81 và 8'h82. Đồng thời cuối mỗi byte dữ liệu nhận, tín hiệu ACK tích cực được báo về cho Slave.

• **Slave Transmit**

Hình 7 trình bày kết quả mô phỏng cho trường hợp slave truyền. Sau khi nhận đúng địa chỉ gửi từ Master, FSM phía Slave chuyển từ trạng thái IDLE sang trạng thái TRANSMIT để gửi dữ liệu, dữ liệu truyền đi lần lượt là 8'h7f, 8'h80, 8'h81, 8'h82. Tín hiệu i\_txff\_rd được bật để tăng con trỏ i\_txff\_wptr để đọc dữ liệu từ TX FIFO và gửi dữ liệu.

Sau khi so sánh kết quả mô phỏng so với lý thuyết và các yêu cầu thiết kế, chúng tôi nhận thấy rằng module I2C được thiết kế hoạt động chính xác, đáp ứng được yêu cầu thiết kế với nhiều cấu hình khác nhau (như được tóm tắt trong bảng 8).

**4. KẾT LUẬN**

Thiết kế module I2C được trình bày trong bài báo này sử dụng chuẩn bus AMBA APB. Thiết kế hỗ trợ nhiều cấu hình và tốc độ truyền nhận. Bên cạnh đó, thiết kế có thể được sử dụng như một soft IP core ở quá trình thiết kế Front-End khi thiết kế IC. Ngoài ra, trong khi hầu hết các thiết kế vi mạch ít được trình bày chi tiết, bài báo này có giá trị như một tài liệu tham khảo cho việc học tập các môn liên quan tới thiết kế hệ thống số và thiết kế vi mạch số.



## TÀI LIỆU THAM KHẢO

- [1] Pong P. Chu. RTL Hardware Design using VHDL. A JOHN WILEY & SONS, INC, 2006.
- [2] Stephen Brown and Zvonko Vranesic. Fundamentals of Digital Logic with Verilog Design (third edition). McGraw Hill, 2014.
- [3] D. Trivedi, A. Khade, K. Jain and R. Jadhav, "SPI to I2C Protocol Conversion Using Verilog," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-4.
- [4] A. Anagha and M. Mathurakani, "Prototyping of dual master I2C bus controller," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, 2016, pp. 2124-2129.
- [5] A. Anagha and M. Mathurakani, "Prototyping of dual master I2C bus controller," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 2016, pp. 2124-2129.
- [6] P. Bagdalkar and L. Ali, "Hardware Implementation of I2C Controller on FPGA and Validation Through Interfacing with Low-Cost ADC," 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2020, pp. 887-891.
- [7] C. Liu, Q. Meng, T. Liao, X. Bao and C. Xu, "A Flexible Hardware Architecture for Slave Device of I2C Bus," 2019 International Conference on Electronic Engineering and Informatics (EEI), Nanjing, China, 2019, pp. 309-313.
- [8] V. Patel K.S and B. R, "Design and Verification of Wishbone I2C Master Device," 2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS), Bangalore, India, 2018, pp. 1-5.
- [9] AMBA™ Specification (Rev 2.0), [developer.arm.com/docs/ihi0011/a/amba-specification-rev-20](http://developer.arm.com/docs/ihi0011/a/amba-specification-rev-20).
- [10] Wael M. El-Medany. Reconfigurable CRC IP core design on xilinx spartan 3AN FPGA. International Journal of Computer Applications in Technology (IJCAT), 55(4), 2017.
- [11] V. Venkataramanan; S. Lakshmi; Vineet A. Kanetkar. Design and implementation of LTE physical layer on FPGA. International Journal of Computer Applications in Technology (IJCAT), 61(1), 2019.

**Tác giả chịu trách nhiệm bài viết:**

TS. Đỗ Duy Tân

Trường Đại học Sư phạm Kỹ thuật TP.HCM

Email: tandd@hcmute.edu.vn