

THIẾT KẾ BỘ NHỚ SRAM 32KB KẾT HỢP KỸ THUẬT DỰ TRỮ HÀNG VÀ CỘT

A DESIGN OF 32KB SRAM COMBINING ROW AND COLUMN REDUNDANCY

Võ Thanh Trí, Lê Bình Sơn, Bùi Trọng Tú

Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Tp. HCM
 {vttri, lbson, btuu}@fetel.hcmus.edu.vn

Tóm tắt - Trong bài báo này, chúng tôi sẽ trình bày một thiết kế bộ nhớ SRAM đồng bộ có dung lượng 32 KB sử dụng kiến trúc tế bào nhớ 6T và công nghệ CMOS UMC 90 nm. Mục tiêu chính của thiết kế là tối ưu công suất tiêu thụ, tần số hoạt động, độ ổn định và khả năng đạt được sản lượng cao sau khi chế tạo. Do đó, các kỹ thuật tự cân chỉnh thời gian, giải mã đa tầng, cơ chế đa hợp và dự trữ hàng/cột đã được chúng tôi kết hợp sử dụng trong thiết kế. Các kết quả mô phỏng cho thấy bộ nhớ có thời gian truy cập dao động trong khoảng 0,66 ns đến 2,15 ns và tần số hoạt động từ 389 MHz đến 2,6 GHz tương ứng với các góc công nghệ SS và FF. Dòng rỉ và công suất tiêu thụ lớn nhất của mạch tương ứng là 18,24 μ A và 4,03 μ W/MHz.

Từ khóa - thiết kế; SRAM; CMOS; kỹ thuật dự trữ; 6T.

Abstract - In this paper, we present a design of a 32 KB synchronous SRAM module using 6T memory cell structure, implemented in 90 nm CMOS UMC technology. The goal of the proposed design is to achieve the highest performance in terms of power consumption, operating frequency, stability and the yield after production. Therefore, in this design, techniques such as self-time tracking, pre-decoder, multiplexer, and row/column redundancy are combined. The simulation results show that the access time varies between 0.66 ns and 2.15 ns and the operating frequency goes from 389 MHz to 2.6 GHz at SS and FF corners respectively. The highest leakage current and power dissipation are 18.24 μ A and 4.03 μ W/MHz, respectively.

Key words - design; SRAM; CMOS; redundancy; 6T.

1. Đặt vấn đề

Theo ITRS [1], bộ nhớ nhúng SRAM chiếm khoảng 86% tổng số lượng transistor trên vi xử lý. Điều này cho thấy độ ổn định của SRAM ảnh hưởng rất lớn đến độ ổn định của hệ thống. Với sự phát triển của công nghệ CMOS, mật độ tích hợp của bộ nhớ SRAM ngày càng cao, hoạt động nhanh hơn và công suất tiêu thụ thấp hơn. Tuy nhiên, độ ổn định của tế bào nhớ SRAM bị suy giảm do ảnh hưởng của việc hạ thấp điện thế hoạt động. Ngoài ra, trong quá trình chế tạo, các lỗi có thể phát sinh và làm cho chip bị sai lệch so với thiết kế ban đầu và dẫn đến các sản phẩm không đạt yêu cầu. Các lỗi thường gặp như đường dây nối giữa các transistor bị hở (open) hay bị ngắn mạch (short), các lỗi trong quá trình in khắc quang, thay đổi về nồng độ pha tạp, sai lệch về thế ngưỡng giữa các cặp transistor vi sai...

Với bộ nhớ SRAM, do số lượng tế bào nhớ rất lớn, nên xác suất phát sinh lỗi rất cao. Ví dụ chỉ cần 1 tế bào nhớ trong 32 KB bị hỏng thì cũng đồng nghĩa với việc toàn bộ hoạt động của chip sẽ bị ảnh hưởng. Đặc biệt, trong các kiến trúc vi xử lý ngày nay, các bộ nhớ đệm L3 có thể đạt mật độ lên tới 6 MB. Mặt khác, khi công nghệ CMOS ngày càng phát triển, kích thước transistor ngày càng nhỏ thì lỗi phát sinh trên chip lại ngày càng lớn. Do đó, để tăng tính khả dụng cho bộ nhớ, kỹ thuật dự trữ (Redundancy) [7], [8] đã bắt đầu được sử dụng rộng rãi trong công nghiệp trong thời gian gần đây. Theo phân tích và thống kê của IBM [2], với kỹ thuật này ta có thể tăng tính khả dụng sau khi được chế tạo lên trên 90%.

Nguyên tắc chính của phương pháp dự trữ là thay thế hàng hoặc cột chứa tế bào nhớ bị lỗi bởi các hàng hoặc cột dự trữ. Nếu yêu cầu về sản lượng càng cao thì ta càng phải dự trữ một số lượng hàng/cột càng lớn, nên diện tích và công suất tiêu thụ của chip sẽ tăng. Công việc thiết kế cũng trở nên phức tạp hơn và giá thành sản phẩm cũng bị đội lên. Do đó, việc kết hợp kỹ thuật này vào thiết kế SRAM sao cho tối ưu về mặt công suất, diện tích, và tốc độ đóng vai

trò hết sức quan trọng.

Trong bài báo này, chúng tôi đề xuất một thiết kế bộ nhớ SRAM đồng bộ 32 KB sử dụng kiến trúc tế bào nhớ 6T trên công nghệ UMC 90 nm, với mục đích ứng dụng vào các bộ vi xử lý và vi điều khiển. Để kiến trúc đạt được hiệu suất cao, các kỹ thuật như tự cân chỉnh thời gian, giải mã đa tầng, và đa hợp cột [3] sẽ được kết hợp và điều chỉnh để đáp ứng các đặc tính kỹ thuật của thiết kế. Đặc biệt, trong thiết kế này cả hai kỹ thuật dự trữ hàng và dự trữ cột đều được ứng dụng nhằm tăng sản lượng chip sau khi chế tạo nhưng vẫn đạt được yêu cầu về tốc độ và công suất.

Các kết quả mô phỏng cho thấy bộ nhớ có thời gian truy cập dao động từ 0,66 ns đến 2,15 ns và tần số hoạt động từ 389 MHz đến 2,6 GHz. Dòng rỉ và công suất tiêu thụ lớn nhất tương ứng là 18,24 μ A và 4,03 μ W/MHz. So sánh kết quả đạt được với các công trình khác cho thấy thiết kế này có thời gian truy xuất xấp xỉ, nhưng có ưu điểm hơn về công suất tiêu thụ.

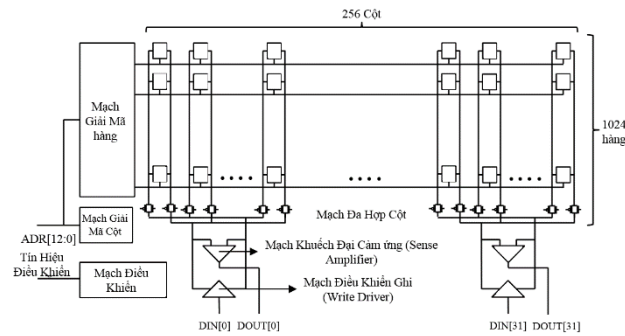
Kỹ thuật dự trữ hàng/cột sẽ được miêu tả vắn tắt trong mục 2 của bài báo. Ở mục 3 chúng tôi sẽ trình bày thiết kế mảng nhớ 32 KB và các kỹ thuật được sử dụng. Kết quả mô phỏng và so sánh với các công trình khác được trình bày trong mục 4. Cuối cùng, các kết luận sẽ được đưa ra trong mục 5.

2. Kỹ thuật dự trữ hàng/cột

2.1. Kỹ thuật dự trữ hàng

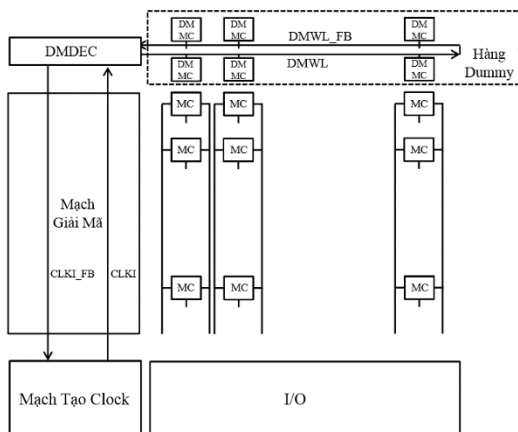
Nguyên tắc cơ bản của dự trữ hàng là thay thế hàng có tế bào nhớ (bitcell) bị lỗi bởi một hàng dự trữ khác như được mô tả ở Hình 1. Sau khi chế tạo xong, các mạch Built-In-Self-Test (BIST) sẽ được sử dụng để kiểm tra hoạt động của SRAM. Địa chỉ của hàng bị lỗi sẽ được lưu trữ vào thanh ghi hoặc một bộ nhớ không bay hơi. Khi bộ nhớ hoạt động, địa chỉ đưa vào bộ nhớ sẽ được so sánh với địa chỉ lưu trong thanh ghi. Nếu hai địa chỉ khác nhau thì bộ nhớ sẽ đọc/ghi dữ liệu một cách bình thường. Nếu hai địa chỉ

SRAM 6T đã được thiết kế tối ưu về tốc độ, diện tích và công suất. Khối điều khiển (controller) sẽ nhận các tín hiệu điều khiển từ hệ thống để tạo ra các tín hiệu nội, điều khiển các khối khác hoạt động nhịp nhàng với nhau. Mạch giải mã hàng và mạch giải mã cột sẽ tiến hành giải mã địa chỉ để chọn hàng và cột tương ứng. Sẽ có tất cả 13 bit địa chỉ. Trong suốt quá trình đọc/ghi, địa chỉ phải được giữ giá trị. Do đó, các mạch chốt địa chỉ được sử dụng. Trong quá trình ghi, các bộ “write driver” sẽ ghi dữ liệu vào bộ nhớ. Trong quá trình đọc, mạch khuếch đại cảm ứng (Sense Amplifier) sẽ khuếch đại sự sai biệt hiệu điện thế trên hai đường bitline, giúp tăng tốc quá trình đọc.



Hình 4. Các khối cơ bản trong thiết kế

3.2. Kỹ thuật tự căn chỉnh thời gian

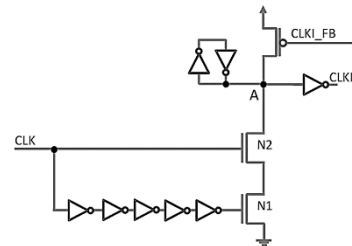


Hình 5. Kỹ thuật self-time với hàng bitcell dùng như tải trên đường wordline

Như đã trình bày, bộ nhớ được thiết kế để hoạt động theo cơ chế đồng bộ với xung clock của hệ thống. Trong đó, mạch điều khiển có chức năng chốt dữ liệu được truyền vào, đồng thời dựa vào xung clock của hệ thống tạo các xung clock nội cho hoạt động của toàn bộ nhớ. Do đó, kỹ thuật tự căn chỉnh thời gian (Self Time Tracking) [3] cần được áp dụng để điều khiển các khối hoạt động nhịp nhàng với nhau. Ví dụ như tạo ra xung clock có độ rộng vừa đủ để mở wordline, tắt/mở pre-charge, tắt/mở sense amplifier, tắt/mở các bộ chốt. Thông thường, bộ tạo xung sẽ dùng cổng AND và cổng đảo. Tuy nhiên, nhược điểm là các bộ tạo xung này chịu ảnh hưởng nhiều bởi sự thay đổi của điện thế hoạt động, nhiệt độ và sai số trong quá trình chế tạo. Độ rộng xung tạo ra có thể không đủ để các khối hoạt động, dẫn đến bộ nhớ hoạt động không chính xác.

Để khắc phục nhược điểm này, chúng tôi sử dụng kỹ thuật tự căn chỉnh thời gian như được mô tả trong Hình 5.

Hai hàng bitcell phụ được thêm vào ở cuối mảng tế bào nhớ (dummy row). Chúng đóng vai trò như tải cho đường wordline. Khi có cạnh lên của xung clock ngoài, bộ tạo xung clock nội sẽ tạo ra xung CLKI (CLKI sẽ chuyển từ thấp lên cao). Xung CLKI này sẽ chạy đến bitcell cuối cùng của hàng dummy và quay trở về bộ tạo xung clock nội để tắt xung CLKI (chuyển từ cao xuống thấp). Kỹ thuật này đảm bảo độ rộng xung CLKI luôn đủ để bộ nhớ hoạt động, bất chấp ảnh hưởng của điều kiện hoạt động.



Hình 6. Mạch tạo xung clock

Hình 6 mô tả hoạt động của mạch tạo xung clock. Trong đó, 2 cổng inverter có vai trò như một mạch giữ yếu (weak keeper) để giữ nút “A” ở mức 1 hoặc 0. Tín hiệu CLKI_FB là đảo của tín hiệu CLKI sau khi đi qua mạch giải mã, hàng dummy và trở về mạch tạo xung clock. Ở trạng thái chưa có lệnh truy xuất, tín hiệu CLK sẽ ở mức thấp. Nút “A” được preset ở mức cao, nên tín hiệu CLKI cũng sẽ ở mức thấp và CLKI_FB ở mức cao. Khi CLK chuyển từ thấp lên cao, nút “A” sẽ bị kéo xuống mức thấp do 2 cổng inverter được thiết kế để dẫn yếu hơn 2 transistor NMOS N1 và N2. Do đó, CLKI được kéo lên cao. Lúc này tín hiệu CLKI_FB vẫn đang ở mức cao. Sau khi tín hiệu CLKI đã đi qua hàng dummy và trở lại mạch tạo xung clock, tín hiệu CLKI_FB sẽ được kéo xuống thấp. Lúc này nút “A” sẽ được kéo lên mức cao nhờ PMOS dẫn mạnh hơn hai cổng đảo. Cuối cùng, CLKI cũng sẽ được kéo về mức thấp và CLKI_FB trở lại mức cao để đợi lệnh truy xuất tiếp theo từ hệ thống. Theo cách này, xung CLKI được tạo ra có độ rộng đủ lớn, vừa đảm bảo đủ thời gian truy xuất tế bào nhớ ở vị trí xa nhất, vừa tối ưu thời gian đóng mở các đường wordline nhằm tiết giảm công suất tiêu thụ.

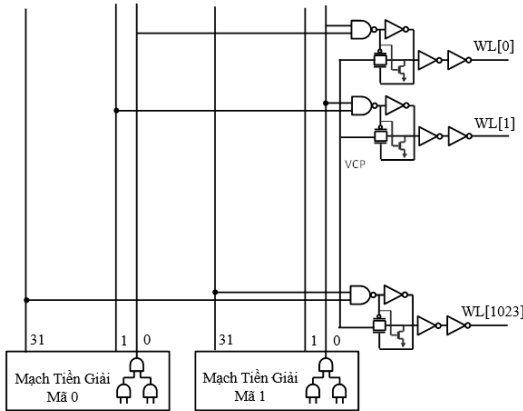
3.3. Giải mã địa chỉ

Trong quá trình đọc/ghi bộ nhớ, chỉ có một hàng tế bào nhớ được chọn. Do đó, chúng ta cần đến bộ giải mã địa chỉ. Chức năng chính của nó là giải mã N bit địa chỉ thành 2^N đường. Mỗi đường sẽ điều khiển việc đóng hay mở một hàng hay cột.

Với kiến trúc thông thường thì bộ giải mã phải cần đến nhiều cổng AND ghép tầng với nhau. Cách ghép như vậy sẽ làm tăng đáng kể thời gian trì hoãn. Để giải quyết vấn đề này, chúng tôi sử dụng kiến trúc giải mã hai tầng như mô tả ở Hình 7. Tầng thứ nhất là tầng tiền giải mã (pre-decoder). SRAM sẽ dùng nhiều bộ tiền giải mã và thường là giải mã từ một sang hai, hai sang bốn, ba sang tám, bốn sang mười sáu đường,... Ngõ ra của các bộ tiền giải mã này sẽ đi đến tầng giải mã thứ hai. Tầng này sẽ tổ hợp các ngõ vào theo nguyên tắc các ngõ ra của bộ tiền giải mã thứ nhất, sẽ AND với lần lượt ngõ ra của bộ tiền giải mã thứ hai và cứ thế cho đến hết.

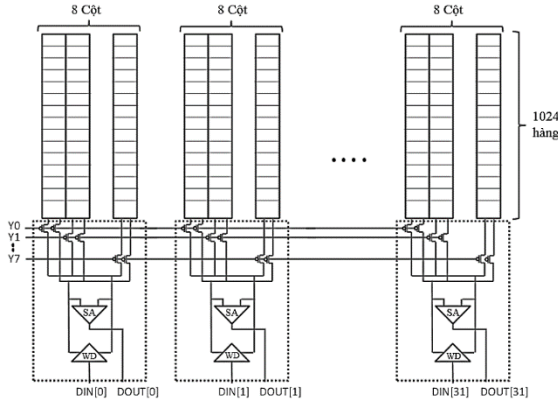
Ngõ ra của tầng thứ hai sẽ điều khiển TGATE để cho xung CLKI đi qua và mở wordline. Tại một thời điểm, chỉ

có 1 wordline được mở, các wordline khác sẽ được giữ ở mức 0 nhờ NMOS kéo xuống. Mỗi wordline sẽ nối với 256 tế bào nhớ (128 tế bào bên trái và 128 tế bào bên phải) nên sẽ có fanout lớn. Vì vậy các bộ đệm tín hiệu (mạch inverter) sẽ được sử dụng thêm để tăng dòng lái toàn bộ wordline.



Hình 7. Kiến trúc mạch giải mã địa chỉ hai tầng.

3.4. Cơ chế đa hợp cột



Hình 8. Kiến trúc mạch giải mã địa chỉ hai tầng

Với cơ chế đa hợp cột ta có thể giảm số lượng tế bào nhớ trên mỗi đường bitline, từ đó giảm điện trở và tụ điện ký sinh trên đường dây. Điều này giúp việc đọc/ghi diễn ra nhanh hơn và tiết kiệm công suất tiêu thụ hơn. Ngoài ra, việc sử dụng kỹ thuật đa hợp cột giúp cho kích thước toàn bộ bộ nhớ cân xứng hơn.

Trong thiết kế này, chúng tôi sử dụng bộ đa hợp 8 sang

1 như được trình bày ở Hình 8. Mảng tế bào nhớ được chia thành 32 khối nhỏ, mỗi khối có 8 cột, mỗi cột có 1024 bitcell. Mỗi khối sẽ gắn với một mạch I/O, tương ứng với một bit dữ liệu. 3 bit địa chỉ được sử dụng để tạo ra các tín hiệu chọn cột (Y0 → Y7). Các tín hiệu này được nối với các pass transistor để chọn ra cột cần được truy xuất. Ví dụ, nếu địa chỉ truy xuất là 000 thì 32 mạch I/O (DIN/OUT[0 → 31]) sẽ được nối tới cột đầu tiên của mỗi khối bộ nhớ tương ứng.

4. Kết quả mô phỏng

4.1. Đánh giá đặc tính của bộ nhớ SRAM được thiết kế

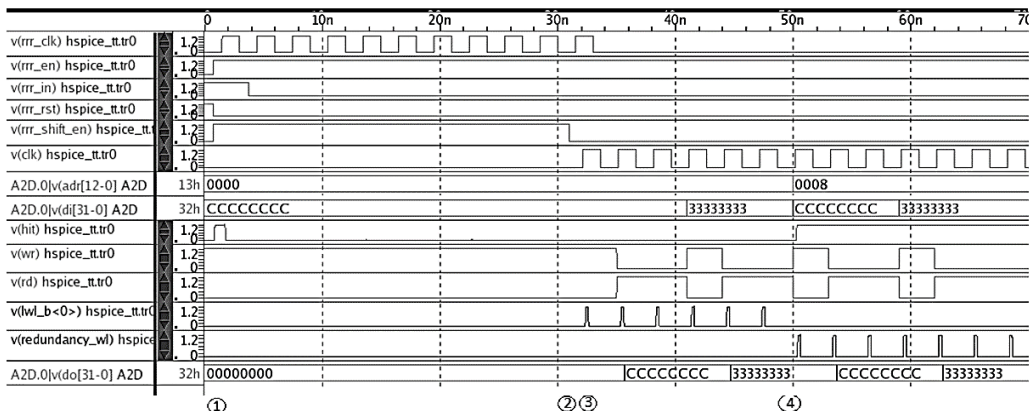
Bảng 1. Mô tả các điều kiện hoạt động khi mô phỏng

Góc	Điện thế	Nhiệt độ	Mô tả
FF	1.32	-45	Điều kiện hoạt động tốt nhất
TT	1.2	25	Điều kiện hoạt động thông thường
SS	1.08	125	Điều kiện hoạt động xấu nhất

Bảng 2. Kết quả mô phỏng của bộ nhớ được thiết kế

Tham số		FF	TT	SS
Thời gian truy xuất	Đọc 1 ở địa chỉ gần nhất (ns)	0,66	1,07	2,15
	Đọc 1 ở địa chỉ xa nhất (ns)	0,66	1,07	2,15
	Đọc 0 ở địa chỉ gần nhất (ns)	0,61	0,933	1,99
	Đọc 0 ở địa chỉ xa nhất (ns)	0,628	1,01	1,05
Chu kỳ xung	Chu kỳ ghi (ns)	0,377	1,02	1,87
	Chu kỳ đọc (ns)	0,953	1,52	2,57
Công suất tiêu thụ	Dòng rỉ (μA)	18,24	10,59	7,03
	Công suất ghi (μW/Mhz)	3,94	3,38	2,76
	Công suất đọc (μW/Mhz)	4,03	3,10	3,09

Các đặc tính điện (công suất tiêu thụ, thời gian,...) của thiết kế được kiểm tra bằng mô phỏng thông qua HSPICE với các điều kiện hoạt động như liệt kê Bảng 1, gồm các góc công nghệ FF (NMOS fast, PMOS fast), TT (NMOS typical, PMOS typical) và SS (NMOS slow, PMOS slow). Các mẫu dữ liệu khác nhau sẽ được ghi vào bộ nhớ và sau đó đọc ra để so sánh. Việc đọc được thực hiện hai lần liên tiếp để kiểm tra việc đọc ở lần đầu tiên có phá hủy dữ liệu đã được lưu trong SRAM hay không.



Hình 9. Kết quả mô phỏng mạch dự trữ khi có tế bào nhớ bị lỗi

Bảng 2 trình bày các kết quả thu được sau khi mô phỏng với các mẫu và điều kiện hoạt động khác nhau. Ở góc FF, thời gian truy xuất chỉ tốn 0,66 ns và tần số hoạt động có thể lên tới 2,6 GHz. Nhưng đồng thời, ở chế độ này, dòng rỉ và công suất tiêu thụ cũng lớn nhất. Ngược lại, ở điều kiện hoạt động SS, thời gian truy xuất dữ liệu khoảng 2,15 ns và tần số hoạt động bị giảm xuống còn 389 MHz, nhưng dòng rỉ cũng giảm chỉ còn một nửa so với góc FF.

Hình 9 trình bày kết quả mô phỏng của mạch dự trữ với trường hợp hàng bị lỗi có địa chỉ là H'0008. Trong đó: (1) Dịch địa chỉ của hàng bitcell bị lỗi vào thanh ghi; (2) Tín hiệu RRR_SHIFT_EN xuống thấp, kết thúc quá trình dịch dữ liệu; (3) Bộ nhớ bắt đầu hoạt động bình thường (Ghi/đọc ở địa chỉ H'0000, tín hiệu HIT không tích cực. Bộ nhớ sẽ ghi/đọc ở các tế bào nhớ bình thường). (4) Ghi/đọc ở địa chỉ H'0008 (Tín hiệu HIT bật lên vì địa chỉ cần ghi/đọc giống với địa chỉ lưu trong thanh ghi. Hàng bitcell dự trữ được sử dụng. REDUNDANCY_WL tích cực để cho phép đọc/ghi vào hàng dự trữ).

4.2. So sánh với các công trình khác

Bảng 3. So sánh với các công trình khác

	[5]	[4]	[6]	Thiết kế
Công nghệ (nm)	180	90	40	90
Thế cung cấp (V)	1,8	1,2	0,65	1,2
Dung lượng (KB)	32	32	64	32
Tế Bào Nhớ	7T	6T	8T	6T
Redundancy	Không	Có	Không	Có
Mạch Hỗ Trợ Đọc/Ghi	Không	Không	Có	Không
Thời gian truy xuất (ns)	-	2,3	5	1,07
Thời gian ghi (ns)	1	-	-	1,02
Thời gian đọc (ns)	2	-	-	1,52
Dòng rỉ (μ A)	-	1,2	564	10,59

Bảng 3 so sánh kết quả của thiết kế với các công trình khác. Để so sánh hiệu suất của thiết kế, chúng tôi sử dụng các thông số đo đặc ở điều kiện hoạt động TT với VDD = 1,2 V và nhiệt độ là 25°C. Đây cũng là điều kiện hoạt động của các thiết kế được so sánh khi mô phỏng trên schematic hoặc trong quá trình test chip.

Có thể thấy, so với [5] và [4] mặc dù có cùng dung lượng và được áp dụng kỹ thuật dự thừa hàng/cột nhưng thiết kế này vẫn có tốc độ truy xuất và thời gian đọc nhanh hơn. Thiết kế trong [4] sử dụng mạch giảm dòng rỉ cực cổng (Gate Leakage Suppression Circuit) nên dòng rỉ nhỏ hơn nhiều so với thiết kế được đề xuất. Mạch này sẽ hạ thấp mức điện thế cấp cho các tế bào nhớ khi chúng ở chế độ

giữ và chỉ nâng mức điện thế lên khi các tế bào nhớ hoạt động ở chế độ đọc/ghi. Tuy nhiên, thời gian trì hoãn giữa mỗi lần tăng/giảm mức điện thế khiến tổng thời gian truy xuất của bộ nhớ cũng tăng lên đáng kể. Tương tự, thiết kế ở công nghệ 40 nm [6] tập trung sử dụng các kỹ thuật để tăng sản lượng khi chế tạo (kiến trúc tế bào nhớ 8T, mạch hỗ trợ đọc/ghi), nhưng đồng thời cũng khiến dòng rỉ lên tới 564 μ A, trong khi tốc độ hoạt động cũng bị ảnh hưởng. Một cách tổng thể, thiết kế trong bài báo này đã đạt được các yêu cầu về tốc độ hoạt động, công suất, sản lượng, và có thể ứng dụng trong các bộ vi xử lý hoặc vi điều khiển.

5. Kết luận

Trong bài báo này chúng tôi đã trình bày một thiết kế bộ nhớ SRAM đồng bộ 32 KB sử dụng kỹ thuật dự trữ hàng và cột để sử dụng trong vi xử lý hoặc vi điều khiển. Kết quả phân tích cho thấy nhờ cách bố trí sơ đồ thiết kế cùng với các kỹ thuật như tự căn chỉnh thời gian, đa hợp và giải mã đa tầng, thiết kế đạt được hiệu suất cao về tốc độ và công suất. Trong các nghiên cứu tiếp theo, chúng tôi sẽ tiến hành đi sâu vào các thuật toán ứng dụng trong các mạch BIST để kết hợp với kỹ thuật dự trữ hàng/cột.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Đại học Quốc gia Thành phố Hồ Chí Minh (VNU-HCM) trong khuôn khổ đề tài mã số C2013-18-02.

TÀI LIỆU THAM KHẢO

- [1] ITRS: International technology road map for semiconductors, test and test equipments. <http://public.itrs.net/>, 2012.
- [2] J. P. Bickford, R. Rosner, E. Hedberg, J. W. Yoder và T. S. Barnett, "SRAM Redundancy - Silicon Area versus Number of Repairs Trade-off", *IEEE/SEMI Advanced Semiconductor Manufacturing Conf. (ASMC)*, 2008, tr. 387 – 392.
- [3] K. Ishibashi and K. Osada, *Low Power and Reliable SRAM Memory Cell and Array Design*, Springer Series in Advanced Microelectronics, 2011.
- [4] Koji Nii, et al., "A 90-nm Low-Power 32-kB Embedded SRAM with Gate Leakage Suppression Circuit for Mobile Applications", *IEEE J. of Solid-State Circuits*, tập 39 (4), 2004, tr. 684-693.
- [5] M. Wiecekowsky và M. Margala, "A 32KB SRAM Cache Using Current Mode Operation and Asynchronous Wave-Pipelined Decoders", *IEEE Int. Conf. on SOC*, 2004, tr. 251-254.
- [6] Nan-Chun Lien, et al., "A 40 nm 512 Kb Cross-Point 8T Pipeline SRAM With Binary Word-Line Boosting Control, Ripple Bit-Line and Adaptive Data-Aware Write-Assist", *IEEE Trans. on Circuits and Systems*, tập 61(12), 2014, tr. 3416-3425.
- [7] Stanley Schuster, "Multiple Word/Bitline Redundancy for Semiconductor Memories", *IEEE J. of Solid State Circuits*, tập SC-13 (5), 1978, tr. 698-703.
- [8] T. Mano, M. Wada, N. Ieda và M. Tanimoto, "A Redundancy Circuit for a Fault-Tolerant 256K MOS RAM", *IEEE J. of Solid State Circuits*, tập SC-17 (4), 1982, tr. 726-731.