

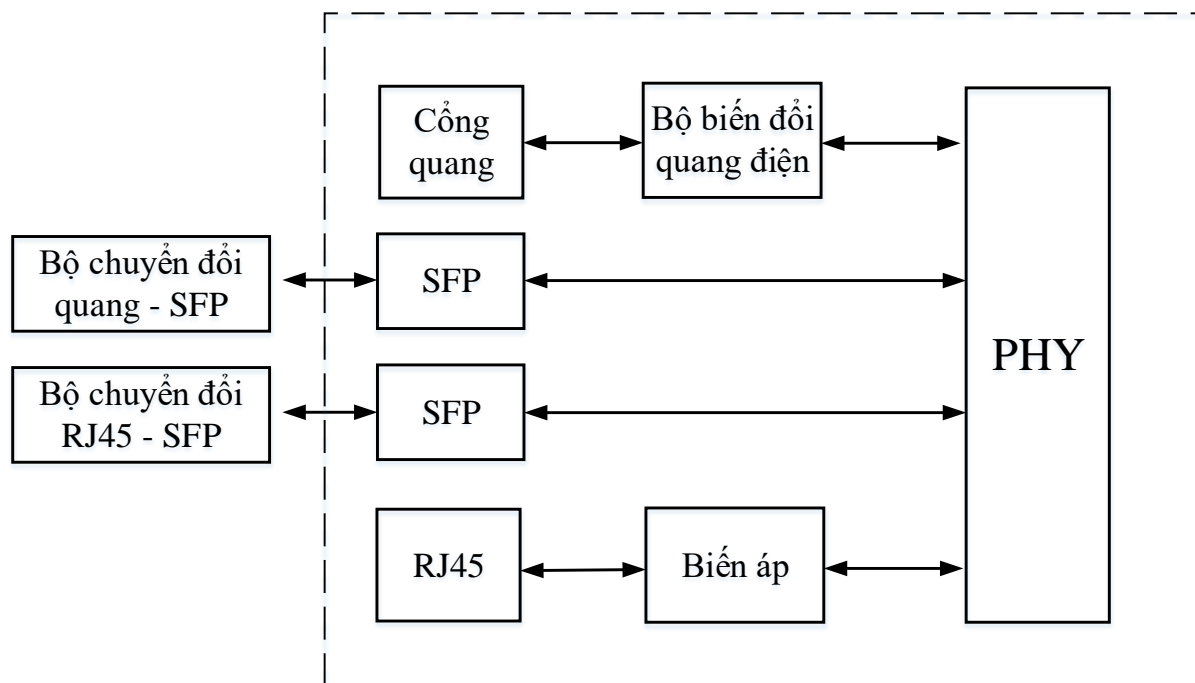
CHƯƠNG 3

THIẾT KẾ GIAO TIẾP NGOẠI VI

Chương trước đã trình bày các khái niệm, định nghĩa của thiết bị chuyển mạch lớp 2 đồng thời tính toán các yêu cầu kỹ thuật để có thể lựa chọn công nghệ thiết kế phù hợp. Chương này sẽ tập trung trình bày về phần giao tiếp ngoại vi của thiết kế switch layer 2, cụ thể bao gồm trình bày về các giao diện vật lý, chi tiết thiết kế mạch in và các yêu cầu kỹ thuật căn bản. Chương này cũng trình bày về các giao tiếp chuẩn giữa chip vật lý và phần FPGA cũng như các khối thiết kế số có nhiệm vụ truyền nhận dữ liệu giữa lõi chuyển mạch và các đường truyền vật lý, còn gọi là các khối điều khiển truy cập đường truyền (Media Access Controller – MAC). Việc kiểm tra mạch, kiểm tra các cổng kết nối dựa trên TEMAC IP Core (Tri-Mode Ethernet MAC Intellectual Property Core) cũng được trình bày ở chương này.

3.1 Giao tiếp giữa cổng Ethernet và chip vật lý

Hình 3.1 là mô hình điển hình giao tiếp từ đường truyền vật lý MDI (Media Dependent Interface) tới chip giao tiếp Ethernet PHY (Physical Ethernet IC).



Hình 3.1 Mô hình giao tiếp giữa cổng Ethernet và chip vật lý

Trong đó:

RJ45: Cổng kết nối cáp xoắn đôi.

*SFP*¹: Viết tắt của Small Form Factor pluggable transceiver, có nghĩa là cổng truyền nhận đa năng kích thước nhỏ, hỗ trợ tất cả các kết nối thông qua bộ chuyển đổi riêng biệt như quang – SFP hoặc RJ45 – SFP.

Cổng quang: Cổng kết nối cáp quang.

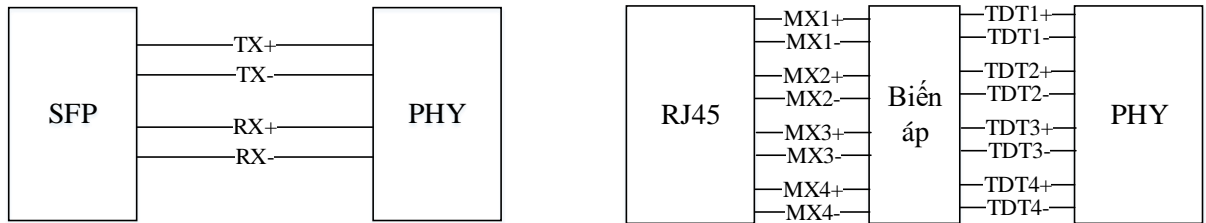
Biến áp: Cách ly các đặc tính điện một chiều giữa dây dẫn và mạch điện, biến đổi tín hiệu phù hợp với đầu vào chip PHY.

Bộ biến đổi quang điện: Biến đổi các tín hiệu quang thành tín hiệu điện để đưa vào mạch điện.

Chip PHY: Chip vật lý biến đổi tín hiệu từ môi trường vật lý lên lớp cao hơn.

¹ Trong đề án này sử dụng thuật ngữ SFP thay cho cổng truyền nhận đa năng.

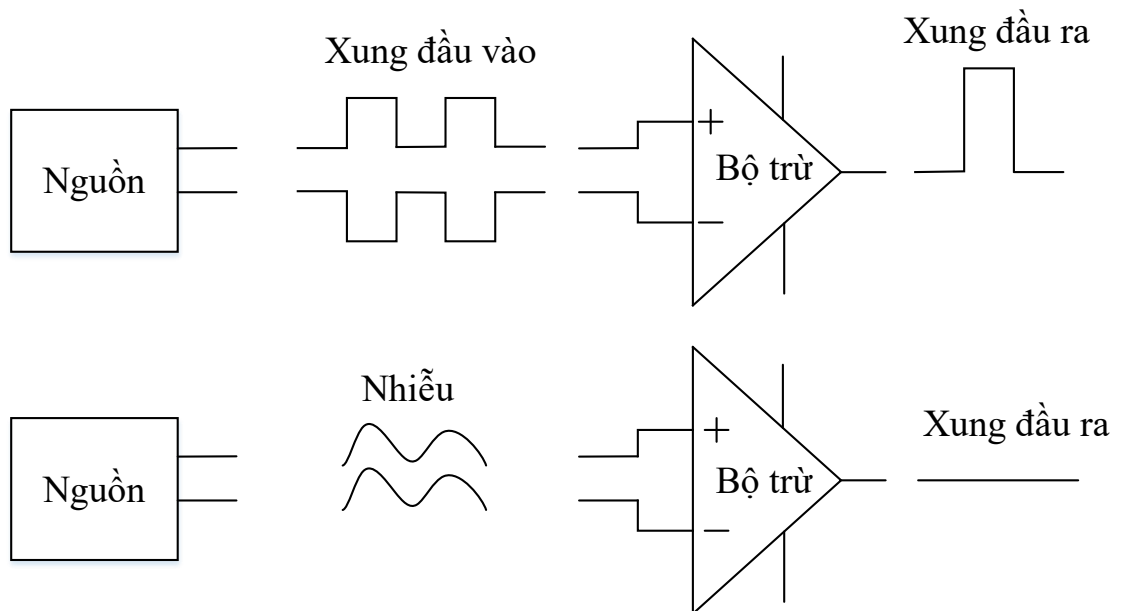
Các giao diện kết nối giữa cổng vật lý đến chip vật lý được gọi là các giao diện phụ thuộc vào môi trường MDI, thường sử dụng các cặp vi sai để truyền dữ liệu (Hình 3.2).



a) Giao diện kết nối giữa SFP đến chip vật lý b) Giao diện kết nối giữa RJ45 đến chip vật lý

Hình 3.2 Giao diện kết nối giữa cổng vật lý đến chip vật lý

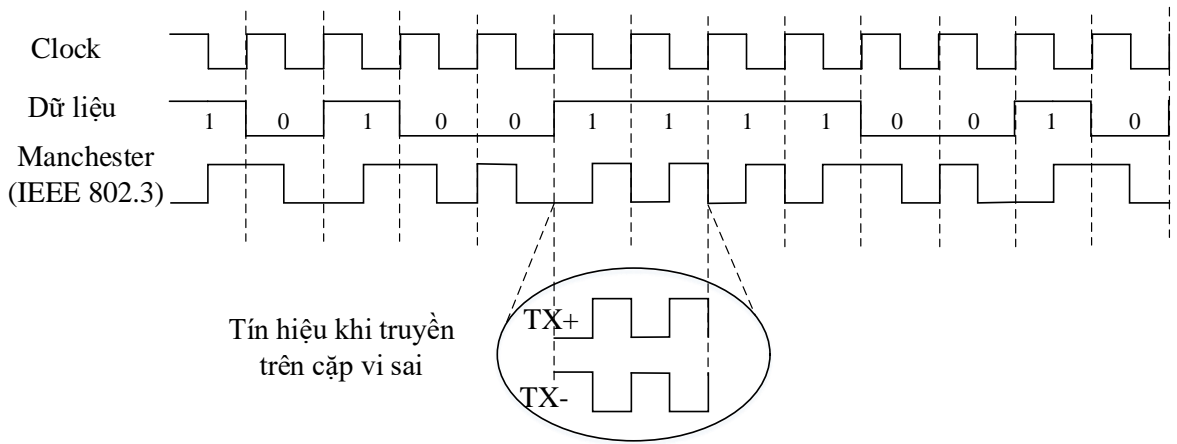
Tín hiệu vi sai sử dụng hai tín hiệu bổ sung, kỹ thuật này gửi cùng một tín hiệu trên cùng một cặp đường truyền, chênh lệch điện áp giữa cặp dây sẽ đưa ra mức logic của tín hiệu cần truyền đi. Ưu điểm khi sử dụng cặp vi sai là có thể truyền được xa hơn và chống nhiễu tốt. Khi có nhiễu sẽ ảnh hưởng đến cả hai dây trong cặp, mà phía nhận tín hiệu chỉ phát hiện sự chênh lệch điện áp giữa hai tín hiệu (Hình 3.3).



Hình 3.3 Đặc điểm của cặp tín hiệu vi sai khi truyền nhận

Ví dụ đối với cáp xoắn đồng ở chế độ 10 Mbps thì sử dụng mã Manchester. Mã Manchester theo chuẩn IEEE 802.3 mã hoá bit 0 thành tín hiệu có mức điện áp chuyển từ cao xuống thấp ở giữa bit, mã hoá bit 1 thành tín hiệu có mức điện áp

chuyển từ thấp lên cao [16]. Hình 3.4 ví dụ về mã hoá Manchester và cách truyền trên cặp dây xoắn.



Hình 3.4 Mã hoá Manchester

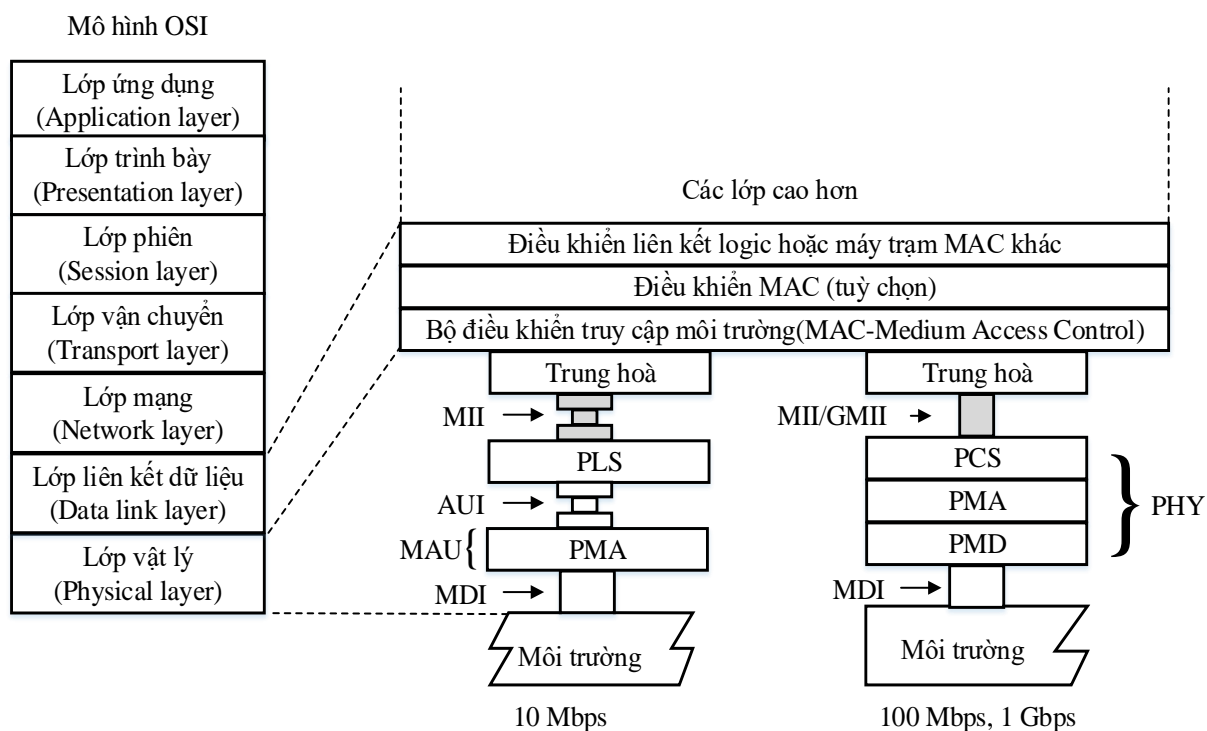
Ví dụ về cổng vật lý RJ45 sử dụng các chip vật lý như: RTL8211E/RTL8211EG [26] được tích hợp, hỗ trợ ba tốc độ 10/100/1000 Mbps. AR8031 [9], AR8035 [10], AR8033 [25] cũng hỗ trợ ba tốc độ 10/100/1000 Mbps. Bên cạnh các chip vật lý hỗ trợ cho đơn cổng ra thì còn có các chip vật lý hỗ trợ nhiều đầu ra như RTL8218B [27], 88E1111 [19]...

3.2 Giao tiếp giữa chip vật lý và FPGA

Do có nhiều phương tiện truyền dẫn Ethernet như cáp đồng trục, cáp xoắn đôi đồng, cáp quang vì vậy cần có một chuẩn giao tiếp đồng nhất với mọi phương tiện truyền dẫn để có thể dễ dàng thay đổi giao diện vật lý mà không cần thay đổi phần cứng MAC. Do đó MAC có thể sử dụng với bất kỳ giao diện lớp vật lý nào mà không phụ thuộc vào phương tiện truyền dẫn. Các giao diện giao tiếp giữa chip vật lý và FPGA gồm có giao diện MII, RMII, GMII, RGMII, SGMII.

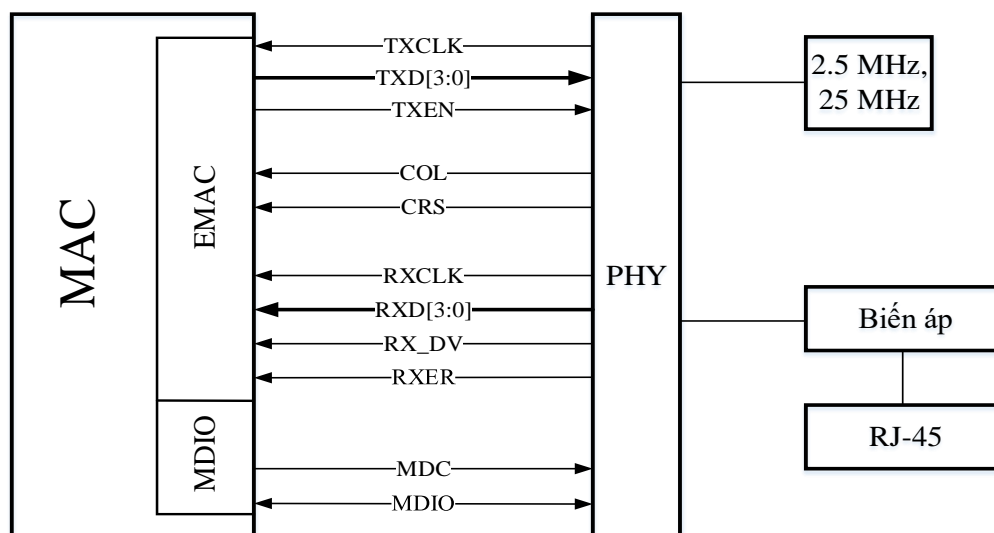
3.2.1 Giao diện MII

Giao diện MII giữa các bộ điều khiển truy cập phương tiện truyền dẫn (MAC) và các lớp vật lý (PHY) là khác nhau. Giao diện MII nằm giữa lớp vật lý và lớp liên kết dữ liệu đóng vai trò trung gian giữa giao diện người dùng và giao diện vật lý, không phụ thuộc vào phương tiện truyền dẫn (Hình 3.5). Mục đích của giao diện này là cung cấp một kết nối đơn giản, dễ thực hiện giữa MAC và PHY, hỗ trợ tốc độ truyền lên tới 100 Mbps.



Hình 3.5 Vị trí của MII trong mô hình OSI [18]

Các tín hiệu kết nối giữa PHY và MAC bằng giao thức MII được mô tả như Hình 3.6.



Hình 3.6 Sơ đồ tín hiệu sử dụng giao diện MII giữa PHY và MAC

Trong đó:

PHY: Chip vật lý kết nối với cổng Ethernet cụ thể trong hình là RJ45 thông qua biến áp. Hoạt động với xung nhịp tần số 2.5 MHz hoặc 25 MHz.

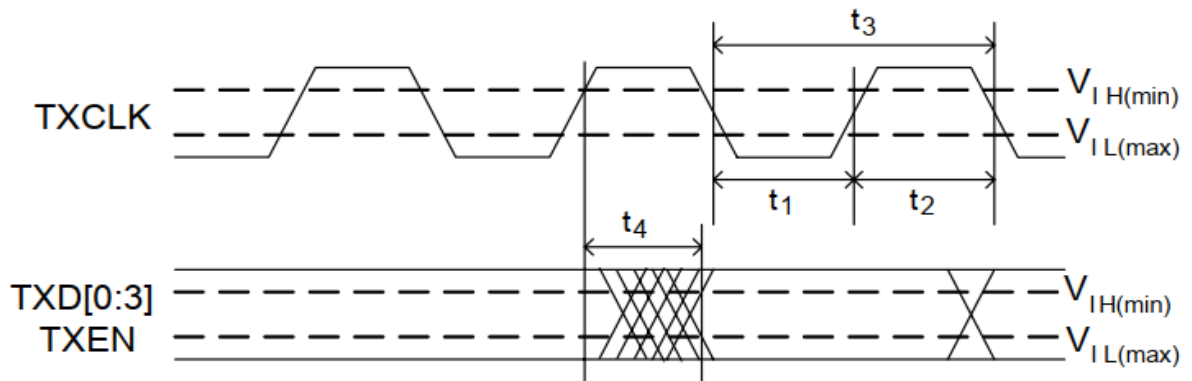
MAC: Bộ xử lý tín hiệu người dùng bao gồm giao diện điều khiển MDIO và giao diện biến đổi MII – MAC.

Tín hiệu phía truyền: Gồm tín hiệu đồng bộ (TXCLK), bốn tín hiệu dữ liệu (TXD), tín hiệu cho phép truyền (TXEN).

Tín hiệu phía nhận: Gồm tín hiệu đồng bộ (RXCLK), bốn tín hiệu dữ liệu (RXD), tín hiệu thông báo có dữ liệu (RXDV), tín hiệu thông báo lỗi (RXER), tín hiệu phát hiện sóng mang (CRS), tín hiệu phát hiện xung đột (COL).

Giao diện điều khiển sử dụng MDIO: bao gồm tín hiệu đồng bộ MDC và tín hiệu dữ liệu MDIO.

Dưới đây là đặc tính thời gian quá trình truyền gói tin từ MAC đến PHY của giao diện MII (Hình 3.7).



Hình 3.7 Đặc tính thời gian của chu kỳ truyền dữ liệu MII [26]

Trong đó:

TXCLK : Tín hiệu đồng bộ cho việc truyền dữ liệu

TXD[0:3] : 4 tín hiệu dữ liệu

TXEN : Tín hiệu thông báo truyền dữ liệu

$V_{IH(min)}$: Mức điện áp cao tối thiểu [V]

$V_{IL(max)}$: Mức điện áp thấp tối thiểu [V]

t_1, t_2 : Thời gian tín hiệu đồng bộ ở mức thấp và mức cao [ns]

t_3 : Chu kỳ của tín hiệu đồng bộ [ns]

t_4 : Thời gian chuyển mức tín hiệu [ns]

Đối với MII thì tín hiệu điều khiển TXEN và các tín hiệu dữ liệu TXD[0:3] được đồng bộ tại sườn lên của tín hiệu TXCLK. Do đó 1 chu kỳ TXCLK truyền

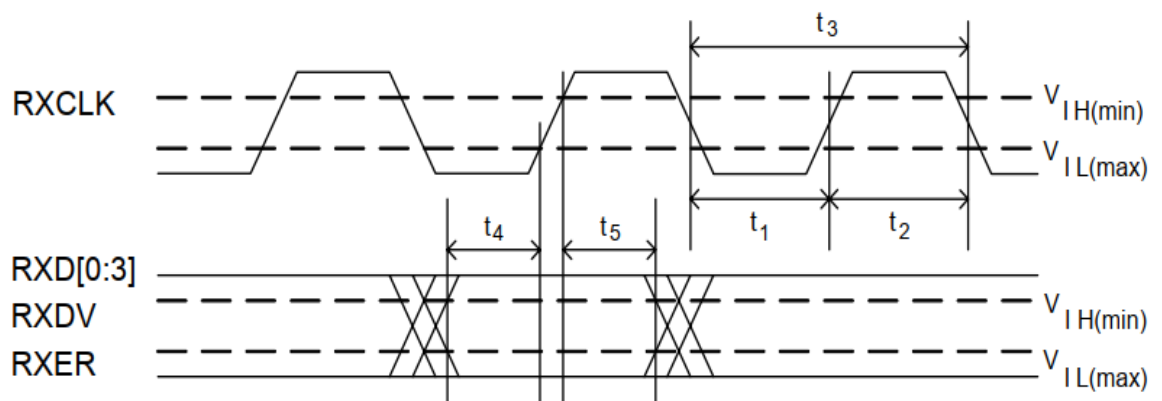
được 4 bits dữ liệu. Chính vì vậy đối với TXCLK có tần số 2.5 MHz thì tốc độ truyền của MII là 10 Mbps, đối với TXCLK có tần số 25 MHz thì tốc độ truyền của MII là 100 Mbps.

Mỗi loại chip PHY khác nhau có thông số t_1 , t_2 , t_3 , t_4 khác nhau với độ chính xác phụ thuộc vào công nghệ làm chip. Ví dụ đối với chip RTL8211EG có các tham số được trình bày ở Bảng 3.1 [26].

Bảng 3.1 Các tham số thời gian truyền MII của chip RTL8211E

Tham số	Tốc độ truyền	Giá trị tối thiểu	Giá trị thông thường	Giá trị tối đa	Đơn vị
t_1 , t_2	100 Mbps	40	50	60	%
	10 Mbps	40	50	60	%
t_3	100 Mbps	-	40	-	ns
	10 Mbps	-	400	-	ns
t_4	100 Mbps	0	-	25	ns
	10 Mbps	0	-	25	ns

Đối với quá trình nhận dữ liệu từ PHY đến MAC thì các tín hiệu có đặc tính như Hình 3.8.



Hình 3.8 Đặc tính thời gian của chu kỳ nhận dữ liệu MII [26]

Trong đó:

RXCLK : Tín hiệu đồng bộ cho việc nhận dữ liệu

RXD[0:3] : 4 tín hiệu dữ liệu

RXDV	: Tín hiệu thông báo có dữ liệu
RXER	: Tín hiệu thông báo có lỗi
$V_{IH(min)}$: Mức điện áp cao tối thiểu [V]
$V_{IL(max)}$: Mức điện áp thấp tối thiểu [V]
t_1, t_2	: Thời gian tín hiệu đồng bộ ở mức thấp và mức cao [ns]
t_3	: Chu kỳ của tín hiệu đồng bộ [ns]
t_4	: Thời gian tối thiểu để tín hiệu thay đổi trước khi có sườn lên [ns]
t_5	: Thời gian tối thiểu tín hiệu phải duy trì mức điện áp sau khi có sườn lên [ns]

Đối với quá trình nhận dữ liệu từ PHY đến MAC thì tại sườn lên của RXCLK, dữ liệu nếu thay đổi phải thay đổi trước sườn lên RXCLK một khoảng t_4 và phải duy trì mức điện áp đó sau sườn lên RXCLK một khoảng khoảng thời gian t_5 . Nhờ đó tại sườn lên RXCLK dữ liệu đã ở mức điện áp ổn định.

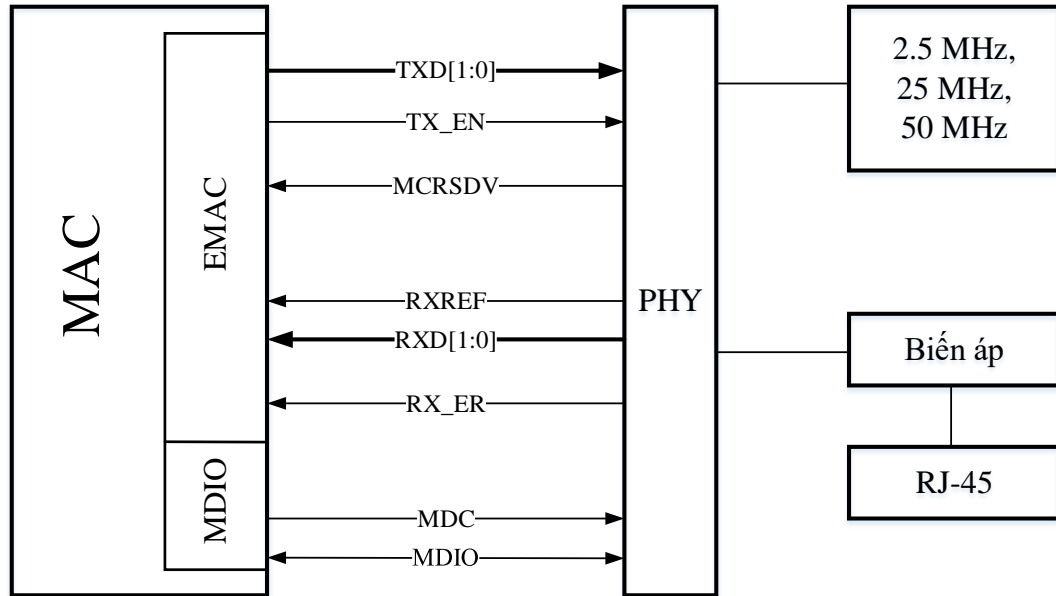
Đối với chip vật lý RTL8211EG có các tham số t_1, t_2, t_3, t_4, t_5 được trình bày ở Bảng 3.2 [26].

Bảng 3.2 Các tham số thời gian nhận MII của chip RTL8211E

Tham số	Tốc độ truyền	Giá trị tối thiểu	Giá trị thông thường	Giá trị tối đa	Đơn vị
t_1, t_2	100 Mbps	40	50	60	%
	10 Mbps	40	50	60	%
t_3	100 Mbps	-	40	-	ns
	10 Mbps	-	400	-	ns
t_4	100 Mbps	10	-	-	ns
	10 Mbps	10	-	-	ns
t_5	100 Mbps	10	-	--	ns
	10 Mbps	10	-	-	ns

3.2.2 Giao diện RMII

Giao diện RMII là một tiêu chuẩn được phát triển dựa trên MII để giảm số lượng tín hiệu của MII. Các tín hiệu kết nối giữa PHY và MAC bằng giao thức RMII được mô tả như Hình 3.9.

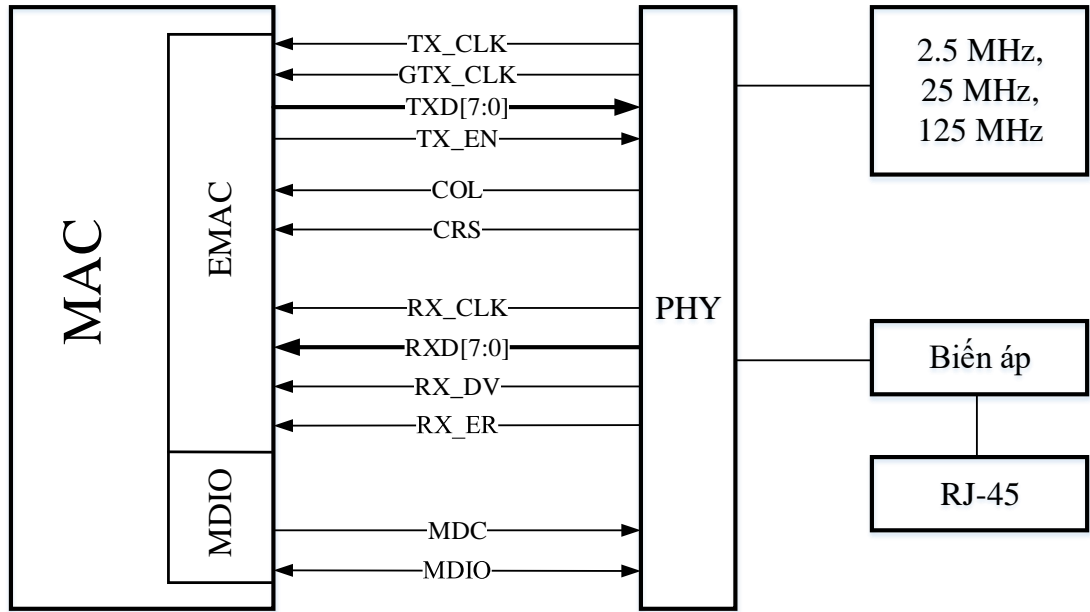


Hình 3.9 Sơ đồ tín hiệu sử dụng giao diện RMII giữa PHY và MAC

Giao diện này là chuẩn giao diện được phát triển để rút gọn số tín hiệu MII. Thay vì sử dụng bốn chân tín hiệu mỗi bên truyền và nhận thì RMII chỉ sử dụng hai tín hiệu mỗi bên. Thêm nữa RMII gộp chung hai tín hiệu RXCLK và TXCLK thành một tín hiệu REFCLK. Do số tín hiệu giảm từ 4 bits về 2 bits nên để tốc độ đạt được như MII thì tần số REFCLK bằng 2 lần so với RXCLK và TXCLK và bằng 50 MHz. Về phần giao diện quản lý giống với MII.

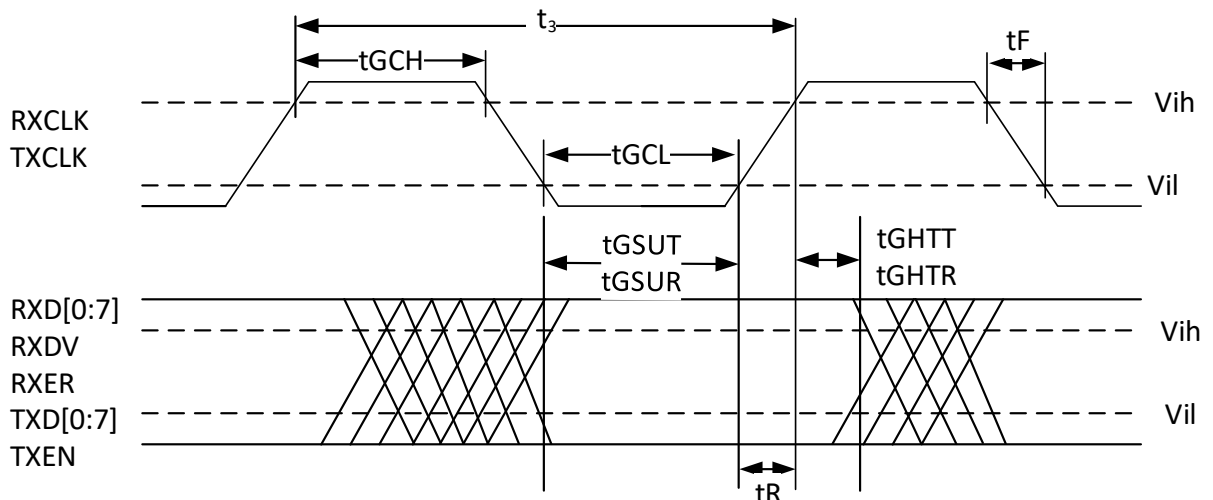
3.2.3 Giao diện GMII

Giao diện MII tốc độ Gigabit (Gigabit media-independent interface), giống như MII, nằm giữa lớp vật lý và lớp liên kết dữ liệu, đóng vai trò trung gian giữa giao diện người dùng và giao diện vật lý, không phụ thuộc vào phương tiện truyền dẫn nhưng khác với MII, GMII hỗ trợ tốc độ lên đến 1000 Mbps. Các tín hiệu kết nối giữa PHY và MAC bằng giao thức GMII được mô tả như Hình 3.10.



Hình 3.10 Sơ đồ tín hiệu sử dụng giao diện GMII giữa PHY và MAC

So với MII thì GMII sử dụng 8 tín hiệu dữ liệu, nên để đạt được tốc độ 1000 Mbps thì GMII sử dụng xung nhịp có tần số 125 MHz là GTXCLK đối với phía truyền. Thêm nữa GMII có thể tương thích với tốc độ 10/100 Mbps thông qua tín hiệu TX_CLK với tần số giống như MII. Đối với RXCLK khi hoạt động ở chế độ 1000 Mbps thì RXCLK có tần số 125 MHz, đối với chế độ 100 Mbps thì RXCLK có tần số 25 Mbps và đối với chế độ 10 Mbps thì RXCLK có tần số 2.5 Mbps. Khi hoạt động ở chế độ 10/100 Mbps chỉ sử dụng bốn tín hiệu dữ liệu, giống như MII. Dưới đây là đặc tính thời gian quá trình truyền nhận gói tin từ MAC đến PHY và ngược lại của giao diện GMII (Hình 3.11).



Hình 3.11 Đặc tính thời gian quá trình truyền nhận gói tin của GMII [26]

Trong đó:

TXCLK	: Tín hiệu đồng bộ cho việc truyền dữ liệu
RXCLK	: Tín hiệu đồng bộ cho việc nhận dữ liệu
TXD[0:7]	: 8 tín hiệu dữ liệu truyền
RXD[0:7]	: 8 tín hiệu dữ liệu nhận
RXDV	: Tín hiệu thông báo có dữ liệu
RXER	: Tín hiệu thông báo có lỗi
TXEN	: Tín hiệu thông báo truyền dữ liệu
tR	: Thời gian chuyển từ mức thấp lên mức cao [ns]
tF	: Thời gian chuyển từ mức cao xuống mức thấp [ns]
Vih	: Mức điện áp cao tối thiểu [V]
Vil	: Mức điện áp thấp tối thiểu [V]
tGCH	: Thời gian tín hiệu đồng bộ ở mức cao [ns]
tGCL	: Thời gian tín hiệu đồng bộ ở mức thấp [ns]
tGSUT	: Thời gian [ns]
tGHTT	: Thời gian [ns]
tGSUR	: Thời gian [ns]
tGSTR	: Thời gian [ns]
t ₄	: Thời gian tối thiểu để tín hiệu thay đổi trước khi có sườn lên [ns]
t ₅	: Thời gian tối thiểu tín hiệu phải duy trì mức điện áp sau khi có sườn lên [ns]

RXCLK và TXCLK có tần số là 125 MHz, một chu kỳ truyền được 8 bits dữ liệu, chính vì vậy tốc truyền của GMII đạt được 1000 Gbps. Do sử dụng tín hiệu đồng bộ có tần số cao nên trong quá trình truyền nhận đều phải ổn định tín hiệu giống như bên nhận của giao diện MII.

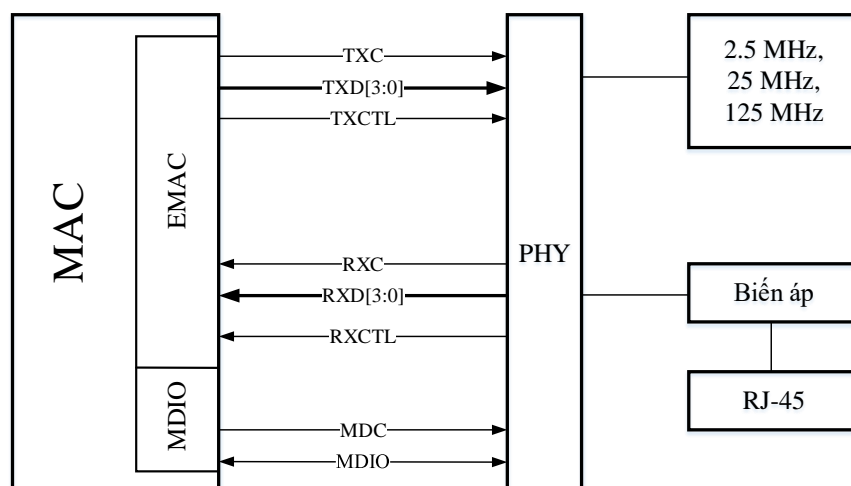
Đối với chip vật lý RTL8211EG có các tham số Tg_{cc}, Tg_{ch}, Tg_{cl}, Tr, Tf, Tg_{sut}, Tg_{htt}, Tg_{sur}, Tg_{htr} được trình bày ở Bảng 3.3 [21].

Bảng 3.3 Các tham số thời gian truyền nhận GMII của chip RTL8211E

Tham số	Giá trị tối thiểu	Giá trị thông thường	Giá trị tối đa	Đơn vị
Tgcc	7.5	8	8.5	ns
Tgch	2.5	-	-	ns
Tgcl	2.5	-	-	ns
Tr	-	-	1	ns
Tf	-	-	1	ns
Tgsut	2.5	-	-	ns
Tghtt	0.5	-	-	ns
Tgsur	2	-	-	ns
Tghtr	0	-	-	ns

3.2.4 Giao diện RGMII

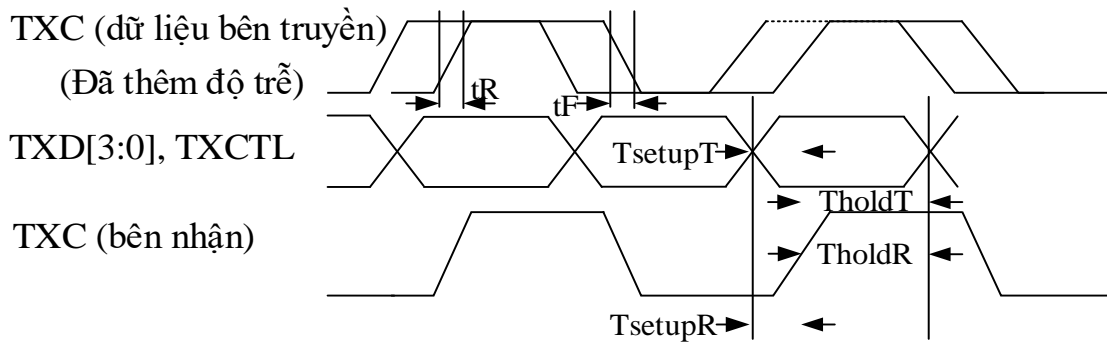
Giao diện RGMII (Reduced Gigabit Media- Independent Interface – MII) là giao diện thay thế cho GMII. Giao diện RGMII giảm một nửa số chân so với giao diện GMII. Với lý do này khi thiết kế nhiều cổng thì việc dùng RGMII là lựa chọn tối ưu. Không có sự thay đổi hoạt động nào của lõi khi có sự yêu cầu lựa chọn giữa GMII và RGMII. Các tín hiệu kết nối giữa PHY và MAC bằng giao thức RGMII được mô tả như Hình 3.12.



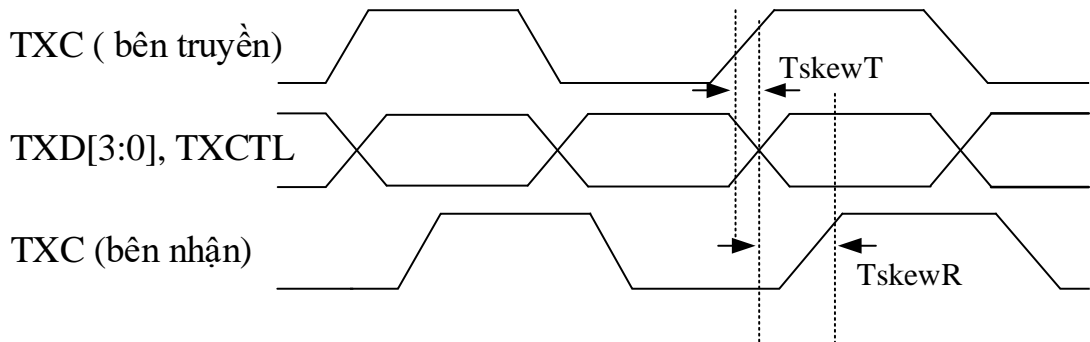
Hình 3.12 Sơ đồ tín hiệu sử dụng giao diện RGMII giữa PHY và MAC

Giao diện RGMII sử dụng 4 tín hiệu dữ liệu thay vì 8 tín hiệu như GMII. Có thể thấy giao diện RGMII giống với MII và thực tế là RGMII có thể hỗ trợ MII khi sử dụng RXC và TXC có tần số 2.5 MHz và 25 MHz. RGMII hỗ trợ tốc độ truyền 1000 Mbps mà chỉ sử dụng 4 chân dữ liệu, do đó chu kỳ truyền dữ liệu RGMII gấp đôi so với GMII. Tuy nhiên RGMII sử dụng tín hiệu đồng bộ 125 MHz thay vì 250 MHz cho nên biện pháp kỹ thuật được sử dụng là truyền dữ liệu theo cả hai sườn của tín hiệu đồng bộ. Đây là thời gian cực kỳ ngắn và dễ rơi vào khoảng bất ổn định khi tín hiệu chuyển đổi mức điện áp. Vậy nên phương pháp được sử dụng làm trễ tín hiệu đồng bộ để thời điểm lấy mẫu vào lúc đường dữ liệu đã ổn định mức điện áp nhờ đó làm giảm lỗi do nhận sai mức điện áp.

Có hai phương pháp làm trễ TXC đó là phía truyền làm trễ và phía nhận làm trễ. Phía truyền làm trễ có nghĩa là TXC được làm trễ ở phía MAC trước khi đưa lên đường truyền (Hình 3.13). Phía nhận làm trễ có nghĩa TXC được làm trễ ở bên PHY, còn trên đường truyền thì TXD và TXCTL vẫn được đồng bộ với sườn lên của TXC (Hình 3.14) [21].

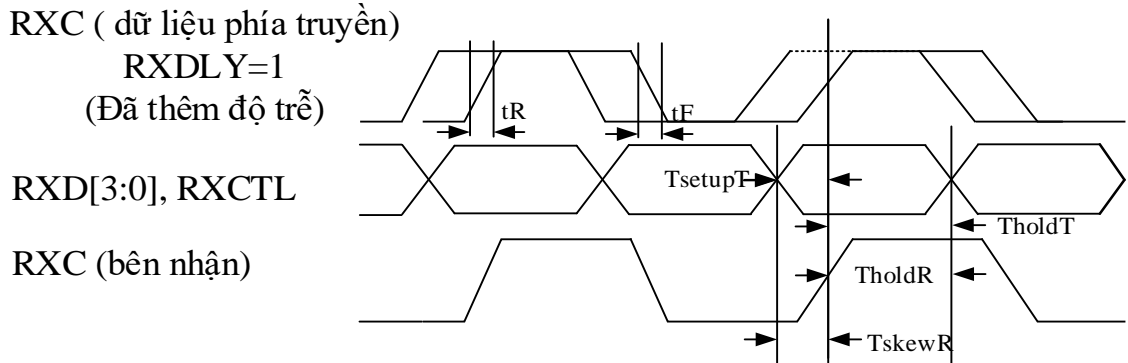


Hình 3.13 Tín hiệu RGMII với TXC được làm trễ bên phía truyền

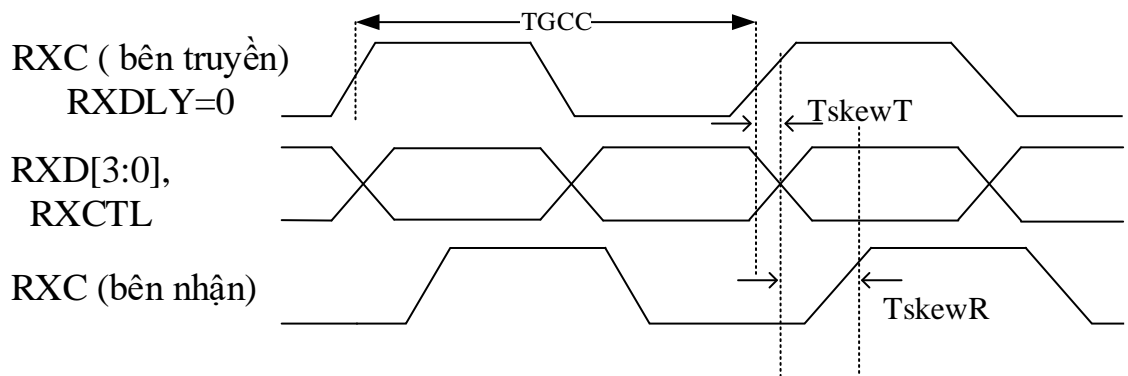


Hình 3.14 Tín hiệu RGMII với TXC không được làm trễ bên phía truyền

Tương tự đối với các tín hiệu RX thì cũng có hai phương pháp làm trễ là MAC làm trễ (Hình 3.15) và PHY làm trễ (Hình 3.16) [21].



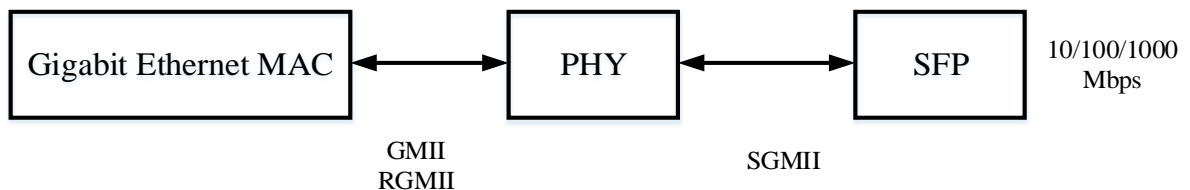
Hình 3.15 Tín hiệu RGMII với RXC được làm trễ bên phía nhận



Hình 3.16 Tín hiệu RGMII với TXC không được làm trễ bên phía nhận

3.2.5 Giao diện SGMII

Giao diện SGMII là một biến thể của giao diện MII. Nó được sử dụng cho Gigabit Ethernet nhưng cũng có thể sử dụng cho Ethernet 10/100 Mbps. SGMII chỉ sử dụng một cặp tín hiệu dữ liệu vi sai cho mỗi bên là TX+, TX-, RX+, RX-. PHY có chức năng chuyển đổi SGMII thành GMII/RGMII (Hình 3.17). Các tín hiệu đó được PHY và SFP đồng bộ với tín hiệu vi sai tần số 625 MHz và truyền dữ liệu ở cả hai sườn. Do đó tốc độ truyền là 1250 Mbps và sử dụng mã hoá 8b/10b cho nên tốc độ truyền bit là $1250 \times 8/10 = 1000$ Mbps.

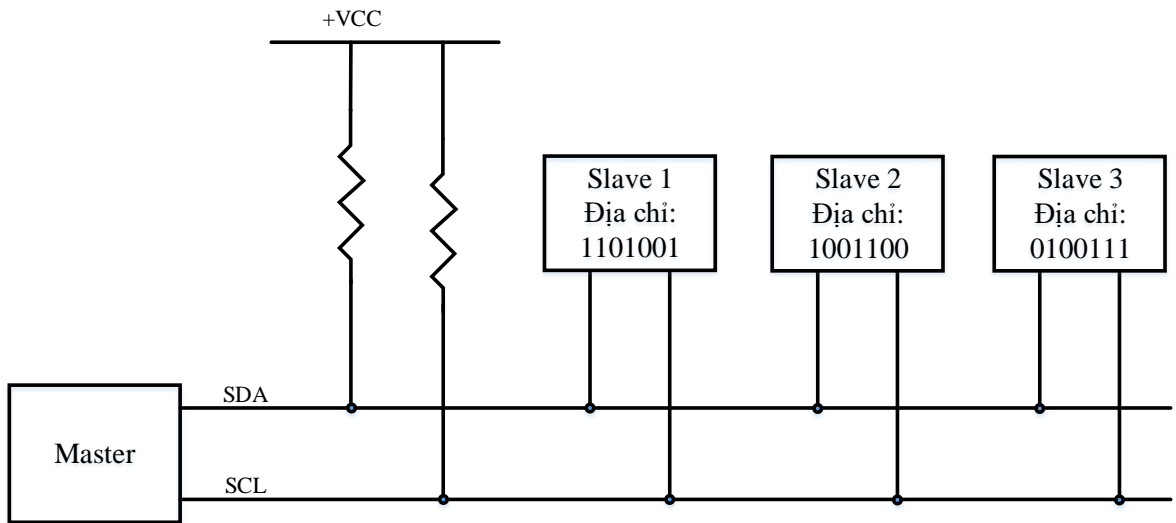


Hình 3.17 Mô hình giao diện SGMII

3.2.6 Giao diện điều khiển

3.2.6.1 Giao diện I²C

I²C (Inter-Integrated Circuit) là một giao diện giao tiếp để truyền dữ liệu giữa một bộ xử lý trung tâm (Master¹) với nhiều IC trên cùng một bảng mạch chỉ sử dụng hai đường truyền tín hiệu [23]. Đây là một loại giao diện giao tiếp nối tiếp đồng bộ, nghĩa là các bit dữ liệu được truyền từng bit một theo các khoảng thời gian đều đặn được thiết lập bởi một tín hiệu đồng bộ. Kênh I²C gồm hai tín hiệu là SCL (Serial Clock Line) và SDA (Serial Data Line). Dữ liệu được truyền đi được gửi qua đường SDA được đồng bộ với tín hiệu đồng bộ SCL. Tất cả các thiết bị trên mạng I²C được kết nối với cùng đường SCL, SDA và để phân biệt giữa nhiều slave² được kết nối với cùng một kênh I²C, mỗi thiết bị slave được gán một địa chỉ vật lý 7 bit cố định (Hình 3.18).



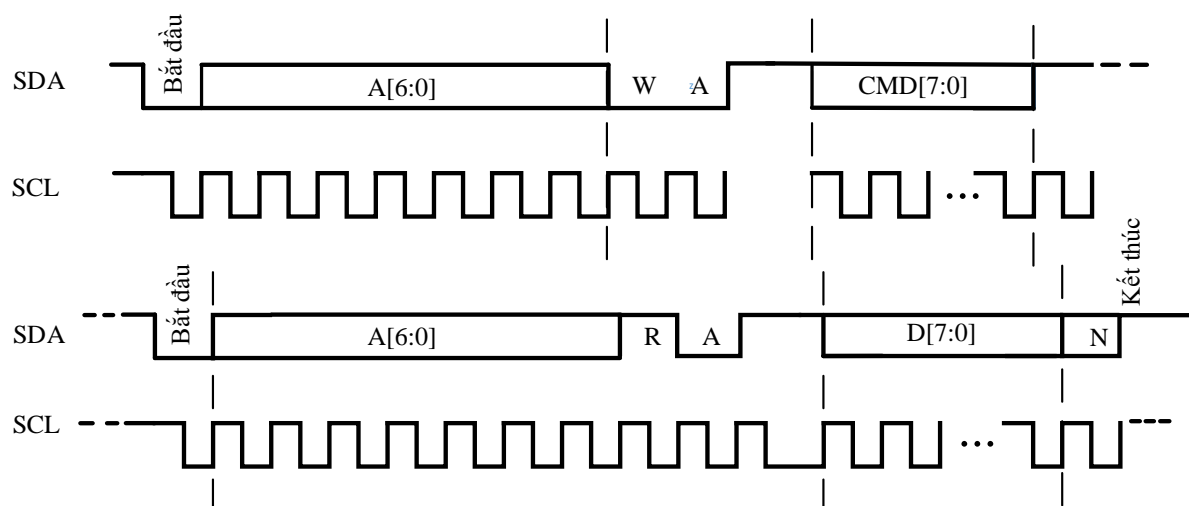
Hình 3.18 Sơ đồ kết nối mạng I²C

Ở trạng thái chờ cả hai tín hiệu SDA và SCL đều ở mức cao. Khi có một thiết bị muốn truyền dữ liệu thì SDA bị kéo xuống mức thấp trong khi SCL vẫn ở mức cao. Sau đó SCL dao động để tạo ra tín hiệu đồng bộ còn SDA truyền dữ liệu với chu kỳ bằng với SCL sao cho sườn lên SCL rơi vào giữa thời gian truyền của

¹ Master: Thiết bị chủ. Trong đồ án này sử dụng thay thế từ Master cho thiết bị chủ

² Slave: Thiết bị tớ. Trong đồ án này sử dụng thay thế từ Slave cho thiết bị tớ

1 bit dữ liệu trên SDA. Sau khi truyền xong dữ liệu SCL được đưa lên mức cao, còn SDA được đưa xuống mức thấp trong một chu kỳ bit. Cuối cùng cả hai được đưa lên mức cao quay về trạng thái chờ. Quá trình được mô tả trên Hình 3.19.



Hình 3.19 Quá trình truyền nhận dữ liệu giao thức I2C

Trong đó:

Địa chỉ : 7 bit địa chỉ vật lý của slave

R/W : Bit điều khiển đọc/ghi, W=0: đọc, R=1: ghi

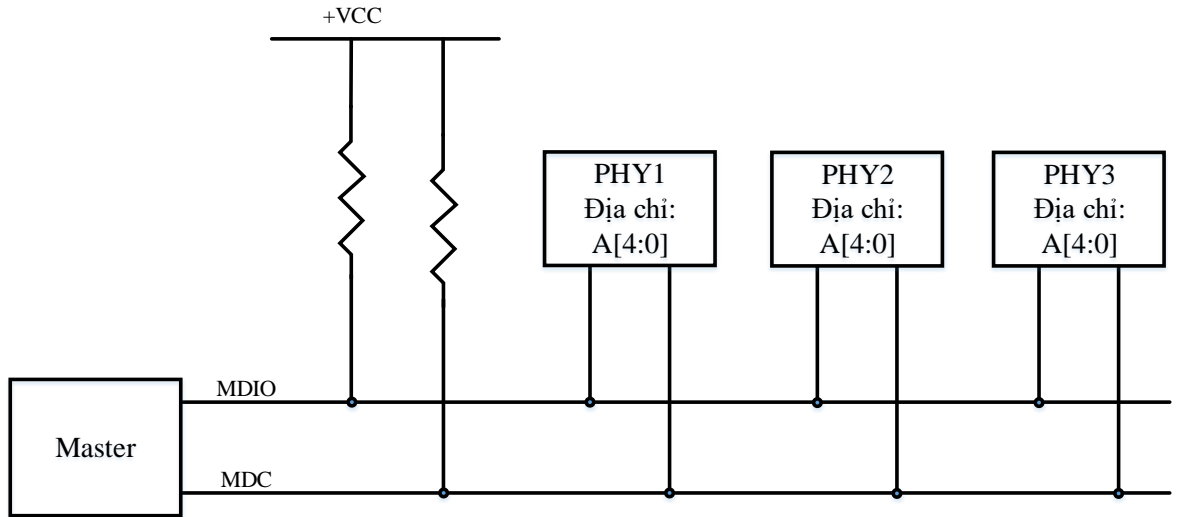
CMD[7:0] : Lệnh ghi từ master

D[7:0] : Dữ liệu từ slave

A(ACK)/N(NACK): Tín hiệu xác nhận

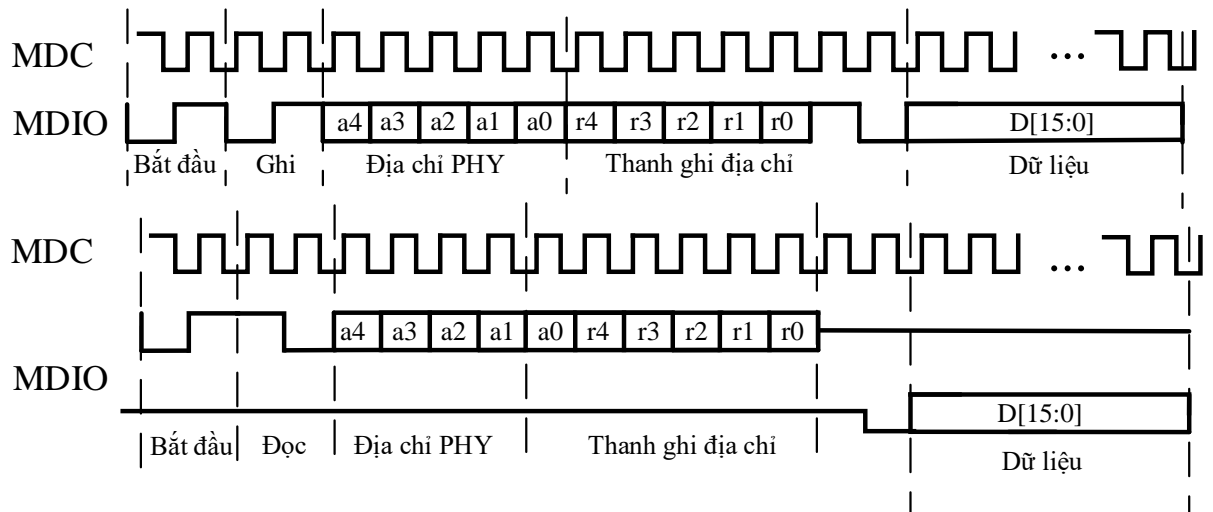
3.2.6.2 Giao diện MDIO

Giao diện MDIO là một giao diện giao tiếp để truyền dữ liệu giữa một bộ xử lý trung tâm với nhiều IC trên cùng một bảng mạch [15]. Cũng giống như I²C là một giao diện giao tiếp nối tiếp đồng bộ và sử dụng hai đường tín hiệu là MDC (tín hiệu đồng bộ) và MDIO (tín hiệu dữ liệu) (Hình 3.20). Tuy nhiên yêu cầu về tần số làm việc và khung dữ liệu khác hoàn toàn so với I²C. Tín hiệu MDC của giao diện MDIO luôn dao động để tạo tín hiệu đồng bộ.



Hình 3.20 Sơ đồ kết nối mạng MDIO

Trước khi truy cập, tối đa 32 PHY sẽ gửi địa chỉ đến Master trên tín hiệu MDIO. Quá trình truy cập bao gồm 16 bits điều khiển, tiếp theo là 16 bit dữ liệu. Các bit điều khiển bao gồm 2 bits bắt đầu, 2 bits truy cập (đọc hoặc ghi), địa chỉ PHY (5 bits), địa chỉ thanh ghi (5 bits) và 2 bits quay vòng trước khi truyền hoặc nhận 16 bits dữ liệu. Quá trình truyền nhận dữ liệu giữa thiết bị chủ và các PHY bằng giao thức MDIO được mô tả như hình Hình 3.21.



Hình 3.21 Quá trình truyền nhận dữ liệu giao thức MDIO

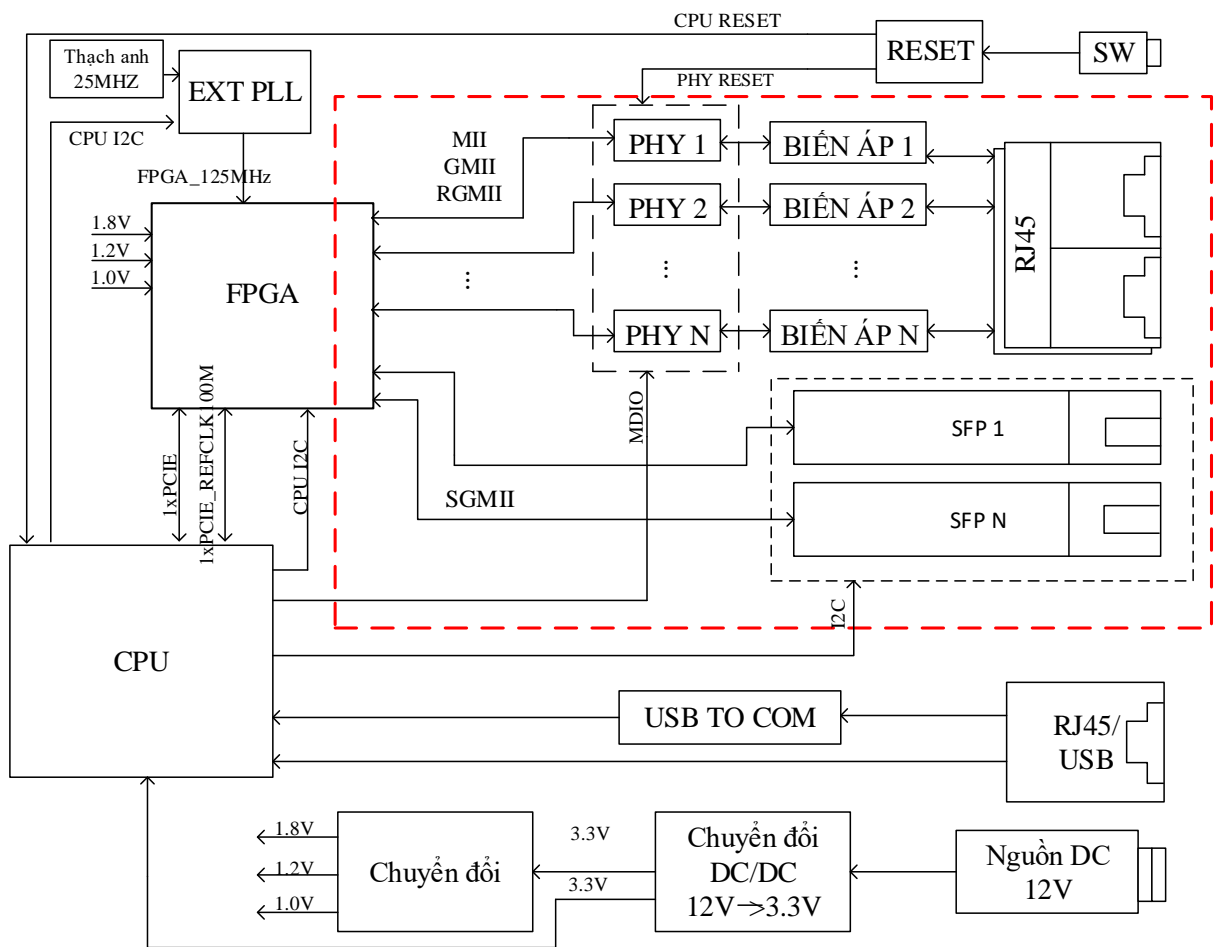
Trong đó:

- A[5:0] : Địa chỉ chip vật lý
- R[4:0] : Địa chỉ thanh ghi trong chip vật lý
- D[15:0] : 16 bit dữ liệu

3.3 Thiết kế mạch in cho phần giao tiếp vật lý

3.3.1 Sơ đồ tổng quan hệ thống Switch

Sơ đồ tổng quan của hệ thống Switch được thể hiện trên Hình 3.22 có các cổng với tốc độ tùy chọn 10 Mbps, 100 Mbps, 1000 Mbps. Phần giao diện người dùng được khoanh đỏ thể hiện trong Hình 3.22. Trong đó RJ45 và biến áp dùng để kết nối với cáp xoắn đôi, PHY1 đến PHYN biến đổi tín hiệu điện sau biến áp thành giao diện MII/GMII/RGMII. PHY1 đến PHYN được quản lý bằng giao diện MDIO. SFP là cổng kết nối đa năng được quản lý bằng giao diện I²C.



Hình 3.22 Sơ đồ tổng quan hệ thống Switch N cổng

3.3.2 Thiết kế mạch in cho các giao diện ngoại vi

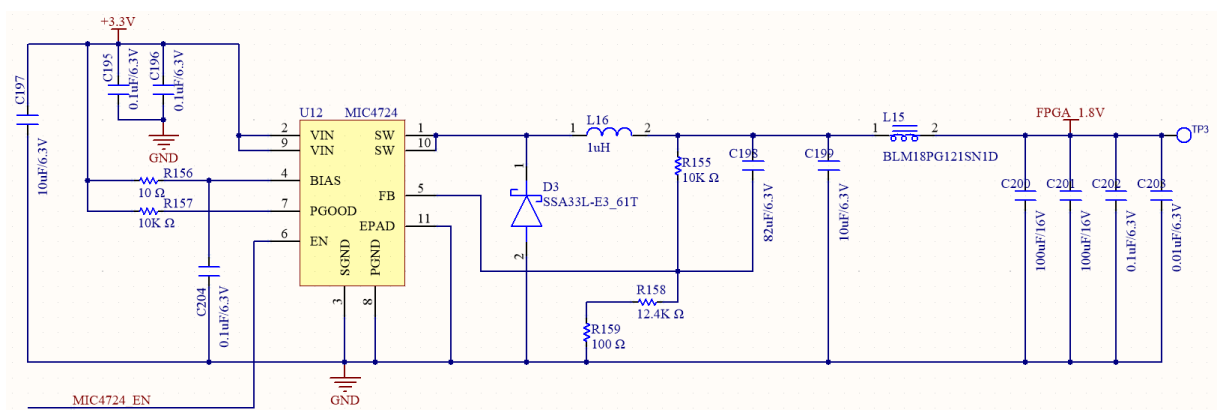
Phần mềm được sử dụng để thiết kế mạch in ở đây là Altium, đây là bộ phần mềm tích hợp cho phép thiết kế từ sơ đồ nguyên lý, mạch in cho tới việc trích xuất ra dữ liệu để chế tạo mạch (Gerber file). Altium hỗ trợ mạnh mẽ, đầy đủ các tính năng như: hỗ trợ các thành phần nhúng và quản lý lớp nâng cao; có

một thư viện rất đầy đủ các phần khác nhau với phân loại thích hợp; các công cụ thiết kế mã hóa và đồ họa với khả năng mô phỏng và gỡ lỗi VHDL; xem thuộc tính thích hợp, thông tin nhà cung cấp trong thời gian thực; cung cấp sức mạnh để dễ dàng tạo ra các thiết kế đa kênh và phân cấp... Altium cũng là một phần mềm hết sức phổ biến và các định dạng thiết kế có thể dễ dàng được trao đổi với các công cụ phần mềm khác.

3.3.2.1 Yêu cầu kỹ thuật

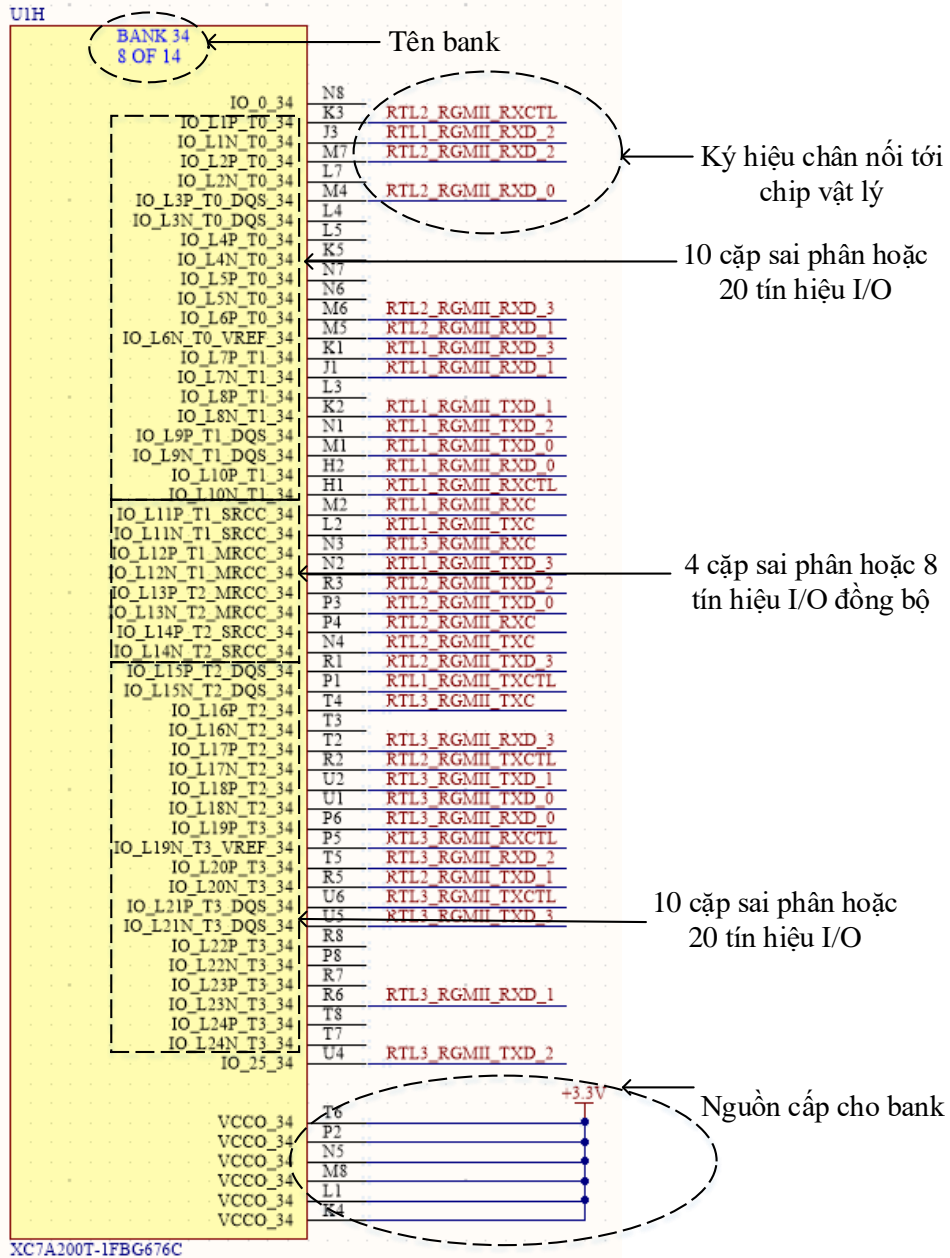
Thiết kế mạch in cho FPGA và các giao tiếp tốc độ cao có những yêu cầu kỹ thuật khá chặt chẽ và mạch in thường phải có ít nhất 6 lớp trở lên để đảm bảo khả năng bố trí linh kiện và kết nối.

Đối với kiến trúc FPGA chia các chân I/O thành các bank¹ đối với mỗi bank độ trễ xấp xỉ như nhau [33] do đó khi thiết kế mạch thì các chân tín hiệu của một giao diện RGMII nên thuộc cùng một bank để đảm bảo yêu cầu về độ trễ tín hiệu đã được nêu ở phần 3.2.4 (Hình 3.24). Khi thiết kế nguồn cấp cho FPGA phải đáp ứng được công suất lớn nhất khi PFGA hoạt động tất cả các chế độ, mỗi nguồn của FPGA cần bật đúng theo chu kỳ yêu cầu và của mỗi loại FPGA là khác nhau. Hình 3.23 ví dụ một trong số nguồn cấp cho FPGA, sử dụng IC MIC4724 cấp nguồn 1.8 V cho FPGA với đầu ra 3 A sai số tối đa 6% [20].



Hình 3.23 Nguồn 1.8 V cấp cho FPGA

¹ bank là một nhóm các chân i/o có cùng nguồn cấp điện áp và mức tín hiệu i/o giống nhau.



Single Region Clock Capable (SRCC): Chân kết nối tín hiệu đồng bộ đơn vùng

Multi-region Clock Capable (MRCC): Chân kết nối tín hiệu đồng bộ đa vùng

Hình 3.24 Chân IO trên bank 34 của XC7A200T-1FBG676C

3.3.2.2 Layout mạch in

Để đáp ứng yêu cầu về mật độ trễ RGMII thì sai lệch giữa các đường RXCTL/TXCTL, RXD/TXD, RXC/TXC nhỏ hơn 20 mil¹ để đảm bảo độ lệch thời gian giữa các đường này nhỏ hơn 3 ps (1 ps tương đương với độ trễ mạch 6 in²). Sai lệch giữa các đường dữ liệu TX, RX phải nhỏ hơn 50 mil, nhưng trong

¹ 1 mil = 0.0254 mm

² 1 in = 1000 mil

quá trình layout sai lệch giữa các đường dữ liệu này sẽ được để thấp nhất có thể và tốt nhất nếu sai lệch nhỏ hơn 5 mil.

Đối với các đường công suất, có dòng điện lớn chảy qua cần đáp ứng đủ độ rộng để đạt được công suất tối đa khi chảy qua nó. Ta có công thức độ rộng dây tối thiểu cần đáp ứng cho đường công suất là:

$$r = 10a \quad (3.1)$$

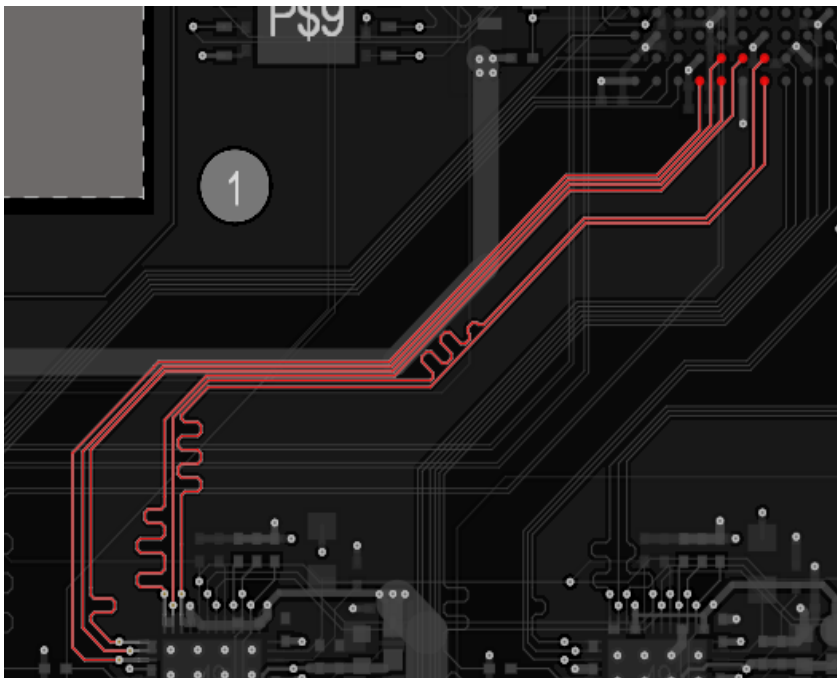
Trong đó:

r : Độ rộng dây tối thiểu cần đạt được [mil]

a : Dòng điện chảy qua [A]

Hình 3.25 và Bảng 3.4 là một ví dụ về sai lệch của các đường dữ liệu TXD và các đường TXC, TXCTL với độ sai lệch chỉ nhỏ hơn 1 mil tương đương thời gian trễ nhỏ hơn 0.3 ps. Ngoài ra thì khoảng cách từ FPGA tới chip vật lý (PHY) nhỏ hơn 250 mm, khoảng cách từ chip vật lý (PHY) tới biển áp nhỏ hơn 100 mm [22] để việc truyền nhận dữ liệu đạt hiệu quả tốt nhất.

Bảng 3.4 Số liệu sai lệch của nhóm RTL0_RGMII_TXD



Tên tín hiệu	Độ dài (mil)
TXC	2008.669
TXCTL	2008.669
TXD_0	2008.669
TXD_1	2008.539
TXD_2	2008.669
TXD_3	2008.669

Hình 3.25 Sai lệch của nhóm RTL0_RGMII_TXD

- Tính toán độ rộng dây là điều quan trọng để phối hợp trở kháng là tốt nhất, hiệu quả truyền nhận là cao nhất. Khi tính toán độ rộng dây cần quan tâm đến chất

liệu làm mạch (FR-4 đang được sử dụng phổ biến), độ dày của lớp đồng 1 oz (35 μm)/2 oz (75 μm).

- Đối với các đường vi sai thì giữa các cặp phải cách nhau tối thiểu là:

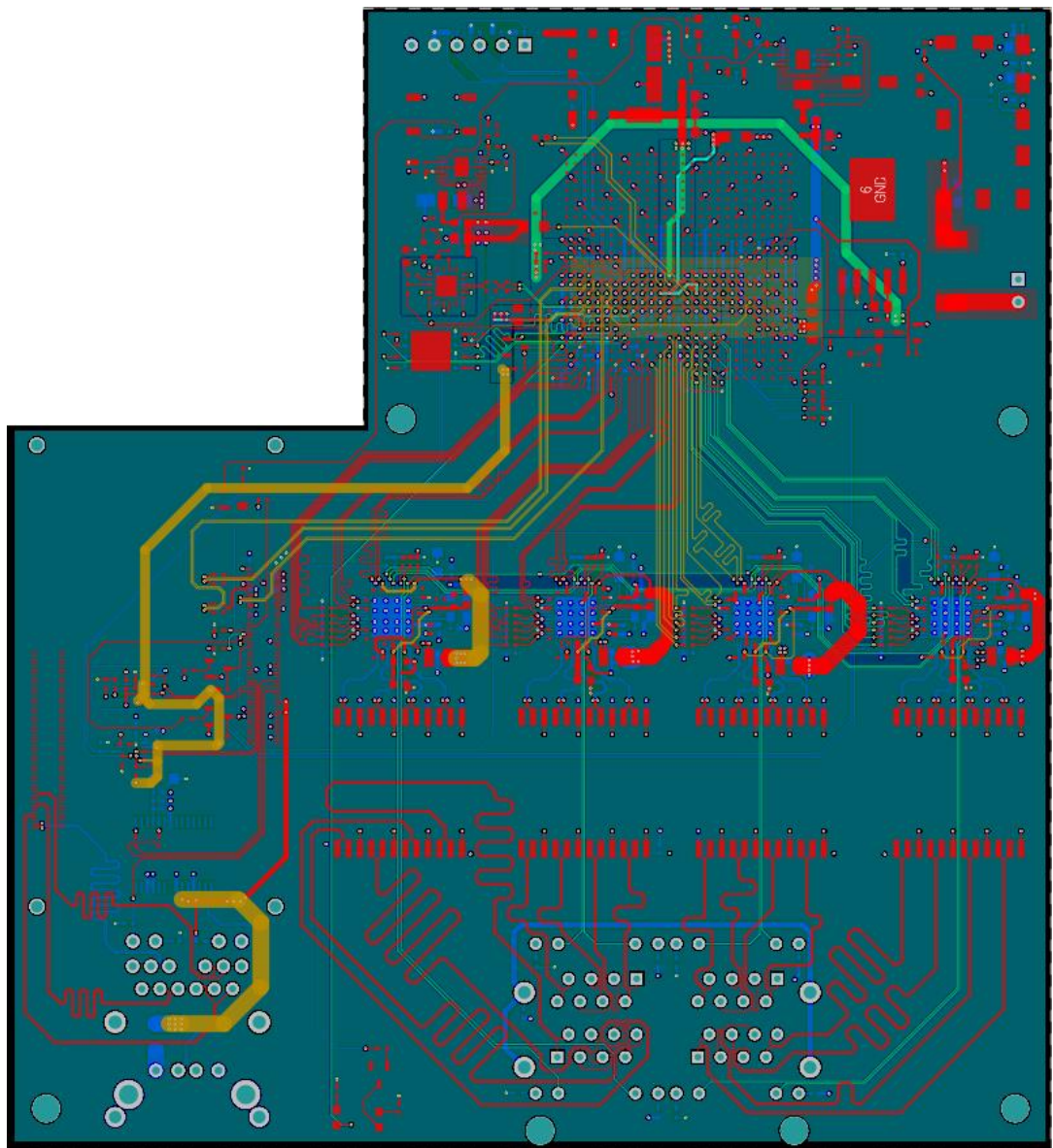
$$d = 3w \quad (3.2)$$

Trong đó:

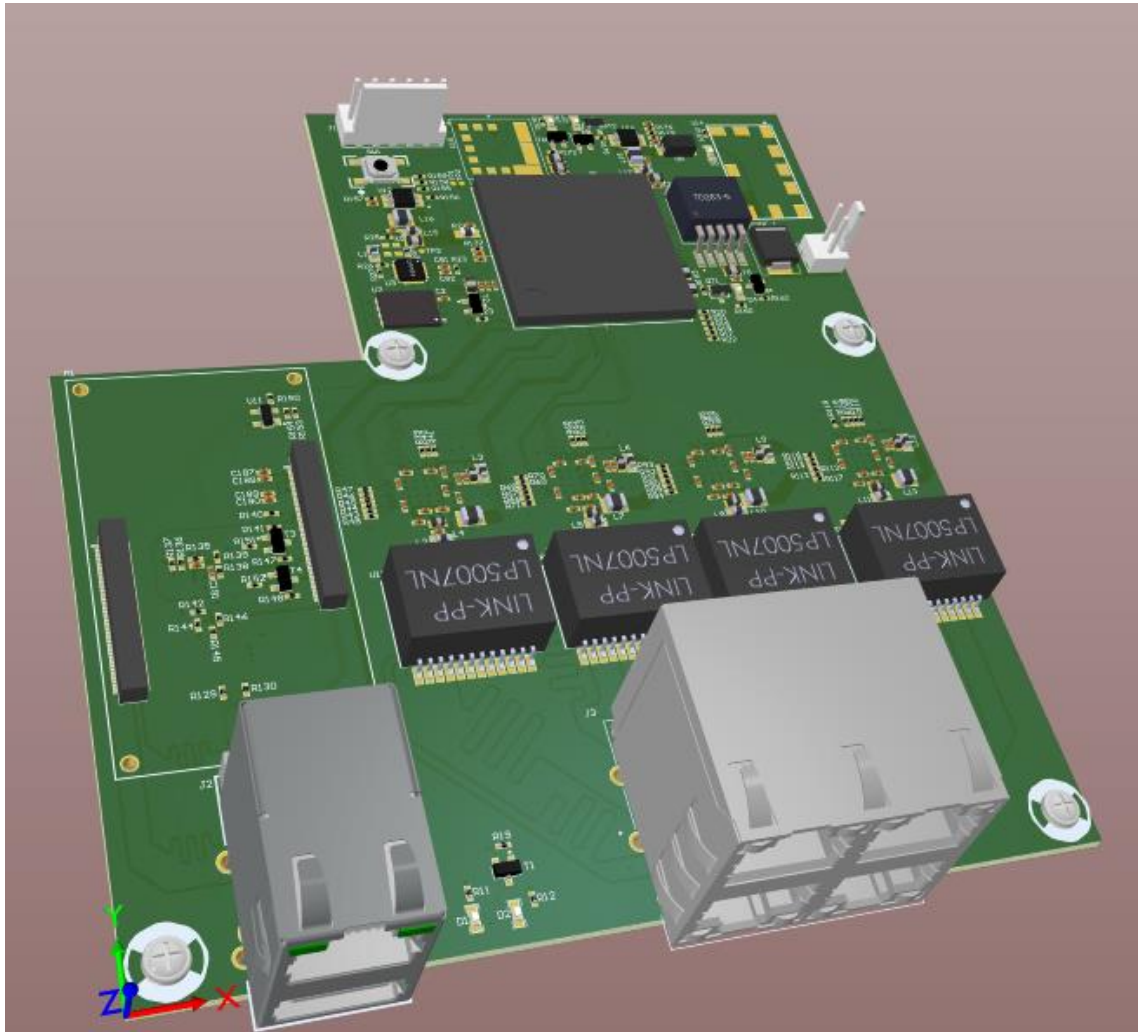
d : khoảng cách tối thiểu giữa các cặp đường vi sai [mil]

w : độ rộng đường dây vi sai [mil]

Hình 3.26 và Hình 3.27 ví dụ layout của Switch 4 cổng RJ45 tốc độ cao nhất đạt được 1000 Mbps.



Hình 3.26 Ví dụ layout của Switch 4 cổng dưới góc nhìn 2D

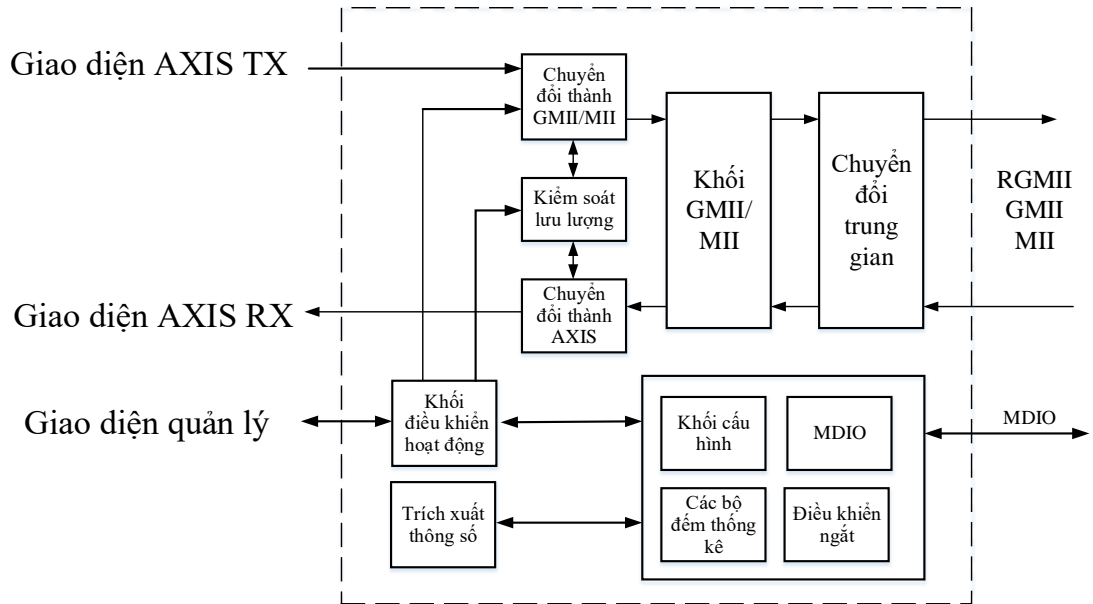


Hình 3.27 Ví dụ Layout của Switch 4 cổng dưới góc nhìn 3D

3.4 Thiết kế giao tiếp MAC

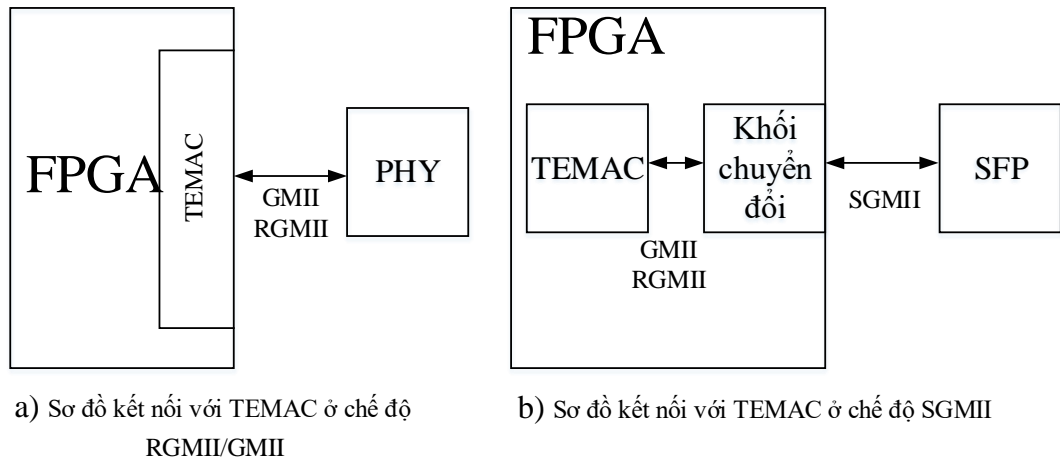
3.4.1 Giới thiệu tổng quan TEMAC

Lõi sở hữu trí tuệ bán dẫn Ethernet MAC ba chế độ (Tri-Mode Ethernet MAC Intellectual Property Core – TEMAC IP Core) là một khối có thể thay đổi tham số để phù hợp với nhiều thiết bị mạng như bộ chuyển mạch, bộ định tuyến. TEMAC cho phép triển khai Ethernet hỗ trợ nhiều tốc độ bao gồm 10 Mbps, 100 Mbps, 1000 Mbps và 2500 Mbps. TEMAC được thiết kế theo chuẩn IEEE 802.3. Đối với chế độ 1000/2500 Mbps TEMAC kết nối với Xilinx 1G/2.5G PCS/PMA. Ở chế độ 1000 Mbps, TEMAC có thể kết nối với các PHY theo chuẩn công nghiệp với giao diện GMII hoặc RGMII. Ở chế độ 10/100 Mbps, TEMAC sử dụng giao diện MII.



Hình 3.28 Sơ đồ khối chức năng của TEMAC [39]

Ở chế độ RGMII/GMII thì các tín hiệu vào ra của TEMAC được nối trực tiếp với các chân I/O của FPGA và nối tới chip vật lý (Hình 3.29a). Ở chế độ SGMII thì các tín hiệu vào ra của TEMAC được nối với một khối chuyển đổi SGMII thành RGMII/GMII trước khi đưa ra chân I/O của FPGA nối tới cổng SFP (Hình 3.29b).



Hình 3.29 Sơ đồ kết nối với TEMAC ở chế độ RGMII/GMII và SGMII

3.4.2 Giao diện AMBA® AXI4-Stream

Để có thể xử lý việc đọc ghi nhiều luồng với tốc độ cao, ta sử dụng giao thức AMBA® (Advanced Microcontroller Bus Architecture) AXI-Stream cho giao tiếp giữa các khối trong FPGA. AMBA là giao diện điều khiển kiến trúc kênh dữ liệu cho vi điều khiển được phát triển cho ARM. Giao diện này giúp kết nối

nhiều luồng dữ liệu và có khả năng đọc ghi độc lập. Thế hệ 3 và 4 của AMBA được gọi là Advanced eXtensible Interface, là một giao diện cung cấp khả năng xử lý nhiều luồng song song với hiệu năng cao, hỗ trợ nhiều master/slave (chủ/tớ) cùng giao tiếp trên một kênh duy nhất, chủ yếu sử dụng cho các thiết kế mạng truyền thông trong vi mạch.

Theo như AMBA®, AXI có các ưu điểm sau [8]:

- Phù hợp cho các thiết kế yêu cầu băng thông rộng và độ trễ thấp.
- Có thể hoạt động ở tần số cao mà không cần các cầu nối phức tạp.
- Phù hợp yêu cầu giao diện của rất nhiều thành phần.
- Cung cấp tính linh hoạt cao khi thực hiện trên các kiến trúc đa kết nối.

Cũng theo AMBA®, AXI cung cấp các tính năng sau [8]:

- Điều khiển độc lập dựa trên địa chỉ.
- Hỗ trợ truy cập không đồng bộ.
- Hỗ trợ kênh dữ liệu kích thước đa dạng.
- Kênh đọc ghi độc lập.
- Hỗ trợ chia luồng ngay cả trên một master.

AMBA® AXI cũng cung cấp nhiều tín hiệu tùy chọn khác có thể tùy biến theo mục đích người dùng, cho nên đây là giao diện đa năng có thể dùng cho rất nhiều thiết kế khác nhau.

Giao thức AXI4-Stream là một giao thức trong họ AXI có hỗ trợ hỗ trợ truyền nhận dữ liệu theo luồng có kích thước khác nhau và có thể phục vụ đồng thời nhiều luồng dữ liệu hoặc nhiều dịch vụ khác nhau trên một luồng. Đây là một trong những giao thức được dùng nhiều trong thiết kế Switch và các thiết bị chuyển mạch nói chung. AXI4-Stream cung cấp các định nghĩa sau [8]:

a) Định nghĩa byte

Byte dữ liệu: là một byte của dữ liệu chứa thông tin cần truyền từ nguồn tin đến đích.

Byte vị trí: là một byte thể hiện mối quan hệ vị trí của dữ liệu trong luồng dữ liệu. Byte này không chứa bất kỳ thông tin liên quan nào đến dữ liệu cần truyền.

Byte trống: là một byte không chứa bất kì thông tin nào, ví dụ như byte được thêm vào để có đủ một khối dữ liệu.

b) Định nghĩa tín hiệu

AXI4-Stream định nghĩa các chân tín hiệu như Bảng 3.5.

Bảng 3.5 Các chân tín hiệu của AXI-Stream

Tên tín hiệu	Nguồn tín hiệu	Mô tả
ACLK	Nguồn đồng bộ chung	Tín hiệu đồng bộ cho cả truyền và nhận. Mọi tín hiệu được lấy mẫu tại sườn lên của ACLK.
ARESETN	Nguồn tạo tín hiệu khởi tạo	Tín hiệu khởi tạo. ARESETN là tín hiệu tích cực thấp.
TVALID	Master	Thông báo rằng khối master có dữ liệu muốn truyền đi.
TREADY	Slave	Thông báo rằng khối slave sẵn sàng nhận dữ liệu. Dữ liệu chỉ được truyền đi khi đồng thời cả TVALID và TREADY đều ở mức cao.
TDATA[(8n-1):0]	Master	Kênh dữ liệu. Kích thước kênh dữ liệu được tính theo byte.
TSTRB[(n-1):0]	Master	Thẻ hiện thông tin của byte ở vị trí tương ứng cần được coi là byte dữ liệu hay là byte chỉ vị trí.
TKEEP[(n-1):0]	Master	Thẻ hiện byte nào mang thông tin cần được xử lí. Những byte được đánh dấu là trống có thể loại bỏ khỏi luồng dữ liệu.
TLAST	Master	Báo hiệu kết thúc của một lần truyền.
TID[(i-1):0]	Master	Chỉ số nhận dạng của các luồng dữ liệu.
TDEST[(d-1):0]	Master	Cung cấp thông tin đích cho luồng dữ liệu.
TUSER[(u-1):0]	Master	Cung cấp dữ liệu người dùng tự định nghĩa để truyền các thông tin liên quan đến luồng dữ liệu.

Trong đó:

- n : Độ rộng kênh dữ liệu [byte]
- i : Độ rộng của TID [bit]
- d : Độ rộng của TDEST [bit]
- u : Độ rộng của TUSER [bit]

c) Truyền nhận dữ liệu

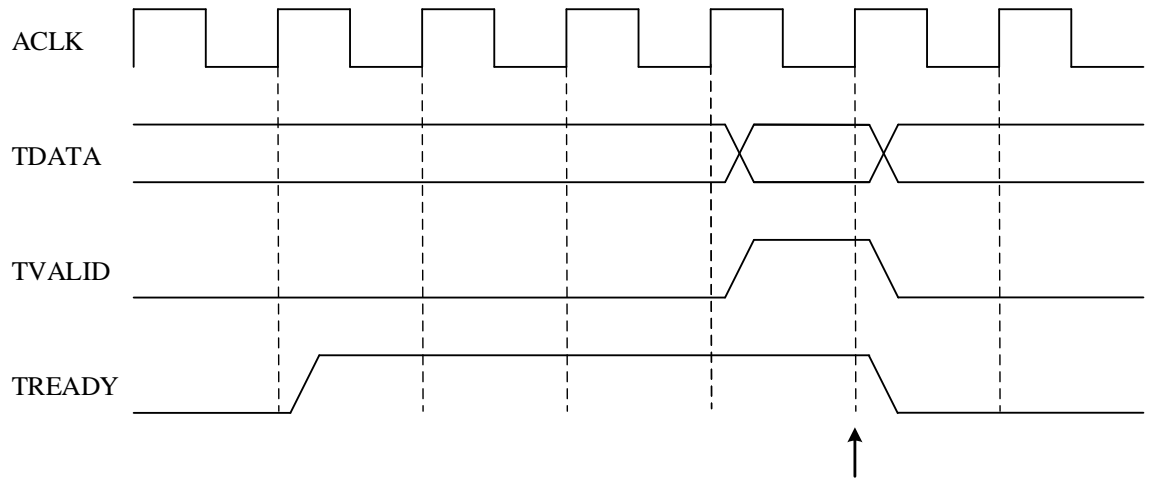
Việc truyền nhận dữ liệu thông qua AXI-Stream yêu cầu quá trình bắt tay giữa master và slave. Quá trình bắt tay được thực hiện dựa vào hai tín hiệu TVALID và TREADY. Dữ liệu chỉ được coi là truyền thành công khi cả TVALID và TREADY đều ở mức cao. TVALID và TREADY có thể đặt ở mức cao khác chu kì hoặc cùng chu kì ACLK.

Master không được đợi cho TREADY được đặt mức cao rồi mới đặt TVALID. Một khi master TVALID được đặt mức cao thì phải duy trì cho đến khi quá trình bắt tay thành công.

Slave có thể đợi cho đến khi có TVALID rồi mới đặt TREADY lên mức cao để phản hồi. Khi TREADY đã được đặt mức cao, slave có thể xoá TREADY bất cứ lúc nào, kể cả khi đang có hoặc không có TVALID.

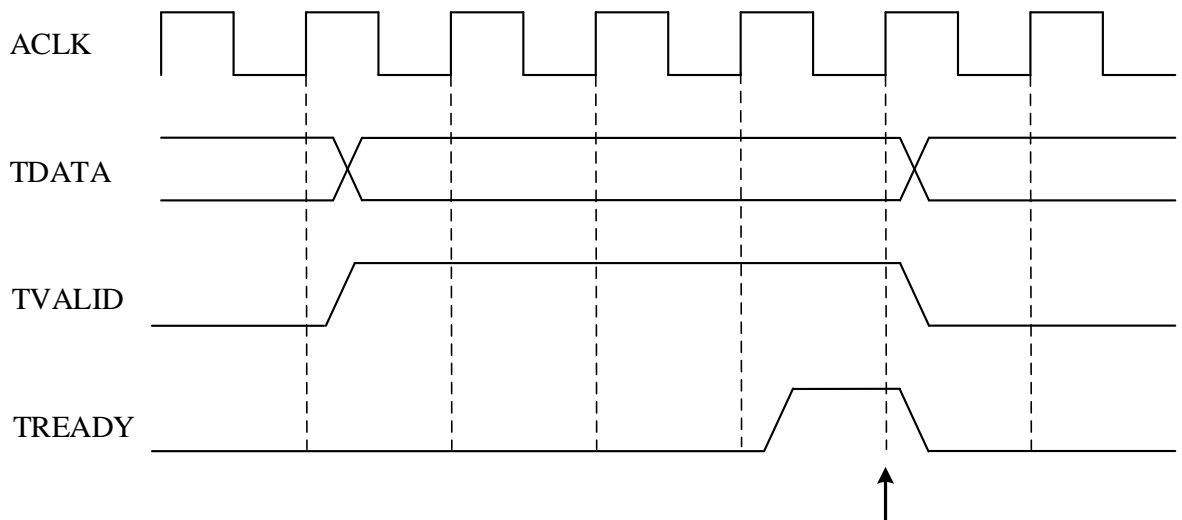
Dưới đây là một vài ví dụ về quá trình bắt tay [8]:

Bắt tay khi TREADY có trước TVALID: Hình 3.30 mô tả slave đặt TREADY mức cao trước khi master có dữ liệu. Điều này cho thấy thiết bị đích có thể nhận dữ liệu và các tín hiệu điều khiển trong một chu kì ACLK. Trong trường hợp này, dữ liệu được truyền thành công tại chu kì ACLK mà TVALID = 1. Mũi tên biểu thị thời điểm truyền thành công.



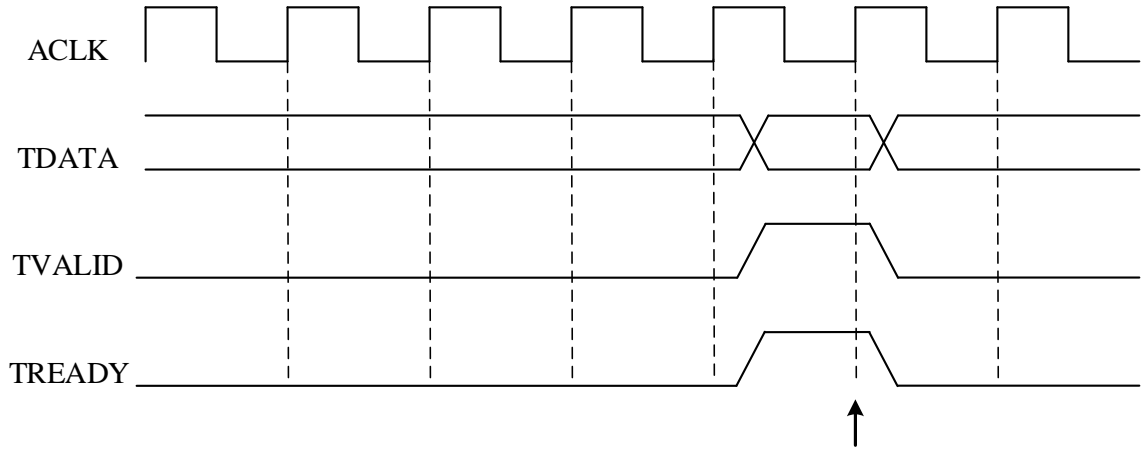
Hình 3.30 Quá trình bắt tay khi TREADY có trước TVVALID

Bắt tay khi TVVALID có trước TREADY: Hình 3.31 mô tả master có dữ liệu cần gửi ra và đặt TVVALID = 1. Một khi TVVALID = 1 thì TDATA phải giữ nguyên cho đến khi quá trình bắt tay thành công, nghĩa là phải đợi cho TREADY = 1. Lúc đó slave mới sẵn sàng nhận dữ liệu và việc truyền dữ liệu thành công ngay khi TREADY = 1. Mũi tên biểu thị thời điểm truyền thành công.



Hình 3.31 Quá trình bắt tay khi TVVALID có trước TREADY

Bắt tay khi TVVALID có đồng thời với TREADY: Hình 3.32 biểu thị TVVALID và TREADY cùng ở mức cao trong cùng chu kỳ ACLK. Khi đó dữ liệu được truyền thành công ngay tại sườn lên của chu kỳ sau.



Hình 3.32 Quá trình bắt tay khi TVALID và TREADY có đồng thời

d) Định nghĩa dữ liệu

TDATA là đường dữ liệu dùng để truyền thông giữa master và slave.

Đánh số thứ tự cho từng byte: Trong một luồng dữ liệu, byte có chỉ số thấp là byte đến trước trong luồng. Trong một luồng dữ liệu đầy đủ, không có byte trống, vị trí của byte được định nghĩa như sau:

- Thứ tự byte n trong luồng được đánh số bắt đầu từ 0.
- Thứ tự lần truyền thành công t được đánh số bắt đầu từ 0.
- Độ rộng kênh dữ liệu là w bytes.
- $\text{INT}(x)$ là hàm số làm tròn xuống của số nguyên x .

$$\text{INT}(x) = \lfloor x \rfloor \quad (5.1)$$

Như vậy, byte n sẽ được truyền đi tại lần truyền t , với:

$$t = \text{INT}(n/w) \quad (5.2)$$

tại byte thứ b với:

$$b = n - t \times w \quad (5.3)$$

tương ứng với $\text{TDATA}[(8b+7):8b]$.

Kiểu byte: Các kiểu byte trong TDATA bao gồm 3 kiểu byte đã được định nghĩa là byte dữ liệu, byte vị trí và byte trống.

Hợp, tách, chuyển đổi độ rộng kênh: Một thiết kế có thể làm việc với độ rộng kênh linh hoạt rất dễ ứng dụng trên nhiều thiết bị. Nhưng thiết kế như vậy không hề đơn giản. Nếu hướng tới một thiết kế được tối ưu cho một độ rộng kênh

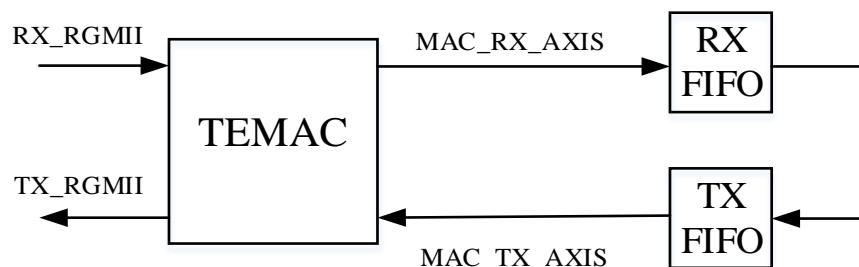
nhất định, nhưng có thể làm việc với nhiều độ rộng kênh khác nhau thông qua các bộ chuyển đổi thì việc thiết kế trở nên dễ dàng hơn rất nhiều. AXI4-Stream có các lưu ý giúp thiết kế các bộ chuyển đổi độ rộng kênh dễ dàng hơn.

Thường trong đa số ứng dụng, giao diện của chúng có độ rộng tính theo byte là một lũy thừa của 2. Tuy nhiên, giao thức AXI4-Stream không có yêu cầu cố định về độ rộng kênh, trừ việc kênh dữ liệu phải là một số nguyên lần theo byte, cho nên độ rộng kênh của giao diện tính theo byte có thể là bất cứ số nguyên dương nào.

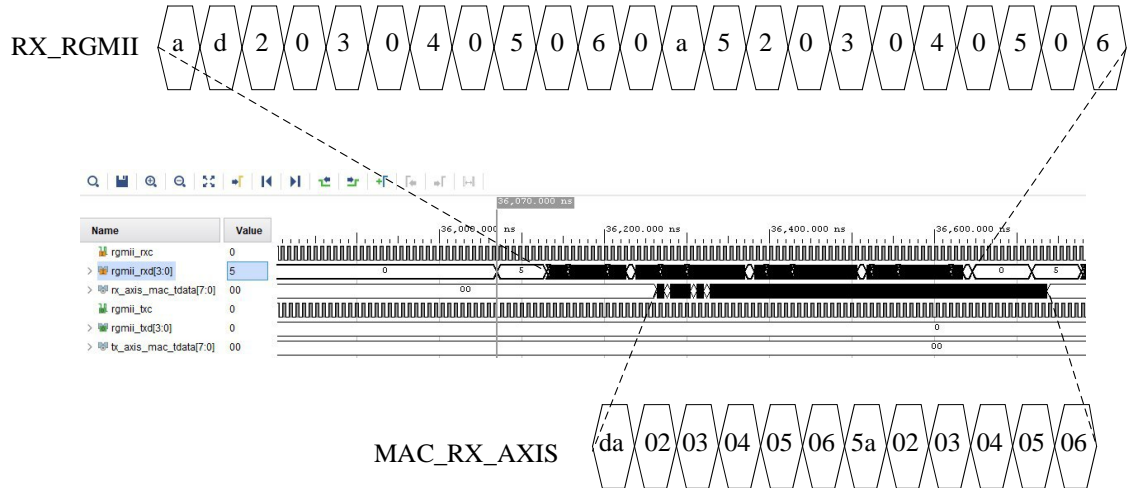
Mọi chuyển đổi độ rộng kênh đều cần phải truyền đạt được mọi byte có nghĩa, tức là chúng có bit TKEEP tương ứng là 1. Nếu đầu ra không phải là số nguyên lần của số byte có nghĩa, các byte trống sẽ được thêm vào để đầu ra có độ rộng yêu cầu, đồng thời các bit TKEEP tương ứng với các byte trống thêm vào được đặt bằng 0 để báo hiệu.

3.4.3 Mô phỏng thiết kế TEMAC cho giao tiếp RGMII

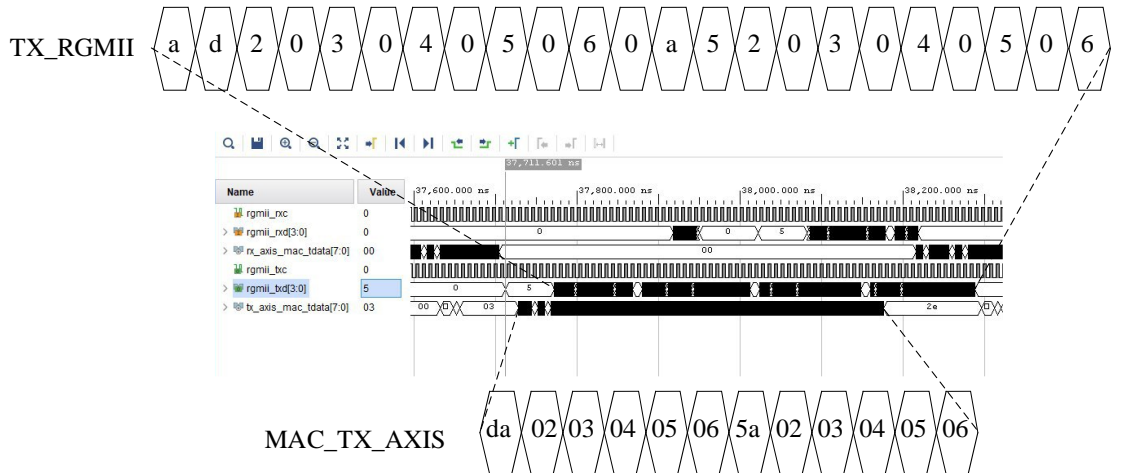
Mô phỏng bằng phương pháp nối lặp (Hình 3.33): Tạo dữ liệu RGMII truyền vào phía RX_RGMII của TEMAC, tín hiệu được biến đổi RX_RGMII thành AXIS và đẩy ra MAC_RX_AXIS, được ghi vào RX FIFO và sau đó đến TX FIFO (RX FIFO và TX FIFO được dùng để đồng bộ giữa MAC_RX_AXIS và MAC_TX_AXIS). Sau đó TEMAC biến đổi MAC_TX_AXIS thành TX_RGMII và truyền đi. Kiểm tra trên màn hình mô phỏng nếu RX_RGMII và TX_RGMII giống nhau thì kết luận TEMAC được cấu hình đúng (Hình 3.34).



Hình 3.33 Mô hình mô phỏng bằng phương pháp nối lặp



Hình 3.34 Giản đồ sóng tín hiệu RX_RGMII và MAC_RX_AXIS

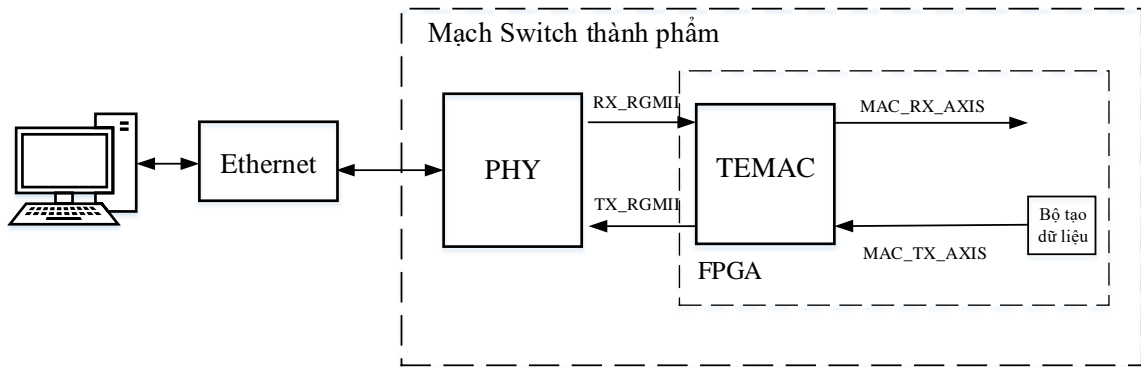


Hình 3.35 Giản đồ sóng tín hiệu TX_RGMII và MAC_TX_AXIS

3.4.4 Kiểm tra trên mạch

a) Kiểm tra đường truyền phần TX

Tạo dữ liệu MAC_TX_AXIS theo chuẩn khung Ethernet để truyền đến TEMAC. TEMAC sẽ biến đổi MAC_TX_AXIS thành TX_RGMII đến phía TX chip PHY, sau đó chip PHY biến đổi TX_RGMII thành tín hiệu để truyền trên phương tiện truyền dẫn Ethernet. Kết nối máy tính với đường truyền để nhận lại khung dữ liệu đã truyền đi (Hình 3.36). Bằng cách so sánh dữ liệu nhận về trên máy tính (bằng phần mềm Wireshark) với dữ liệu được tạo ra từ bộ tạo dữ liệu (Hình 3.37). Nếu hai bên giống nhau ta khẳng định đường truyền phía TX đảm bảo chính xác. Lưu ý ra dữ liệu đẩy lên theo đường TX phải có kích thước đủ lớn (~1518 bytes) để kết luận đường truyền cơ bản không bị lỗi bit.



Hình 3.36 Sơ đồ kiểm tra đường truyền phía TX

No.	Time	Source	Destination	Protocol	Length	Info
385	0.001080	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	495	U P, func=SNRME; DSAP
386	0.001080	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	496	I P, N(R)=111, N(S)=11
387	0.001081	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	60	I P, N(R)=21, N(S)=22;
388	0.001081	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	61	S, func=SREJ, N(R)=22;
389	0.001081	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	62	I P, N(R)=22, N(S)=23;
390	0.001082	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	63	U, func=SABM; DSAP 0x3
391	0.001082	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	64	I P, N(R)=23, N(S)=24;
392	0.001082	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	65	S, func=RR, N(R)=24; D
393	0.001082	00:0a:35:02:66:ea	5a:01:02:03:04:05	ISO	66	Unknown ISO protocol (
394	0.001083	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	67	U P, func=UP; DSAP ISO
395	0.001083	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	68	I P, N(R)=25, N(S)=26;
396	0.001083	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	69	S, func=RRR, N(R)=26;
397	0.001084	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	70	I P, N(R)=26, N(S)=27;
398	0.001084	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	71	U P, func=Unknown; DSA
399	0.001084	00:0a:35:02:66:ea	5a:01:02:03:04:05	LLC	72	I P, N(R)=27, N(S)=28;

Offset	Hex	ASCII
0000	5a 01 02 03 04 05 00 0a 35 02 66 ea 00 2e 2e 2d	Z-----5-f----
0010	2c 2b 2a 29 28 27 26 25 24 23 22 21 20 1f 1e 1d	,+*)('&% \$#! ...
0020	1c 1b 1a 19 18 17 16 15 14 13 12 11 10 0f 0e 0d
0030	0c 0b 0a 09 08 07 06 05 04 03 02 01

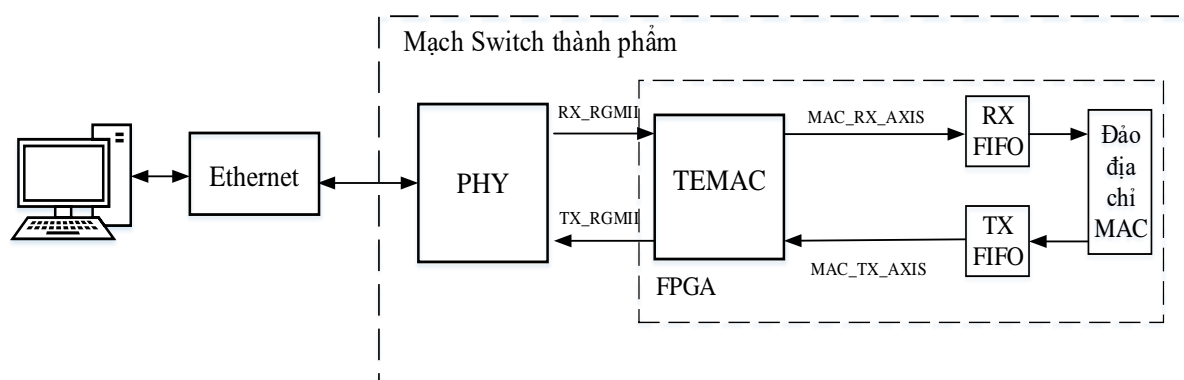
Hình 3.37 Kết quả thu được trên Wireshark khi kiểm tra đường truyền TX

b) Kiểm tra RX bằng phương pháp nối vòng dữ liệu

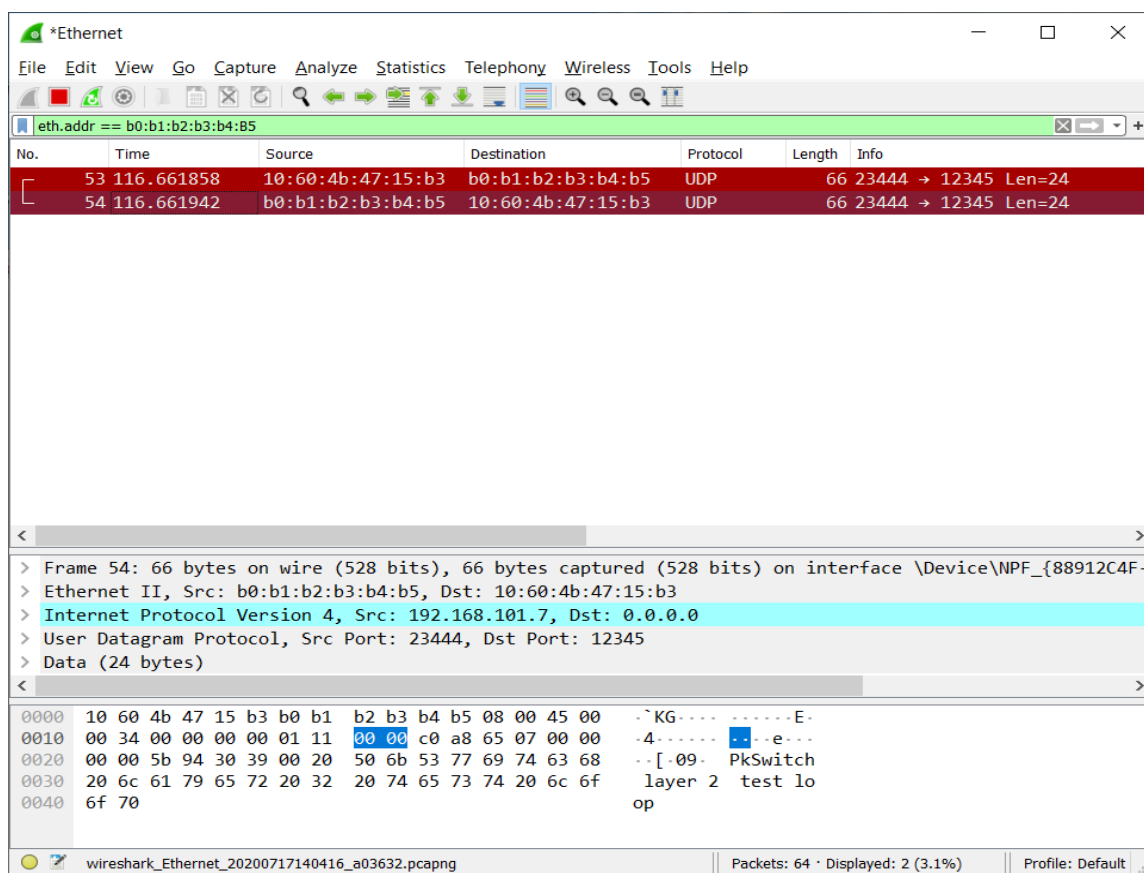
Sau khi đường truyền phần TX đảm bảo cổng Ethernet hoạt động thì phải kiểm tra phần RX. Tạo khung dữ liệu Ethernet từ máy tính (bằng phần mềm SendPacket¹) truyền xuống đường truyền Ethernet. Kết nối mạch Switch thành

¹ Send packet là một phần mềm được phát triển bởi nhóm làm đề tài sử dụng ngôn ngữ C# và Visual Studio với mục đích truyền nhận gói tin UDP qua cổng Ethernet nối từ máy tính xuống mạch FPGA. Chi tiết về phần mềm này không trình bày kỹ trong đồ án vì đó chỉ là thiết kế phụ trợ.

phẩm vào đường truyền để truyền nhận dữ liệu theo như hình. Quan sát dữ liệu thu được trên máy tính (bằng phần mềm Wireshark¹), nếu dữ liệu thu được giống với dữ liệu truyền đi (chỉ có địa chỉ MAC bị đảo giữa nguồn và đích) thì ta khẳng định cả đường truyền cả phía TX và RX đều chính xác. Kết quả thu được trên Wireshark và cấu trúc gói tin mô tả trên Hình 3.38, Hình 3.39 và Hình 3.40.

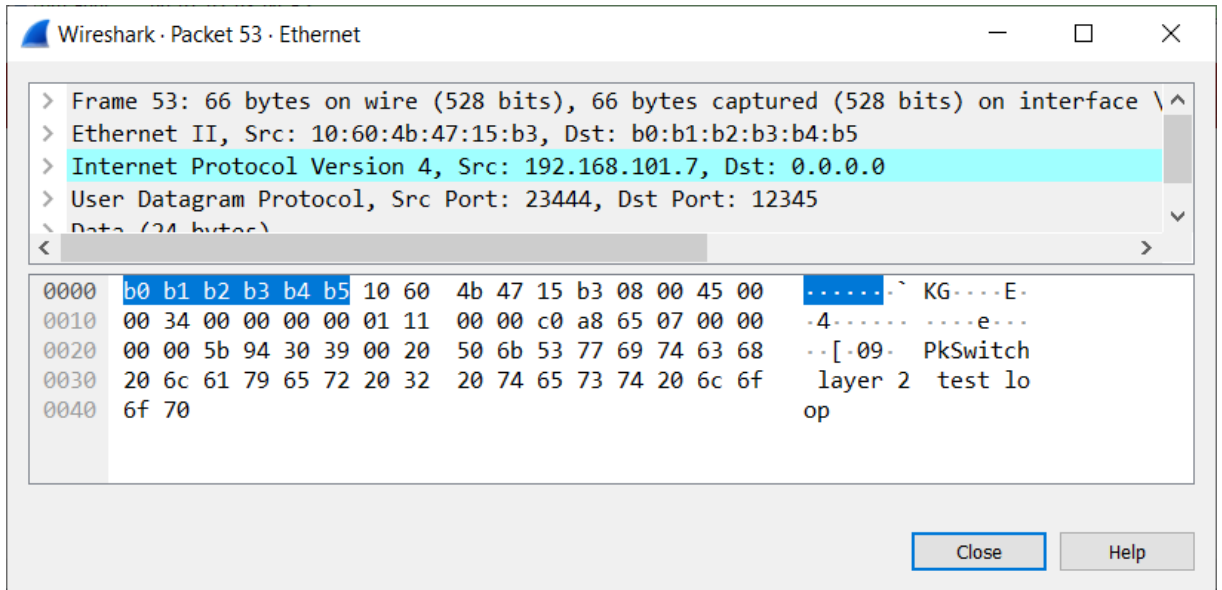


Hình 3.38 Sơ đồ kiểm tra RX bằng phương pháp nối vòng dữ liệu

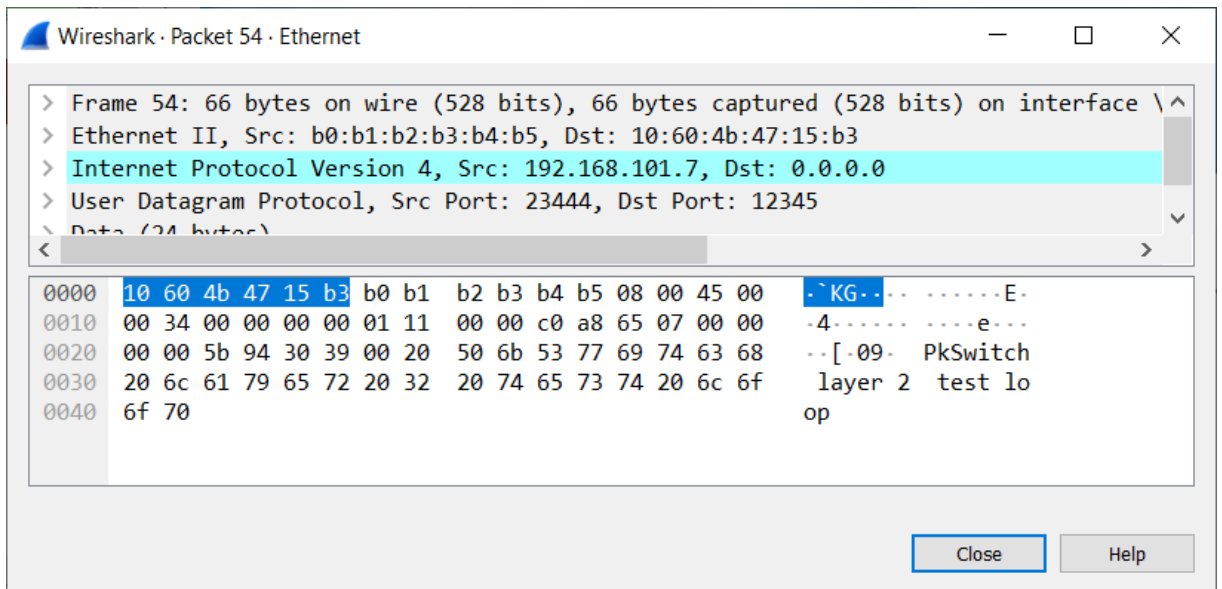


Hình 3.39 Kết quả thu được trên Wireshark khi kiểm tra đường truyền RX

¹ Wireshark là một phần mềm có thể theo dõi các gói tin đi qua các giao diện Ethernet trên máy tính, có sẵn tại trang web <https://www.wireshark.org/>.



Hình 3.40 Cấu trúc gói tin gửi đi

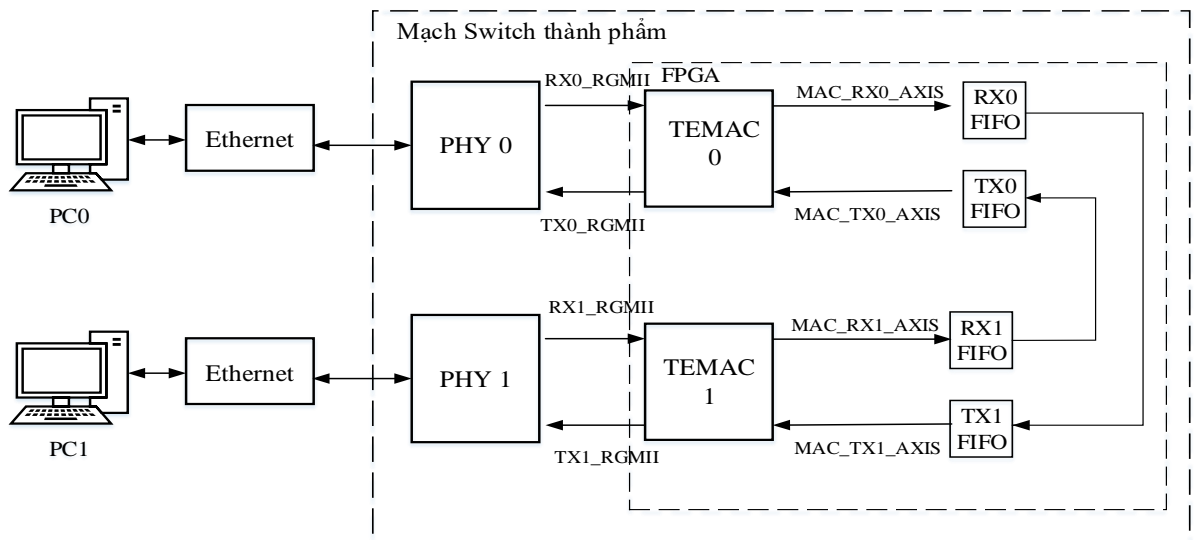


Hình 3.41 Cấu trúc gói tin nhận về

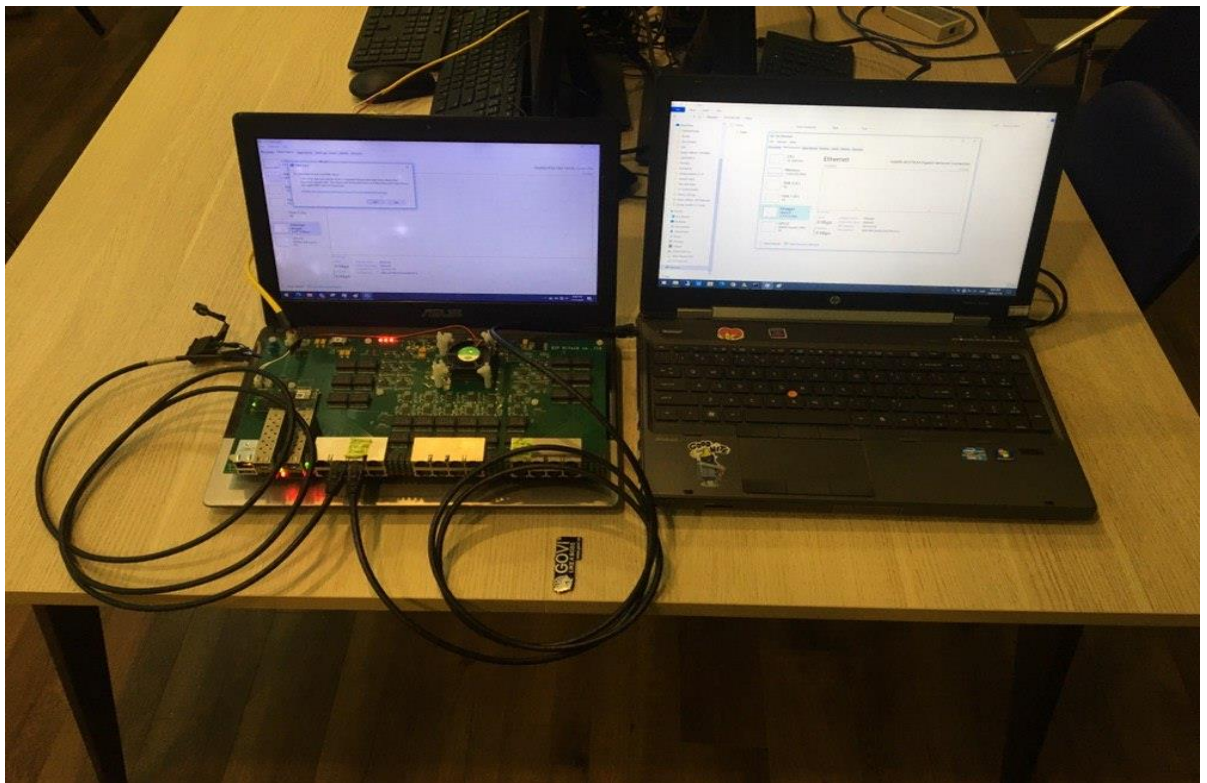
c) Kiểm tra nối chéo trong FPGA giữa hai TEMAC để truyền nhận dữ liệu

Kết nối hai máy tính với mạch Switch thành phẩm đã được nối vòng hai TEMAC trong FPGA (Hình 3.42, Hình 3.43). Thử truyền nhận dữ liệu giữa hai máy tính bằng lệnh ping và sao chép dữ liệu. Nếu hai máy ping được cho nhau nghĩa là đường truyền trong mạch Switch được đảm bảo chính xác. Sau đó thực hiện sao chép dữ liệu giữa hai máy nếu tốc độ truyền nhận dữ liệu đạt ~125 MB/s thì đường truyền trên mạch Switch đạt tốc độ 1000 Mbps. Ta thực hiện kiểm tra

ba phương pháp trên với tất cả các cổng trên mạch Switch thành phẩm để đảm bảo tất cả giao diện Ethernet đều hoạt động tốt.

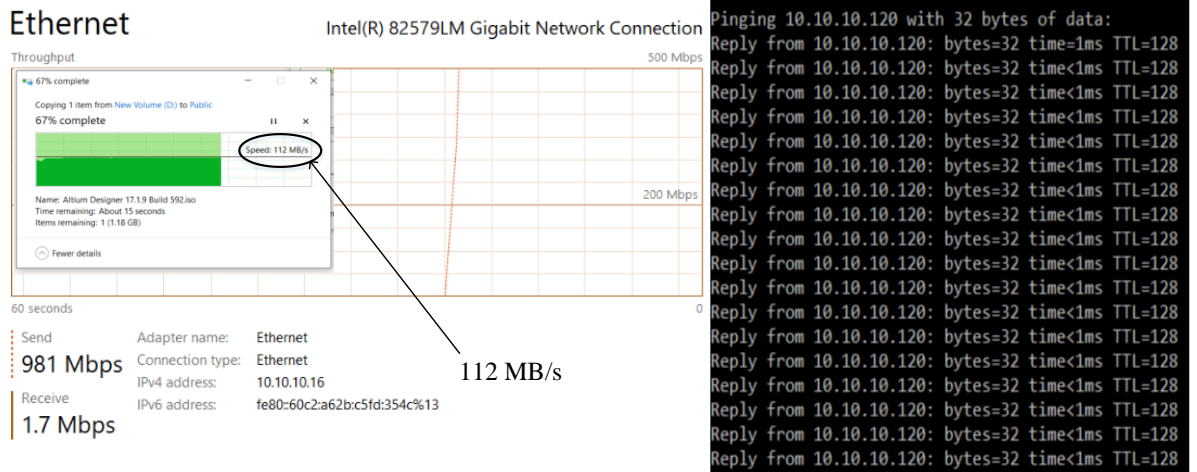


Hình 3.42 Sơ đồ kiểm tra bằng phương pháp nối chéo



Hình 3.43 Mô hình kiểm tra nối chéo trên thực tế

Kết quả thực tế thu được thể hiện ở Hình 3.44.



Hình 3.44 Kết quả sao chép dữ liệu giữa hai máy và ping giữa hai máy tính

Với các kết quả thu được, tốc độ truyền dữ liệu tốt nhất giữa hai máy khi kiểm tra với mô hình nối chéo là 981 Mbps, đạt 98.1 % tốc độ truyền dữ liệu tối đa 1 Gbps.

3.5 Kết luận chương 3

Chương 3 đã nêu ra mô hình điển hình cách giao tiếp giữa cổng vật lý và chip vật lý. Đồng thời cũng chỉ ra các chuẩn giao tiếp giữa MAC và PHY, để MAC có thể sử dụng với bất kỳ giao diện vật lý nào mà không phụ thuộc vào phương tiện truyền dẫn. Chương này cũng đưa ra sơ tổng quan hệ thống Switch, cách thiết kế mạch in cho các giao diện ngoại vi. Bên cạnh đó cũng kiểm tra, đánh giá mạch in thiết kế được dựa trên TEMAC IP Core (Tri-Mode Ethernet MAC Intellectual Property Core). Ở các chương tiếp theo sẽ trình bày các thiết kế bộ nhớ đệm, bộ nhớ CAM, TCAM và khối xử lý chuyển mạch.