BẤT ĐỒNG BỘ: MỘT TIẾP CẬN MỚI CHO VIỆC THIẾT KẾ SOC

ASYNCHRONOUS: A NEW APPROACH FOR SOC DESIGN

Đinh Đức Anh Vũ
Khoa Công Nghệ Thông Tin, Đại học Bách khoa, Tp. Hồ Chí Minh, Việt nam

BẢN TÓM TẮT

Mục đích của bài báo này là giới thiệu một lĩnh vực thiết kế các vi mạch tích hợp không dùng tín hiệu đồng bộ, tức là các vi mạch tích hợp thường được biết dưới cái tên "vi mạch bất đồng bộ" (BĐB). Bài viết không chủ ý giới thiệu chi tiết các vấn đề kỹ thuật, mà chỉ thu hút sự quan tâm của đọc giả bằng cách trình bày các nguyên lý hoạt động chính của vi mạch BĐB và triển vọng của việc ứng dụng nguyên lý BĐB này trong việc thiết kế SoC ngày nay. Phần tham khảo sẽ giúp các đọc giả quan tâm có thể tìm đọc thêm chi tiết.

ABSTRACT

The goal of this paper is to introduce the domain of un-clocked or clockless integrated circuits, i.e. what is now commonly known under the name of "asynchronous circuits". The purpose is not to address the field in details, but to draw reader's interest by presenting the main principles and some perspectives of asynchronous circuits on System-on-a-Chip (SoC) design today. The bibliography should enable the interested readers to go further into details.

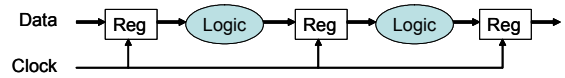
GIỚI THIỆU

Với độ phức tạp của vi mạch ngày càng cao và việc thiết kế vi mạch với chế độ hoạt động dựa trên xung đồng bộ truyền thống đang gặp phải những khó khăn nhất định, người ta đang đi tìm kiếm những giải pháp thay thế ([10]). Vi mạch hoạt động theo cơ chế bất đồng bộ (không có xung đồng bộ) là một trong những giải pháp cứu cánh cho những vấn đề đó ([10][11][12]). Bài báo nhằm giới thiệu tổng quan về vi mạch BĐB trong việc thiết kế vi mạch SoC.

Bài viết được tổ chức thành 2 phần. Phần 1 trình bày các nguyên tắc hoạt động chính của mạch BĐB. Dựa vào các nguyên tắc này, phần 2 sẽ giới thiệu các vấn đề mà loại vi mạch BĐB này có thể giải quyết trong việc thiết kế các SoC.

1. PHƯƠNG PHÁP THIẾT KẾ VI MẠCH BẤT ĐỒNG BỘ

Như được mô tả trong Hình 1, mạch đồng bộ được xây dựng từ các mạch tổ hợp và các thanh ghi. Hoạt động của mạch được điều khiển bởi một xung đồng bộ. Xung đồng bộ này kích hoạt đồng thời việc lưu trạng thái của mạch trong các thanh ghi. Khi một trạng thái mới xảy ra, mạch tổ hợp bắt đầu việc tính toán trạng thái kế tiếp cho chu kỳ đồng bộ lần sau. Điều này phù hợp với hình thức đặc tả mạch ở mức thanh ghi (RTL) mà ngày nay được dùng rất phổ biến. Việc thừa nhận mô hình này kéo theo 3 hệ quả quan trọng trong việc thiết kế vi mạch: i) logic tổ hợp đơn giản (do bỏ qua các hazard), ii) cơ chế giao tiếp giữa các thành phần trong mạch không có gì đặc biệt, và iii) một giả thiết về thời gian toàn cục phải được chú ý để bảo đảm mạch hoạt động: đường tổ hợp dài nhất (critical path) không được vượt quá một chu kỳ đồng bộ.

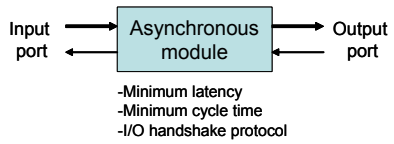


Hình 1 Mô hình cơ bản của mạch đồng bộ

Mạch BĐB không tuân theo mô hình này.

1.1. Giải phẫu một mạch BĐB

Vi mạch BĐB được xây dựng từ việc kết nối giao tiếp của các môđun BĐB. Sự hoạt động của môđun BĐB được điều khiển bởi sự hiện diện của dữ liệu tại ngõ nhập của nó (tương tự như mô hình dòng dữ liệu). Hình 2 mô tả một môđun BĐB trong đó dữ liệu được nhận từ ngõ nhập, được tính toán và sau đó được gởi ra ngõ xuất. Giao tiếp dữ liệu trên các ngõ I/O không được điều khiển bởi một tín hiệu bên ngoài giống như xung đồng bộ, mà được điều khiển bởi một *nghi thức giao tiếp* (giao thức) thực hiện ngay trong chính môđun đó. Giao thức này cần sự trao đổi thông tin 2 chiều giữa bên phát và bên thu, và được gọi là giao thức bắt tay. Giao thức là cơ sở của nguyên tắc hoạt động theo thứ tự trong mạch BĐB. Nguyên tắc đầu tiên là: một môđun chỉ bắt đầu hoạt động khi và chỉ khi tất cả dữ liệu cần thiết cho sự hoạt động đều sẵn sàng. Khi kết quả của sự hoạt động đã được lưu trữ, các dữ liệu ở ngõ nhập sẽ được giải phóng. Nguyên tắc thứ hai là: môđun gởi kết quả ra ngõ xuất khi và chỉ khi ngõ xuất này ở trạng thái sẵn sàng, nghĩa là nó đã được giải phóng ở cuối chu kỳ trao đổi thông tin lần trước đó. Việc hiện thực các giao thức trong vi mạch BĐB là cái giá phải trả cho sự hoạt động tự do không có xung đồng bộ và việc tự điều khiển cục bộ.



Hình 2 Đặc tính cơ bản của một môđun BĐB

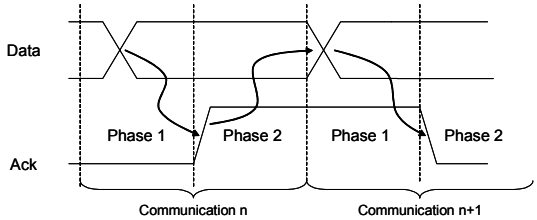
Môđun BĐB có thể thực hiện các chức năng ở các mức chi tiết khác nhau: chức năng ở mức bit, chức năng số học ở mức word và ngay cả các giải thuật phức tạp khác. Tuy nhiên, môđun BĐB phải luôn có 3 đặc điểm chính sau:

- Mọi kênh giao tiếp của môđun đều tuân theo một giao thức đồng nhất.
- Môđun có một *forward latency* tối thiểu, đó chính là thời gian tối thiểu cần thiết cho môđun để tính toán tập dữ liệu nhập và đặt kết quả ra ngõ xuất. Điểm khác biệt quan trọng với vi mạch đồng bộ là forward latency của vi mạch BĐB có thể thay đổi theo dữ liệu được tính toán hoặc theo số dữ liệu được lưu trữ trong pipeline của môđun (xem chi tiết thêm trong [1][2][3][4]).
- Môđun có một *thông lượng* (throughput) tối đa tương ứng với tần số tối đa mà môđun có thể xử lý dữ liệu vào. Chỉ số này thường được đặc trưng bằng *chu kỳ tối đa* (maximum cycle time) của môđun. Chỉ số này không phải là nghịch đảo của forward latency bởi vì môđun phải giải phóng kênh giao tiếp trước khi nhận tập dữ liệu mới (xem chi tiết thêm trong [4]).

Các bài báo [2]-[6] trình bày chi tiết các thông tin cần thiết để tối ưu hóa hiệu suất hoạt động của môđun BĐB.

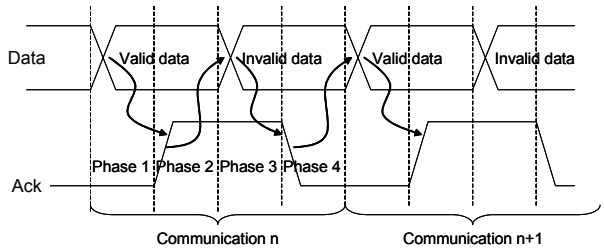
1.2. Giao thức

Có 2 loại giao thức chính: giao thức 2 bước và giao thức 4 bước. Hình 3 mô tả giao thức 2 bước: dữ liệu đến được phát hiện (bước 1) và xác nhận (acknowledgement – ack) được phát ra (bước 2). Sự phát hiện dữ liệu và việc tạo ra ack được thực hiện trên cạnh lên hoặc xuống của các tín hiệu, do đó cơ chế tín hiệu trong giao thức này dựa trên sự kiện (event-based).



Hình 3 Nghi thức giao tiếp 2 bước

Giao thức 4 bước dựa trên mức tín hiệu (level-based). Giao thức này đòi hỏi bước quay về không cho cả dữ liệu và ack (Hình 4). Dữ liệu được phát hiện khi giá trị dữ liệu thay đổi từ không hợp lệ (invalid) sang hợp lệ (valid) (bước 1). Sau đó, tín hiệu ack được bật sang mức *tích cực* (active) (bước 2). Đáp lại, giá trị dữ liệu được chuyển về trạng thái không hợp lệ (bước 3) và cuối cùng tín hiệu ack được đưa về mức không tích cực (inactive) (bước 4).



Hình 4 Nghi thức giao tiếp 4 bước

Giao thức nào tốt hơn? Thoạt nhìn sơ qua, giao thức 2 bước nhanh hơn và tiêu thụ công suất ít hơn. Tuy nhiên, điều này không hoàn toàn đúng bởi vì khi hiện thực vi mạch bằng công nghệ CMOS, logic dựa trên sự kiện tốn kém hơn logic dựa trên mức. Hơn nữa, với giao

thức 4 bước, người ta có thể thực hiện rất hiệu quả các quá trình tối ưu. Nói chung, giao thức 2 bước được dùng khi chu kỳ hoạt động của vi mạch bao gồm các thành phần chậm (có forward latency lớn), ví dụ, khi các tín hiệu đi vào/ra mạch thông qua các pad. Ngược lại, giao thức 4 bước được dùng khi các thành phần mạch được tự do sắp xếp (và do đó cân bằng các latency) để tối ưu hiệu suất của mạch.

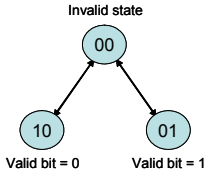
1.3. Cơ chế tín hiệu (Signaling)

Như mô tả trong phần trên, vi mạch BĐB hoạt động khi có sự hiện diện của dữ liệu và tín hiệu xác nhận (ack). Trong phần này, ta bàn đến cơ chế phát hiện dữ liệu và việc tạo ra tín hiệu ack. Việc phát hiện dữ liệu hợp lệ đồng nghĩa với một yêu cầu tính toán (request); và việc tạo ra tín hiệu ack đồng nghĩa với việc tính toán đã được hoàn tất và kênh giao tiếp có thể được giải phóng. Do đó, tín hiệu ack cũng thường được gọi là *tín hiệu hoàn tất* (completion signal).

Mã hóa dữ liệu/Request

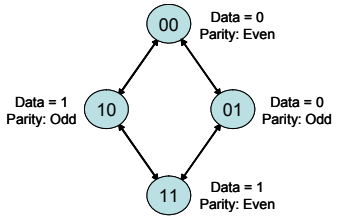
Cách đơn giản nhất để mã hóa dữ liệu của một kênh giao tiếp là cơ chế "*bó dữ liệu*" (bundled data). Trong cơ chế này, một tín hiệu request được đi kèm với dữ liệu để mang thông tin hợp lệ/không hợp lệ. Nó tương tự như việc tạo ra một tín hiệu đồng bộ đi kèm với mỗi kênh giao tiếp ([6][7]). Kỹ thuật này có thể dùng cho giao thức 2 bước hoặc 4 bước.

Cách phức tạp hơn là mã hóa thông tin hợp lệ/không hợp lệ vào trong dữ liệu. Giả sử một bit dữ liệu được truyền trên kênh giao tiếp. Nếu dùng giao thức 4 bước, trạng thái của kênh có thể là: không hợp lệ, hợp lệ với dữ liệu ở trạng thái logic '1', hợp lệ với dữ liệu ở trạng thái logic '0'. Do đó 2 bit (2 dây nối) sẽ được dùng để mã hóa trạng thái kênh. Kỹ thuật này gọi là mã hóa dual-rail (Hình 5).



Hình 5 Mã hóa dual-rail – 3 trạng thái

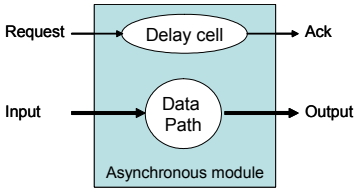
Nếu dùng giao thức 2 bước, ta phải cần 4 trạng thái. Vẫn giả sử rằng cần truyền 1 bit dữ liệu trên kênh giao tiếp, bên thu phải có khả năng phát hiện sự thay đổi của mã mỗi khi một bit dữ liệu mới được phát đi và ngay cả nếu giá trị của dữ liệu không thay đổi. Do đó 2 mã khác nhau cần được gán cho '0' và '1' như mô tả trong Hình 6 ([8]). Các môđun BĐB thực hiện máy trạng thái như Hình 6 trong mỗi kênh giao tiếp.



Hình 6 Mã hóa dual-rail – 4 trạng thái

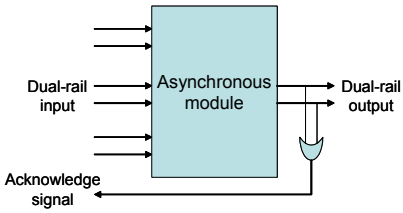
Tạo tín hiệu Ack/Hoàn tất

Kỹ thuật đơn giản đầu tiên là làm trễ tín hiệu request một khoảng thời gian bằng với thời gian cần thiết cho việc tính toán của môđun (matched-delay). Tín hiệu trễ này có thể được dùng thẳng như tín hiệu ack. Kỹ thuật này cơ bản dựa trên một giả thiết: thời gian trễ từ tín hiệu request đến tín hiệu ack bằng hoặc lớn hơn thời gian trễ trong *đường dữ liệu* (data path). Do đó trong thực tế, thời gian trễ được tính quá mức để bao hàm thời gian dò đường, sự biến thiên của quá trình chế tạo – điện áp – nhiệt độ. Hình 7 minh họa kỹ thuật này cho cơ chế bundled data.



Hình 7 Tạo tín hiệu Hoàn tất với kỹ thuật matched delay

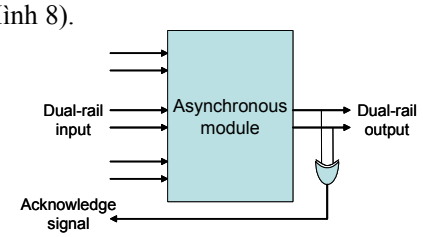
Một kỹ thuật khác dùng trong CMOS là cảm nhận dòng điện tiêu thụ của môđun để phát hiện sự kết thúc của quá trình tính toán ([9]). Người ta đưa ra nhiều loại mạch khác nhau, nhưng các mạch này đều không hiệu quả và rất khó thiết kế.



Hình 8 Tạo tín hiệu Hoàn tất với mã hóa dữ liệu 3 trạng thái

Kỹ thuật tốt nhất là dùng dữ liệu mã hóa. Xét một môđun được thiết kế cho phép xử lý dữ liệu được mã hóa theo 3 trạng thái và dùng giao thức 4 bước. Môđun này tạo ra ngõ xuất dual-rail mà

trạng thái dễ dàng nhận diện nhờ vào một cổng OR (Hình 8).



Hình 9 Tạo tín hiệu Hoàn tất với mã hóa dữ liệu 4 trạng thái

Hình 9 mô tả kỹ thuật tương tự khi 4 trạng thái được dùng để mã hóa một bit dữ liệu và giao thức là 2 bước.

1.4. Kết luận

Dựa vào các nguyên lý cơ bản được trình bày trong phần này, người ta thấy rằng vi mạch BĐB được xây dựng từ các *môđun đồng thời* (concurrent module), trao đổi thông tin 2 chiều theo một nghi thức bắt tay trên các kênh giao tiếp. Thực chất, tuần tự của toàn bộ hệ thống được thực hiện qua việc bắt tay giữa các môđun. Do đó, việc điều khiển của mạch BĐB được phân tán trên toàn các môđun. Hệ quả chính là ngoài việc thực hiện phép tính toán cần thiết, môđun phải tích hợp một phần cứng đặc thù cho việc phát hiện dữ liệu và tạo ra ack.

2. BẤT ĐỒNG BỘ ĐỐI VỚI THIẾT KẾ SOC

Với những đặc điểm được nêu trong phần trước, chúng tôi trình bày ở đây những triển vọng mà phương pháp thiết kế vi mạch bất đồng bộ mang lại cho việc thiết kế SoC trong tương lai, từ lĩnh vực công nghệ cấp thấp tới thiết kế cấp cao và phương pháp luận.

2.1. Biến thiên thời gian lan truyền

Tất cả các tham số công nghệ và sự lựa chọn cách hiện thực vật lý có thể dẫn đến sự biến thiên thời gian lan truyền trong các thiết bị và sự kết nối giữa các thiết bị này sẽ làm nảy sinh các vấn đề tiềm ẩn trong việc thiết kế vi mạch đồng bộ. Đối với các công nghệ tương lai, các dự báo Roadmap (SIA, EDA) đều nhận định các nhân tố gây nên sự biến thiên thời gian lan truyền của các tín hiệu là:

- građien nhiệt độ trong mạch
- luồng điện trong mạch nguồn
- sự lắp nối tạo ra những đường mạch song song (nhiễu xuyên kênh – crosstalk)

Phương pháp thiết kế vi mạch BĐB mang lại một giải pháp cho tất cả các hiện tượng vật lý và công nghệ làm ảnh hưởng đến sự biến thiên thời gian lan truyền.

2.2. Nhiễu

Vấn đề ảnh hưởng qua lại giữa các khối trong cùng một vi mạch là một vấn đề lớn trong thiết kế mà ngày càng gia tăng và nó gắn chặt với công nghệ. Một khối mạch số có thể gây ra nhiễu khi đặt dưới điện áp nguồn hoặc cho các dòng điện đi qua. Nhiễu này có thể ảnh hưởng đến đặc tính của các khối analog bên cạnh ngay cả khi chúng dùng các giếng (caisson) khác nhau. Một nguồn nhiễu khác là sự phát xạ điện từ của các mạch có xung đồng bộ. Các mạch này có thể làm nhiễu các khối RF của hệ thống. Tuy nhiên, với những tần số hoạt động có thể có trong tương lai, các hài tần (bội số của tần số hoạt động) của xung đồng bộ có thể đạt đến vài GHz. Ở tần số này, chính các đường mạch sẽ là các anten. Giao thoa điện từ giữa các anten này sẽ trở nên rất quan trọng trong tương lai.

Với sự vắng mặt của xung đồng bộ, vi mạch BĐB ít gây nhiễu hơn vi mạch đồng bộ. Công suất phát xạ phụ thuộc vào cường độ và hình dạng của dòng điện tiêu thụ trong mạch. Xung đồng bộ thay đổi theo dòng điện và tạo ra các gai nhọn trong phổ của dòng điện tại các tần số hài tần. Sự thay đổi điện áp tỉ lệ thuận với đạo hàm của dòng điện. Do đó chúng ta dễ dàng hiểu rằng sự phân tán công suất tiêu thụ theo thời gian đóng vai trò tích cực trong việc giảm nhiễu trong mạch nguồn và giảm nhiễu xạ điện từ. Đó là đặc tính quan trọng của vi mạch BĐB nhờ vào sự thiếu vắng xung đồng bộ và sự thay đổi thời gian tính toán tùy theo dữ liệu. Thực ra, ngay khi cả một chương trình có tính chu kỳ, thời gian tính toán cũng thay đổi từ lần thực hiện này đến lần thực hiện khác. Điều này hạn chế việc tạo ra nhiễu.

2.3. Công suất tiêu thụ

Giảm công suất tiêu thụ cũng tác dụng làm giảm nhiễu. Chúng ta đã biết là vi mạch bất đồng bộ chỉ tiêu thụ công suất khi nó thực hiện tác vụ tính toán và chỉ có các thành phần liên quan trong vi mạch tiêu thụ công suất. Tận dụng cơ chế vi mạch BĐB để giảm công suất tiêu thụ phụ thuộc vào từng loại vi mạch. Tất cả các hệ thống xử lý dòng dữ liệu bất thường, nếu được thiết kế theo kiểu dòng dữ liệu của vi mạch BĐB, đều có thể giảm được công suất tiêu thụ. Chế độ nghỉ (standby) là chế độ mặc định của vi mạch BĐB và việc khởi động lại khi có nhu cầu tính toán được thực hiện rất nhanh. Việc giảm

sát công suất tiêu thụ được thực hiện bằng phần cứng ở các cấp độ kiến trúc khác nhau, không cần thiết phải thiết kế một phần mềm (đôi khi rất phức tạp) để điều khiển việc kích hoạt hoặc chuyển sang chế độ nghỉ một phần của hệ thống. Do đó thời gian thiết kế vi mạch được giảm bớt. Dưới góc độ ứng dụng, người ta thường hay nói đến các hệ thống giám sát khi muốn tận dụng đặc tính này.

Sự phát triển công nghệ dẫn đến sự gia tăng tương đối của công suất tiêu thụ tĩnh so với công suất tiêu thụ động. Để khắc phục nhược điểm này, các công nghệ thường tích hợp các transistor với các điện áp ngưỡng khác nhau. Người thiết kế có thể chọn lựa 2 lớp transistor khác nhau: hoặc là transistor tốc độ cao với sự rò rỉ dòng điện tĩnh hoặc transistor tốc độ chậm hơn nhưng công suất tiêu thụ tĩnh gần như bằng không. Việc kết hợp 2 loại transistor này để thiết kế được vi mạch có tốc độ cao và công suất tiêu thấp (gần như bằng 0) không phải dễ dàng. Theo ý kiến chúng tôi, sẽ rất đáng để nghiên cứu trong chừng mực nào thì nghi thức bắt tay bằng yêu cầu/xác nhận trong vi mạch BĐB được dùng để điều khiển các transistor có điện áp ngưỡng cao. Thực ra, ta có thể tưởng tượng ra việc điều khiển sự kích hoạt luận lý/điện của các tế bào hoặc các khối bằng các tín hiệu yêu cầu/xác nhận. Kỹ thuật này cho phép hạn chế công suất tiêu thụ tĩnh, bằng cách chỉ tận dụng hiệu quả về mặt tốc độ của công nghệ.

2.4. Đồng bộ hóa

Một ảnh hưởng khác của sự phát triển công nghệ là sự gia tăng quan hệ của thời gian trễ trong các kết nối đối với thời gian trễ trong các cổng. Sự khác biệt ngày càng lớn này dẫn đến phải chỉnh sửa sơ đồ đồng bộ hóa các thành phần trong mạch. Dự báo roadmap SIA đưa ra một đề nghị rằng các vi mạch phải được cấu tạo từ các khối chức năng hoạt động với các xung đồng bộ cục bộ. Đó chính là khái niệm vi mạch loại GALS (Globally Asynchronous – Locally Synchronous). Việc trao đổi thông tin giữa các khối thông qua các tuyến (bus) truyền thông dài không thể tự thực hiện được theo tần số cục bộ của các khối chức năng. Do đó người ta dùng những kỹ thuật và những nghi thức đồng bộ cho phép bảo đảm việc trao đổi giữa các khối có các xung đồng bộ khác nhau với những chi phí thấp nhất. Việc thiết kế các giao tiếp như vậy giữa các khối đồng bộ là một điểm mạnh của vi mạch BĐB.

2.5. Tái sử dụng

Việc ứng dụng mô hình kiến trúc GALS ngày càng nhiều hơn nhờ vào việc sử dụng gia tăng các khối chức năng IP được thiết kế bởi các nhóm/tổ chức khác nhau. Việc thiết kế vi mạch ngày nay càng tiến dần đến việc lắp ghép các khối chức năng (được thiết kế) từ nhiều nguồn khác nhau, có bản chất khác nhau, hiện thực các qui ước khác nhau về việc đồng bộ hóa cục bộ và nghi thức giao tiếp: bộ xử lý, vi điều khiển, DSP, bộ nhớ, bộ mã hóa/giải mã video và âm thanh, modem, bộ phát/thu RF, giao tiếp truyền thông với bên ngoài (PCI, I^2C, USB, …)

Do đó, vấn đề giao tiếp và đồng bộ hóa giữa các khối chức năng phải được giải quyết thông qua việc dùng bus, fifo, router, bộ nhớ nhiều cổng, …. Những bài toán phức tạp sẽ gặp phải là bài toán "trọng tài" (arbiter), bài toán loại trừ tương hỗ (mutual exclusion) và bài toán chia xẻ tài nguyên. Cơ chế BĐB cho phép giải quyết các vấn đề này và đưa ra các giải pháp hiện thực dễ dàng và nhanh chóng. Trong giai đoạn ngắn hạn, người ta nên nghiên cứu, thiết kế các hệ thống hỗn hợp đồng bộ-BĐB. Một khi quá trình trao đổi giữa các khối đồng bộ được hiện thực theo cơ chế BĐB, người ta sẽ rất dễ dàng trong giai đoạn trung hạn, thậm chí dài hạn, đưa vào các khối BĐB. Phương pháp thiết kế kiến trúc này được cho là sẽ giảm đáng kể thời gian thiết kế của những hệ thống phức tạp.

2.6. Công cụ thiết kế

Việc ứng dụng công nghệ BĐB vào các thiết kế trong công nghiệp sẽ khả thi với sự hỗ trợ của các công cụ trợ giúp thiết kế (tổng hợp, kiểm tra). Ngay cả nếu các công cụ tổng hợp các bộ điều khiển có tồn tại chăng nữa, độ phức tạp của các thiết kế ngày nay đòi hỏi vượt xa khả năng đáp ứng của các công cụ này. Theo chúng tôi, vấn đề này nên được tập trung giải quyết từ mức độ trừu tượng cao: tổng hợp cấp cao. Những ngôn ngữ đặc tả phần cứng đã ra đời khá lâu, nhưng những ngôn ngữ đặc tả kiến trúc và hệ thống mới xuất hiện. Người ta tin rằng quá trình tổng hợp hành vi (behavioral synthesis) sẽ là bước đi kế tiếp và sự tổng hợp hệ thống BĐB từ các ngôn ngữ cấp cao như SystemC sẽ được nghiên cứu nghiêm túc.

Hiện tại, cộng đồng nghiên cứu đang tập trung nghiên cứu về qui trình thiết kế vi mạch BĐB với sự tham gia của các công cụ thiết kế của vi mạch đồng bộ. Cụ thể, người ta phân tách đặc tả hành vi (được mô tả dùng các ngôn ngữ

đặc tả phần cứng HDL hoặc C) thành 2 phần: i) phần sơ đồ dòng dữ liệu được tổng hợp với các công cụ thiết kế của vi mạch đồng bộ và ii) phần điều khiển được tổng hợp bằng công cụ thiết kế đặc biệt. Do đó gánh nặng bây giờ đặt lên việc phát triển những công cụ thiết kế đặc biệt sao cho chúng mạnh và hiệu quả.

Tính mođun và sự điều khiển cục bộ của các vi mạch BĐB hỗ trợ rất nhiều cho các công cụ tổng hợp hành vi. Cũng giống như các công cụ tổng hợp vi mạch đồng bộ, quá trình tối ưu hóa kiểu "retiming" cũng có thể áp dụng cho vi mạch BĐB. Quá trình tối ưu hóa này tương ứng với việc tích hợp trong công cụ những đường ống (pipeline) bất đồng bộ. Hiện tại chưa có công cụ nào thực hiện được điều này, nhưng chúng tôi phỏng đoán rằng đây là vấn đề ưu tiên phát triển của các công cụ CAD.

Đến một lúc nào đó, người thiết kế phải có các phương tiện để so sánh các hiện thực đồng bộ và BĐB. Anh ta sẽ phải chọn lựa: đồng bộ hoặc BĐB. Anh ta cũng phải được trang bị các phương tiện để thiết kế giao tiếp giữa các khối đồng bộ và bất đồng bộ bằng cách dùng hay không các nghi thức giao tiếp chuẩn.

Theo chúng tôi, những công cụ thiết kế vi mạch BĐB phải thoả các mục tiêu sau: i) ngày càng tương thích với các công cụ thương mại, ii) tương thích với các ngôn ngữ đặc tả phần cứng (HDL hoặc các ngôn ngữ giống như C), iii) dễ dàng lắp ghép các khối đồng bộ và BĐB, iv) thuận lợi trong việc thiết kế giao tiếp giữa các khối đồng bộ/BĐB.

Sau quá trình tổng hợp, thư viện phải là vấn đề nhận được sự quan tâm thích đáng. Người ta đã minh chứng rằng vi mạch BĐB có thể được thiết kế chỉ với thư viện chuẩn. Tuy nhiên do những thư viện này được tối ưu cho các vi mạch đồng bộ, do đó có một số thành phần không bao giờ được dùng. Việc thêm vào các thành phần cổng Muller, cổng loại trừ tương hỗ (ME), cổng đồng bộ hóa vào thư viện cho phép giảm kích thước, công suất tiêu thụ và tăng tốc độ của vi mạch.

Điểm quan trọng cuối cùng liên quan đến các công cụ thiết kế là việc phát triển các giải thuật tạo ra các mẫu thử. Quá trình thiết kế dưới góc độ test phải hội tụ đến một kỹ thuật đồng nhất và tổng quát để kiểm tra toàn bộ vi mạch BĐB. Việc tiến đến các chuẩn hiện tại cho việc kiểm tra phải được tiến hành đồng thời với việc phát triển thư viện các tế bào chuẩn. Còn rất nhiều việc phải làm trong lĩnh vực này.

KẾT LUẬN

Người ta cho thấy đã có những vi mạch BĐB với khả năng và độ phức tạp cao được thiết kế và chế tạo. Mặc dù thiếu các công cụ CAD, các nhóm nghiên cứu từ các trường đại học và các công ty đã thành công trong việc phát triển những mẫu thử nghiệm mang ý nghĩa to lớn đối với công nghệ BĐB. Rất khó để có thể so sánh các thiết kế được thương mại hóa, nhưng theo ý kiến chúng tôi, các thiết kế thử nghiệm này chứng tỏ rằng công nghệ BĐB là một công nghệ hứa hẹn khắc phục khoảng cách giữa công nghệ chế tạo và thiết kế trong tương lai.

Đối với triển vọng của phương pháp thiết kế SoC, chúng tôi tin rằng BĐB sẽ là một công nghệ then chốt, đáp ứng các vấn đề như: công suất nhỏ, ít gây nhiễu, phân tán xung đồng bộ, bus on-chip và các thiết kế *hỗn hợp tín hiệu* (mixed-signal).

Nếu chúng ta nhìn vào xu thế công nghệ trong tương lai, ta có thể tiên đoán rằng các nhà thiết kế sẽ phải đương đầu với sự gia tăng của nhiều loại thời gian trễ trên cổng và dây nối. Trong thực tế, với công nghệ chế tạo ngày càng tiên tiến (kích thước vi mạch ngày càng nhỏ), sự biến thiên của các quá trình chế tạo, cộng hưởng dòng điện, građien nhiệt độ, … sẽ ảnh hưởng đến thời gian trễ. Việc trơ với thời gian trễ của mạch BĐB có thể giải quyết các vấn đề này.

TÀI LIỆU THAM KHẢO

[1] Uri Cummings, Andrew Lines, and Alain Martin. An asynchronous pipelined lattice structure filter. In *Proceedings of ASYNC*, pp 126-133, November 1994.

[2] Marc Renaudin, Bachar El Hassan, and Alain Guyot. New asynchronous pipeline scheme: Application to the design of a self-timed ring divider. *IEEE Journal of Solid-State Circuits*, 31(7):1001-1013, July 1996.

[3] Ted E. Williams. *Self-Timed Rings and their Application to Division*. PhD thesis, Stanford University, June 1991.

[4] Ted E. Williams. Performance of iterative computation in self-timed rings. *Journal of VLSI Signal Processing*, 7(1/2):17-31, February 1994.

[5] Ted E. Williams and Mark A. Horowitz. A zero-overhead self-timed 160ns 54b CMOS divider. *IEEE Journal of Solid-State Circuits*, 26(11):1651-1661, November 1991.

[6] K. Y. Yun, P. A. Beerel, and J. Arceo. High-performance asynchronous pipeline circuits. In *Proceedings of ASYNC*, March 1996.

[7] Ivan E. Sutherland. Micropipelines. *Communications of the ACM*, 32(6):720-738, June 1989.

[8] Anthony J. McAuley. Four state asynchronous architectures. *IEEE Transactions on Computers*, 41(2):129-142, February 1992.

[9] Mark E. Dean, David L. Dill, and Mark Horowitz. Self-timed logic using current-sensing completion detection (CSCD). *Journal of VLSI Signal Processing*, 7(1/2):7-16, February 1994.

[10] Anh-Vu Dinh-Duc, A New Style in ASIC Design with Asynchronous Circuits. *In Proceedings of the second National Symposium on Research, Development and Application of Information and Communication Technology*, Sept. 24-25, 2005, Hanoi, Vietnam.

[11] Scott Hauck. Asynchronous design methodologies: An overview. *Proceedings of the IEEE*, 83(1):69-93, January 1995.

[12] Janusz A. Brzozowski and Carl-Johan H. Seger. *Asynchronous Circuits*. Springer-Verlag, 1995.