

## MỤC LỤC

<b>CHƯƠNG I: TÌM HIỂU VỀ PHẦN MỀM ELECTRIC .....</b>	<b>2</b>
1.1.    Phần mềm Electric.....	2
1.2.    Cách cài đặt và khởi động Electric .....	5
1.2.1.    Cách cài đặt.....	5
1.2.2.    Khởi động Electric .....	5
1.3.    Display.....	8
1.3.1.    The Toolbar.....	9
1.3.2.    The Messages Window.....	10
1.3.3.    The Sidebar .....	10
<b>CHƯƠNG II: THỰC HIỆN THIẾT KẾ VÀ LAYOUT MỘT Ô NHỚ SRAM... </b>	<b>12</b>
1.1.    Sơ lược về RAM.....	12
2.2.    SRAM.....	13
2.2.1.    SRAM cells.....	14
2.2.2.    Hoạt động đọc của SRAM .....	15
2.2.3.    Hoạt động ghi của SRAM.....	17
2.3.    Thiết kế vật lý .....	17
2.4.    Thực hiện layout tế bào SRAM.....	20
2.4.1.    Vẽ sơ đồ nguyên lý .....	20
2.4.2.    Thiết kế và layout cell SRAM 6T .....	22
<b>CHƯƠNG III: KẾT LUẬN.....</b>	<b>25</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>26</b>

## CHƯƠNG I: TÌM HIỂU VỀ PHẦN MỀM ELECTRIC

### 1.1. Phần mềm Electric

Electric VLSI Design System là một công cụ thiết kế vi mạch được viết bởi Steven M. Rubin. Electric được sử dụng để vẽ sơ đồ và để làm mạch tích hợp layout. Nó cũng có thể xử lý các ngôn ngữ mô tả phần cứng như VHDL và Verilog.

Electric là một phần mềm chuyên dụng, chứa đầy đủ các công cụ hỗ trợ cho việc thiết kế chuyên nghiệp. Được gọi là hệ thống EDA (Electronic Design Automation). Là phần mềm miễn phí trong nhiều năm, hiện tại nó được viết bằng ngôn ngữ java.

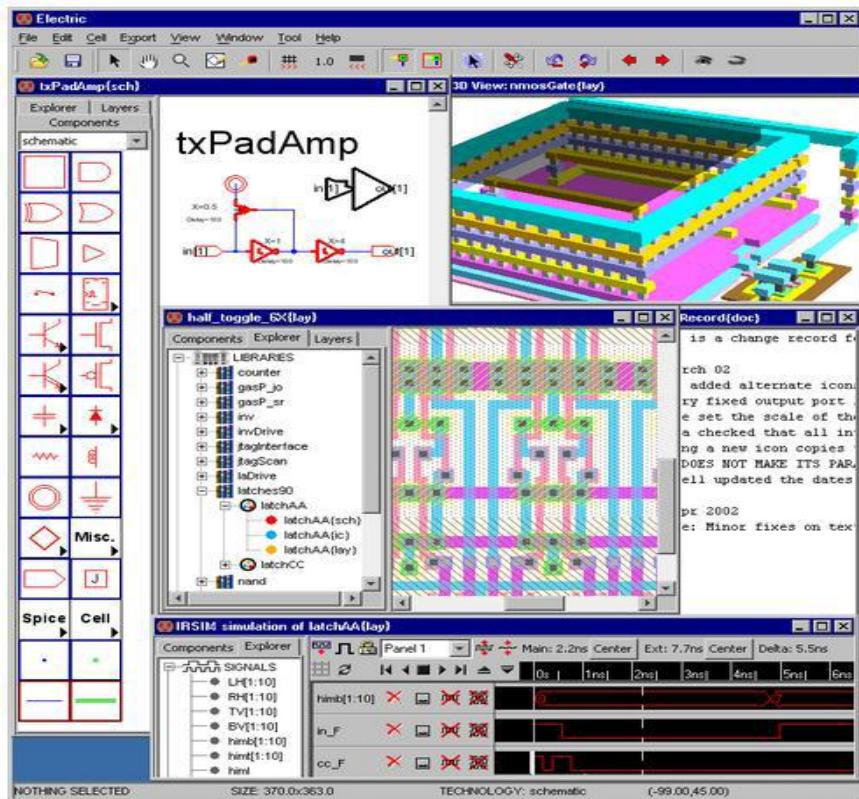
Chức năng nổi bật của Electric là cho phép tạo ra các bản thiết kế giản đồ, layout, theo mỗi công nghệ tùy chọn. Sau đó Electric có thể chuyển đổi các bản layout thành các bản thiết kế netlist hay VHDL, Verilog . . . để phục vụ cho công việc thiết kế top-down một cách dễ dàng, chính xác.

Khả năng mềm dẻo của Electric còn hỗ trợ giao tiếp (import và export) với các định dạng thiết kế ngoài môi trường của Electric hoặc được sử dụng đặc thù như : CIF, GDS II . . . Một đặc điểm khác làm cho Electric trở nên hữu dụng thực sự là nó cung cấp các công cụ kiểm soát lỗi thiết kế và gỡ rối mạnh mẽ như công cụ DRC, công cụ mô phỏng dạng sóng.

Các mạch tích hợp trong Electric được tạo từ các nút (nodes) và các vòng cung (arcs). Nodes là các yếu tố như là các transistor hoặc các contact. Arcs kết nối tới các node. Đây là điểm khác biệt so với những công cụ hỗ trợ thiết kế mạch tích hợp khác như magic hoặc cadence – những phần mềm này thường thiết kế trên các lớp của wafer.

Phần mềm thiết kế hệ thống VLSI là một phần mềm mã nguồn mở về thiết kế hệ thống điện tử tự động hóa (Electronic Design Automation-EDA) có thể xử lí được nhiều hình thức thiết kế mạch bao gồm:

- Vẽ layout của IC
- Vẽ sơ đồ kiến trúc của hệ thống (Schematic Capture)
- Viết ngôn ngữ mô tả (VHDL, Verilog,...)
- .....



**Hình 1.1: Ví dụ về Electric**

Electric có thể thiết kế IC cho MOS hoặc Bipolar với các công cụ hỗ trợ như kiểm tra các luật thiết kế (design rule), mô phỏng, router, tạo layout và các công cụ hỗ trợ khác.

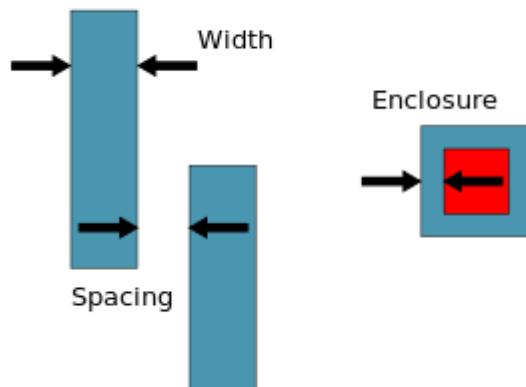
Các công cụ hỗ trợ:

- **Design Rule Checking – DRC** (Kiểm tra luật thiết kế). Đối với mỗi bản thiết kế layout cmos nào cũng phải tuân theo một quy luật thiết kế, quy luật này là tuyệt đối và đòi hỏi người thiết kế phải tuân theo để đảm bảo sự hoạt động của linh kiện cũng như đảm bảo các điều kiện để sản xuất. Electric hỗ trợ một số luật (tiêu chuẩn) cho các thiết kế, tuy nhiên bạn hoàn toàn có thể tạo mới hay chỉnh sửa các luật này cho phù hợp với thiết kế của bạn.

Mục tiêu chính của kiểm tra quy tắc thiết kế (DRC) là để đạt được năng suất cao và độ tin cậy tổng thể cho việc thiết kế. Nếu bị vi phạm quy tắc thiết kế thiết kế có thể không có chức năng. Để đáp ứng mục tiêu cải thiện năng suất chất lượng, DRC đã phát triển từ đo lường và kiểm tra đơn giản Boolean, nhiều quy tắc có liên quan sửa đổi các tính năng hiện có, chèn các tính năng mới, và kiểm tra toàn bộ thiết kế của các hạn chế

quá trình như mật độ lớp. Layout hoàn thành bao gồm không chỉ của các đại diện hình học của thiết kế, mà còn dữ liệu cung cấp hỗ trợ cho sản xuất của thiết kế. Trong khi thiết kế kiểm tra quy tắc tối không nhận rằng các thiết kế sẽ hoạt động một cách chính xác, chúng được xây dựng để xác minh rằng cấu trúc đáp ứng các hạn chế quá trình cho một loại thiết kế và quy trình công nghệ.

### The three basic DRC checks



Hình 1.2: DRC

3 vấn đề cơ bản mà DRC kiểm tra là: chiều rộng, khoảng cách và bao vây.

- **Simulation** (Mô phỏng): với công cụ này, bạn có thể mô phỏng các thiết kế của mình bằng cách cung cấp một số trạng thái tín hiệu cho các đầu vào và quan sát dạng tín hiệu ra. Kết quả của việc mô phỏng sẽ giúp bạn kiểm thử các thiết kế của mình dưới dạng mạch nguyên lý hoặc sau khi thiết kế layout mạch tích hợp.

- **LVS** (Layout vs. Schematic): 2 công cụ cơ bản để thiết kế mạch nguyên lý và chạy layout.

- **Logical Effort** là một kỹ thuật đơn giản được sử dụng để ước tính độ trễ trọng một mạch CMOS. Sử dụng đúng cách, nó có thể hỗ trợ trong việc lựa chọn cửa cho một chức năng nhất định (bao gồm cả số lượng của các giai đoạn cần thiết) và định cỡ cửa để đạt được sự chậm trễ tối thiểu có thể cho một mạch.

## 1.2. Cách cài đặt và khởi động Electric

### 1.2.1. Cách cài đặt

Có nhiều cách để cài đặt và sử dụng phần mềm tùy theo hệ điều hành và các điều kiện:

- Khởi động từ web: Đây là cách đơn giản nhất để thực hiện việc cài đặt. Để có thể thực hiện, cần phải download file Java Web Start:

<http://java.net/projects/Electric/downloads/download/Electric.jnlp>

Sau đó nó sẽ tự động download các file cần thiết về để thực hiện cài đặt, với việc chạy lần đầu tiên sẽ lâu hơn do cần phải download các gói cần thiết về, quá trình khởi động sẽ diễn ra nhanh từ lần mở thứ 2 trở đi.

- Download file JAR từ GNU: Đây là một cách để cài đặt Electric trên UNIX.

- Download file Electric.JNLP từ <http://www.staticfreesoft.com/products.html> để cài đặt Electric trên Windows7.

### 1.2.2. Khởi động Electric

Window là hệ điều hành thường được sử dụng và việc chạy Electric trên Window cũng hết sức đơn giản. Tuy nhiên hạn chế khi chạy Electric trên Window là yêu cầu máy tính phải kết nối mạng trong mỗi lần chạy Electric.

Để chạy Electric trên Window, trước tiên bạn phải cài đặt Java cho máy tính của mình. Có thể tải về và cài đặt Java một cách miễn phí trên trang chủ của Java là:

<http://www.java.com/en/download/index.jsp>

Sau khi cài đặt Java cho máy, bạn truy cập vào trang chủ của Electric là: <http://www.staticfreesoft.com>, vào phần Products và tải về Java Web Start – file cần thiết để bạn có thể chạy Electric. Hoặc đơn giản bạn có thể copy đường link bên dưới để tải về Java Web Start, file tải về là **Electric.jnlp**.

<http://java.net/projects/Electric/downloads/download/Electric.jnlp>

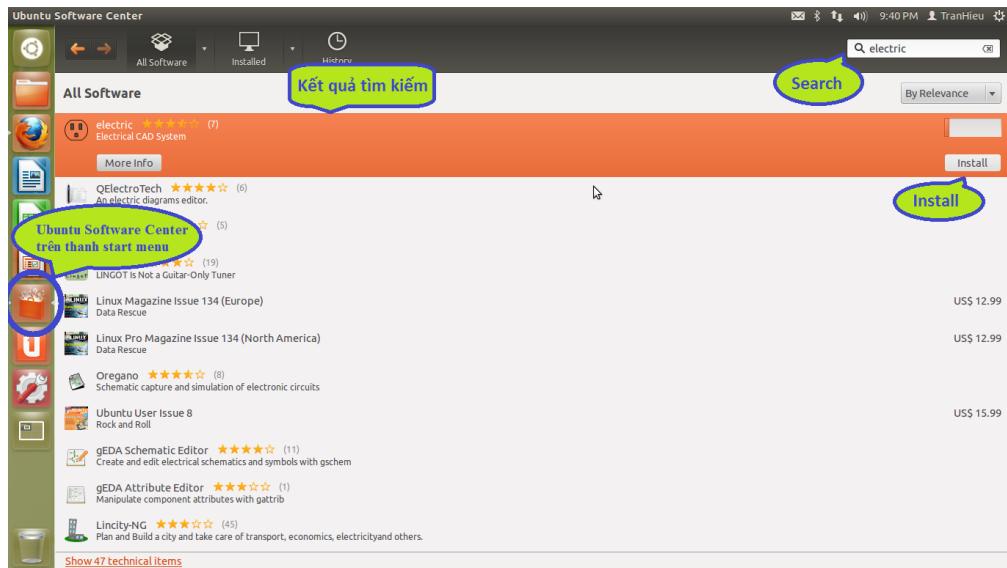
Sau khi tải file Electric.jnlp bạn mở file này lên. Ở lần chạy Electric đầu tiên, sẽ tốn của bạn khoảng 10 phút chờ đợi, còn ở các lần chạy sau thì quá trình khởi động sẽ nhanh hơn.

Cài đặt và chạy Electric trên Ubuntu cũng rất đơn giản, ngoài ra có một ưu điểm đó là nó không yêu cầu kết nối mạng trong mỗi lần khởi chạy. Tuy nhiên do hệ điều hành này ít được sử dụng hơn Window nên cách này ít được sử dụng.

Bạn có thể vào trang chủ của Electric để xem hướng dẫn cách cài đặt Electric trên các hệ điều hành mã nguồn mở nói chung. Ở phần tiếp sẽ trình bày cách cài đặt Electric một cách đơn giản trên hệ điều hành Ubuntu 12.4 bằng hình ảnh.

### **Bước 1:** Tìm kiếm Electric ở Ubuntu Software Center.

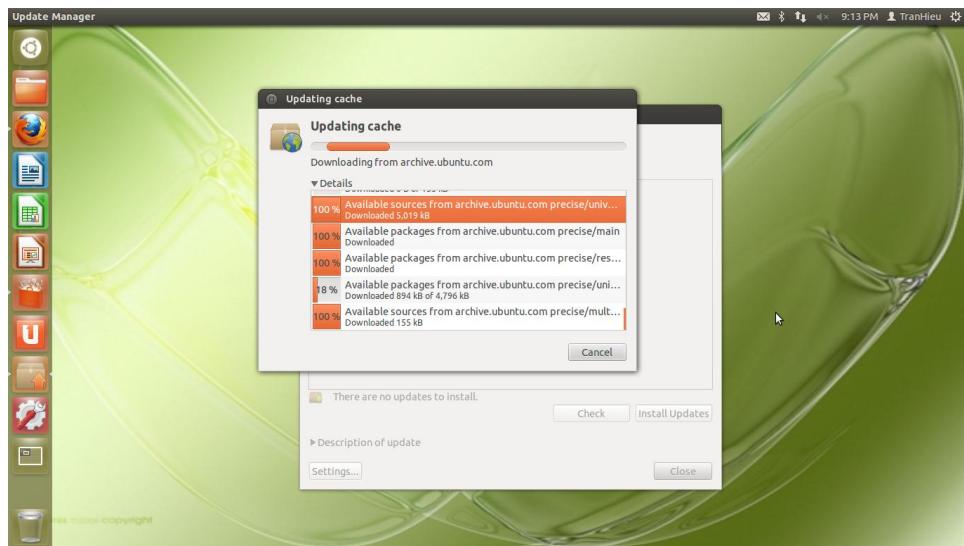
Vào *Ubuntu Software Center* trên thanh Start Menu, chọn chế độ *All Software* và gõ vào ô *Search* từ “Electric”. Khi đó ở phần kết quả sẽ tìm thấy gói (Package) cài đặt Electric như hình dưới.



**Hình 1.3: Giao diện của ubuntu software**

Chú ý: Nếu các bạn không tìm thấy gói cài đặt cho Electric ở *Ubuntu Software Center* thì bạn hãy Update cho hệ điều hành Ubuntu của bạn để *Ubuntu Software Center* cập nhật các gói mới.

Để update Ubuntu bạn click vào phần cá nhân ở góc trên bên phải màn hình của bạn, tìm tới phần Update. Khi đó sẽ hiện ra hình như bên dưới, bạn chọn Check để hiện các dữ liệu Update sau đó ấn Install Update để update.



**Hình 1.4: Update ubuntu software**

Sau khi update xong, hãy quay lại Ubuntu Software Centre để tìm lại.

### Bước 2: Cài đặt

Nhấn nút Install và chờ một lúc để quá trình cài đặt hoàn thành.



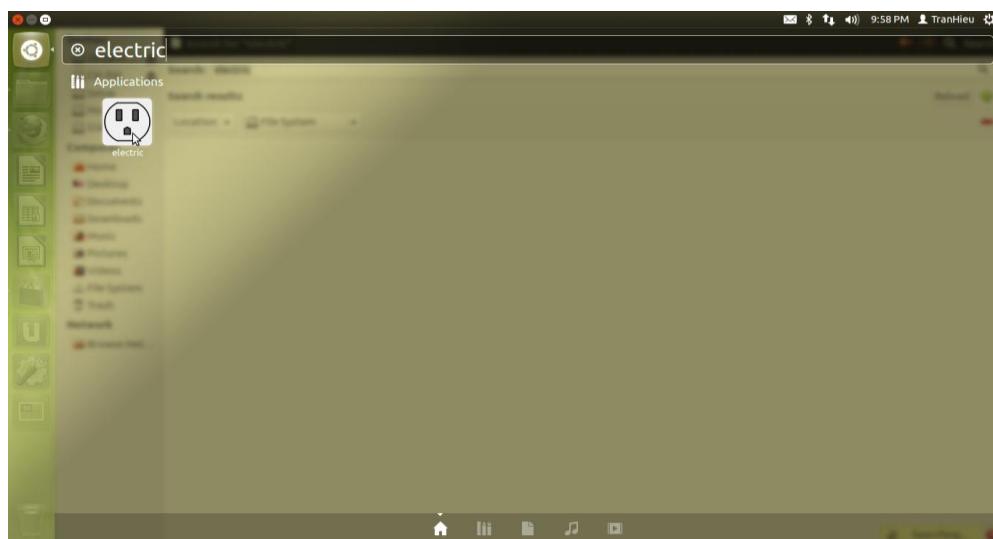
Khi cài đặt, gói công Electric sẽ được tải về và cài mặc định vào ổ hệ thống của Ubuntu (File system).

Sau khi cài đặt xong sẽ có dấu tích xanh hiện lên.



**Hình 1.5: Quá trình cài đặt Electric trên 7buntu**

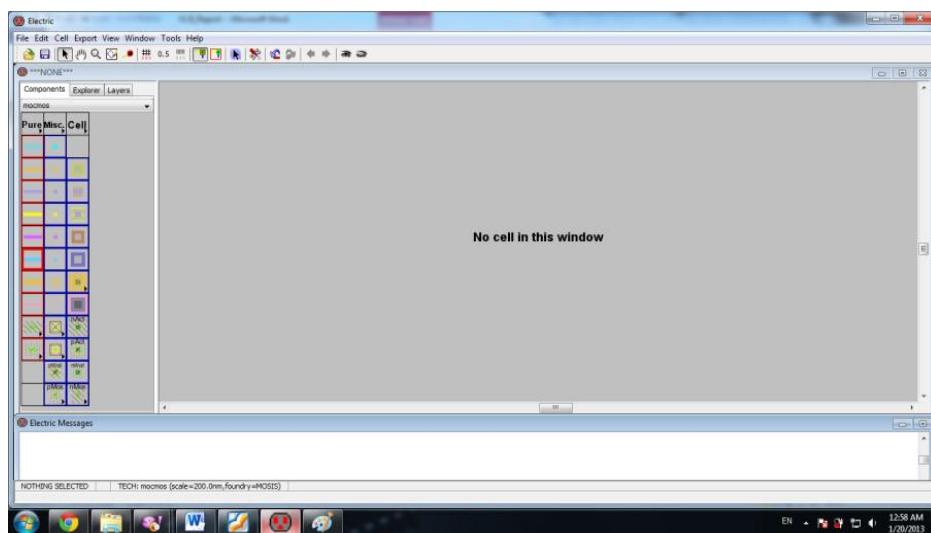
### Bước 3: Khởi động Electric



**Hình 1.6: Khởi động Electric**

Để khởi động Electric, bạn thực hiện việc tìm kiếm Electric tại cửa sổ tìm kiếm ứng dụng như hình vẽ trên. Click vào biểu tượng của Electric và chạy chương trình.

### 1.3. Display

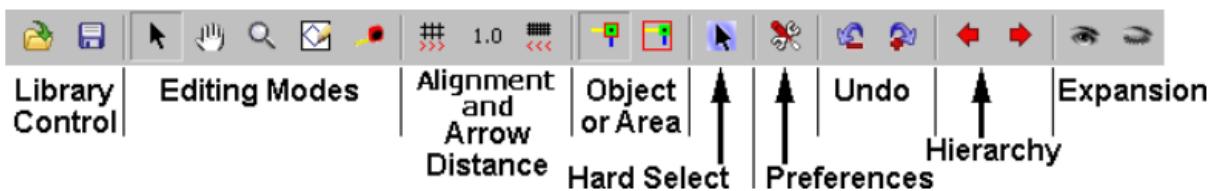


### Hình 1.7: Màn hình hiển thị Electric

Màn hình hiển thị chung của phần mềm Electric thể hiện như hình vẽ. Trên đó bao gồm các thành phần cơ bản như: Tool bar, Messages Window, ...

#### 1.3.1. The Toolbar

Thanh Toolbar nằm gần phần đỉnh của khung cửa sổ làm việc của Electric, nằm bên dưới thanh menu. Thanh này cung cấp một số phím tắt cho một số lệnh, công cụ thường được sử dụng trong quá trình thiết kế.



### Hình 1.8: The Toolbar trong Electric

Các thành phần của thanh Toolbar bao gồm:

- **Library Control:** bao gồm 2 biểu tượng để mở và lưu các thư viện. Thư viện thực hiện quản lý các file thiết kế schematic và PCB của thiết kế một IC.
- **Editing Modes:** bao gồm các biểu tượng selection , panning (công cụ di chuyển màn hình khi zoom), zooming , outline edit và measuring .
- **Alignment and Arrow Distance:** bao gồm các biểu tượng để tăng giảm grid (mạng lưới trong màn hình làm việc), biểu tượng bên tay trái là tăng độ lớn của grid, biểu tượng bên phải là giảm, số ở giữa thể hiện giá trị.
- **Object or Area:** bao gồm các biểu tượng để chuyển đổi sự lựa chọn giữa các đối tượng và định nghĩa khu vực.
- **Hard Select:** trong một mạch điện tích hợp, nhiều đối tượng có thể chồng lên nhau, gây nhầm lẫn khi lựa chọn để chỉnh sửa. Để đơn giản hóa lựa chọn, các đối tượng có thể được đánh dấu để nhanh chóng cho sự lựa chọn. Công cụ Hard Select dùng để thực hiện điều này.
- **Preferences:**
- **Undo:** bao gồm hai biểu tượng để undo hoặc redo một thao tác.

- **Hierarchy:** bao gồm các biểu tượng để chuyển đổi giữa các cửa sổ làm việc (cửa sổ thiết kế schematic hoặc layout) trong một thư viện.
- **Expansion:** bao gồm các biểu tượng để mở rộng hoặc thu hẹp cell hiện tại đang làm việc.

### 1.3.2. The Messages Window

The messages window là một cửa sổ text, cửa sổ này nằm phía dưới của cửa sổ Electric chính. Mỗi khi bạn thao tác một lệnh nào đó thì lệnh đó sẽ được báo dưới dạng text trong cửa sổ này, và kết quả thực hiện lệnh cũng được thể hiện. Nếu có lỗi xảy ra khi thực hiện một lệnh thì lỗi này cũng được báo trong Messages Window.



**Hình 1.9: The Messages Window trong Electric**

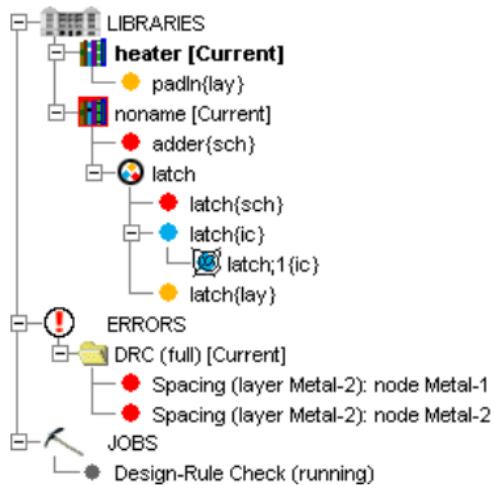
Các dòng text trong Messages Window có thể được lựa chọn cho việc cut , copy và past. Bạn có thể xóa tất cả các dòng text bằng lệnh **Clear** ( trong Window/Messages Window).

### 1.3.3. The Sidebar

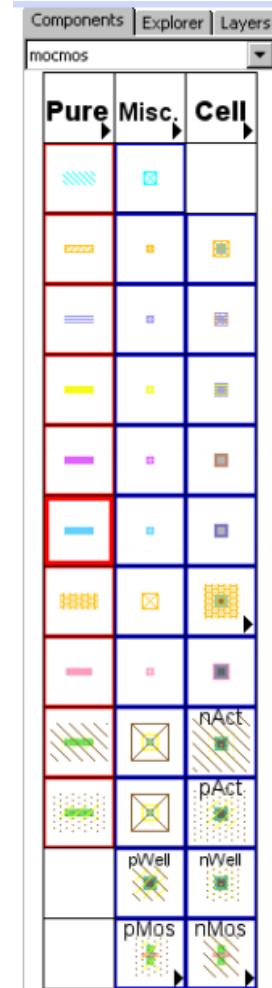
Thanh công cụ Sidebar nằm ở bên lề của cửa sổ Electric. Trên thanh Sidebar có 3 menu chính là: Components menu, Cell Explorer và Layer Visibility.

- ❖ **Components menu:** thể hiện các *node* và *arc* của công nghệ thiết kế đang sử dụng. Ô popup menu ở trên đỉnh cho phép bạn có thể thay đổi công nghệ thiết kế và nhìn thấy các node và arc của công nghệ tương ứng. Trong Components menu, các *node* có viền phác thảo màu xanh và các *arc* có viền phác thảo màu đỏ. Để đặt một *node* vào một ô trong cửa sổ thiết kế, click chuột trái vào một *node* để chọn nó, sau đó chuyển sang cửa sổ thiết kế và click chuột phải vào vị trí trong cửa sổ thiết kế để đặt *node*.

❖ **Cell Explorer:** nằm trong tab Explorer của thanh Sidebar, nó cho thấy một cây phân cấp với ba thành phần chính là: Libraries , Errors và Jobs. Phần Libraries của Explorer là một danh sách các Library và các cell thiết kế. Phần Errors chứa các dòng thông báo nếu có lỗi trong quá trình thiết kế hoặc các lỗi vi phạm luật thiết kế khi kiểm tra bằng DRC.



**Hình 1.10: The Cell Explorer**



## CHƯƠNG II: THỰC HIỆN THIẾT KẾ VÀ LAYOUT MỘT Ô NHỚ SRAM

### 1.1. Sơ lược về RAM

RAM (Random Access Memory) là một loại bộ nhớ. RAM được gọi là bộ nhớ truy xuất ngẫu nhiên vì nó có đặc tính: thời gian thực hiện thao tác đọc hoặc ghi đối với mỗi ô nhớ là như nhau, cho dù đang ở bất kỳ vị trí nào trong bộ nhớ. Mỗi ô nhớ của RAM đều có một địa chỉ. Thông thường, mỗi ô nhớ là một byte (8 bit), tuy nhiên hệ thống lại có thể đọc ra hay ghi vào nhiều byte (2, 4, 8 byte).

RAM khác biệt với các thiết bị bộ nhớ tuần tự (sequential memory device) chẳng hạn như các băng từ, đĩa; mà các loại thiết bị này bắt buộc máy tính phải di chuyển cơ học một cách tuần tự để truy cập dữ liệu.

Bởi vì các chip RAM có thể đọc hay ghi dữ liệu nên thuật ngữ RAM cũng được hiểu như là một bộ nhớ đọc-ghi (read/write memory), trái ngược với bộ nhớ chỉ đọc ROM (read-only memory).

RAM thông thường được sử dụng cho bộ nhớ chính (main memory) trong máy tính để lưu trữ các thông tin thay đổi, và các thông tin được sử dụng hiện hành. Cũng có những thiết bị sử dụng một vài loại RAM như là một thiết bị lưu trữ thứ cấp (secondary storage). Thông tin lưu trên RAM chỉ là tạm thời, chúng sẽ mất đi khi mất nguồn điện cung cấp.

Theo công nghệ chế tạo RAM bao gồm 2 loại SRAM ( Static RAM), DRAM (Dynamic RAM):

SRAM (Static RAM): RAM tĩnh được chế tạo theo công nghệ ECL (dùng trong CMOS và BiCMOS). Mỗi bit nhớ gồm có các cổng logic với 4, 6, 8, ... transistor MOS. SRAM là bộ nhớ nhanh, việc đọc không làm hủy nội dung của ô nhớ và thời gian thâm nhập bằng chu kỳ của bộ nhớ.

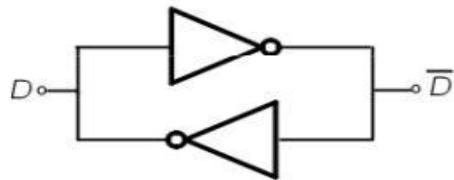
DRAM (Dynamic): RAM động dùng kỹ thuật MOS. Mỗi bit nhớ gồm một transistor và một tụ điện. Việc ghi nhớ dữ liệu dựa vào việc duy trì điện tích nạp vào

tụ điện và như vậy việc đọc một bit nhớ làm nội dung bit này bị hủy. Do vậy sau mỗi lần đọc một ô nhớ, bộ phận điều khiển bộ nhớ phải viết lại nội dung ô nhớ đó. Chu kỳ bộ nhớ cũng theo đó mà ít nhất là gấp đôi thời gian thâm nhập ô nhớ.

Việc lưu giữ thông tin trong bit nhớ chỉ là tạm thời vì tụ điện sẽ phóng hết điện tích đã nạp và như vậy phải làm tươi bộ nhớ sau khoảng thời gian  $2\mu s$ . Việc làm tươi được thực hiện với tất cả các ô nhớ trong bộ nhớ. Công việc này được thực hiện tự động bởi một vi mạch bộ nhớ. Bộ nhớ DRAM chậm nhưng rẻ tiền hơn SRAM.

## 2.2. SRAM

RAM tĩnh sử dụng một tế bào bộ nhớ với thông tin phản hồi bên trong và vẫn giữ được dữ liệu miễn là nó vẫn được cấp nguồn vì vậy nó không cần phải làm mới (refresh).



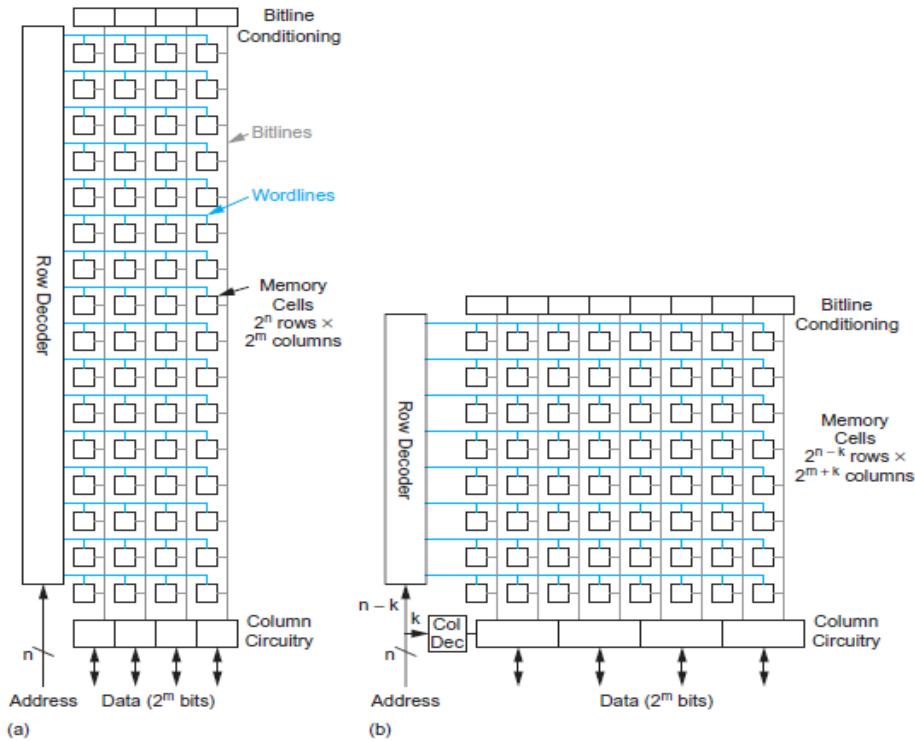
**Hình 2.1: Thành phần bộ nhớ tĩnh**

Nếu  $D$  ở mức cao thì  $\bar{D}$  ở mức thấp, sử dụng mạch phản hồi tích cực.

SRAM có các đặc tính vượt trội sau:

- Kích thước một tế bào nhớ nhỏ hơn một flip-flops.
- Tương thích với các quy trình chuẩn CMOS.
- Nhanh hơn DRAM.
- Dễ dàng sử dụng hơn DRAM.

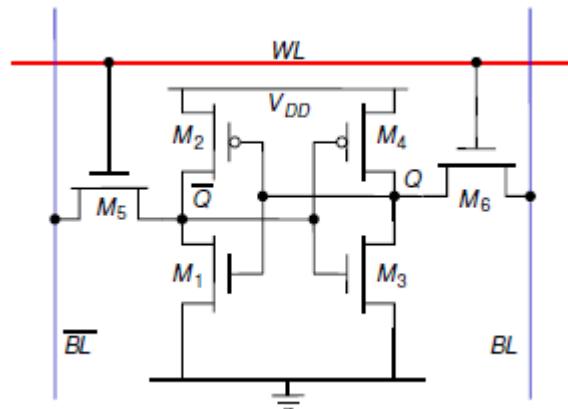
Vì những lý do trên, SRAM được sử dụng rộng rãi trong các ứng dụng như bộ nhớ cache. SRAM bao gồm một loạt các tế bào bộ nhớ cùng với các hàng và các cột mạch



Hình 2.2: Kiến trúc mảng bộ nhớ

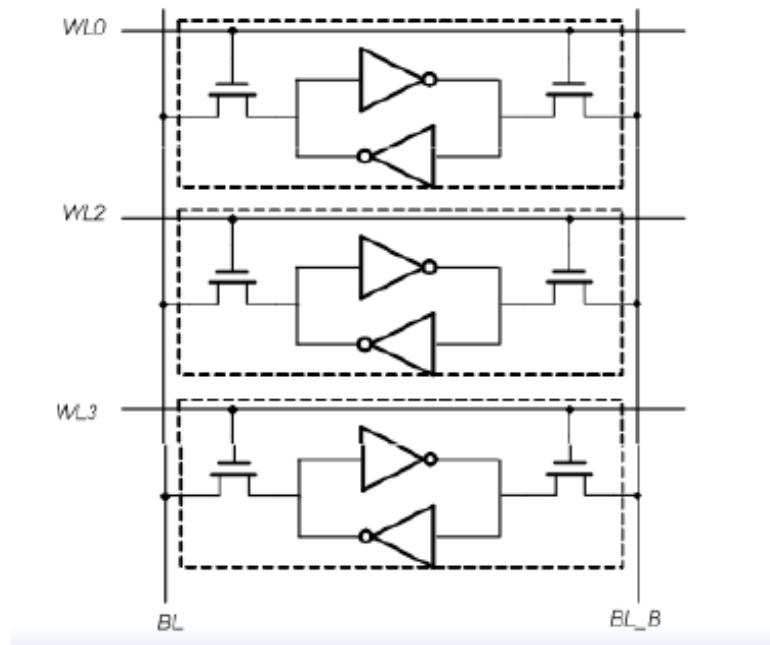
### 2.2.1. SRAM cells

Một tế bào SRAM có thể đọc, ghi dữ liệu và tổ chức dữ liệu miễn là nó vẫn được cung cấp điện áp để hoạt động. Bình thường một flip-flop có thể thực hiện những việc trên, nhưng khích thước là khá lớn. Hình 2.2 cho thấy một tế bào SRAM chuẩn 6-transistor (6T) tính về độ lớn thì nó nhỏ hơn một flip-flop



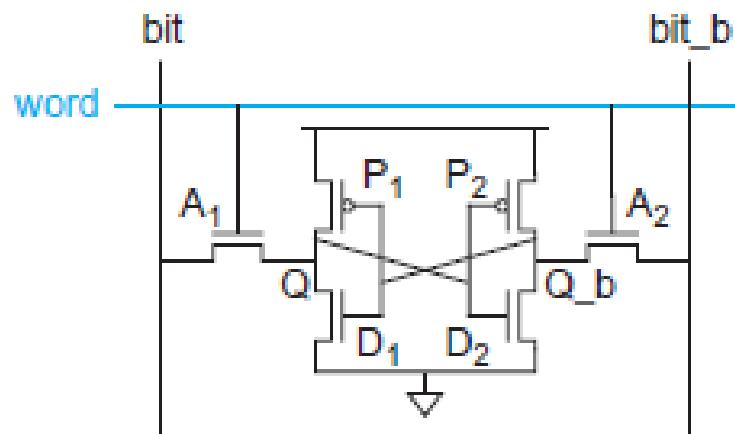
Hình 2.3: Tế bào SRAM 6T

Kích thước tế bào nhỏ dẫn đến các dây nối ngắn hơn và tiêu thụ điện năng thấp hơn do đó có tính năng động. Các tế bào SRAM 6T có một đôi transistor truy cập để đọc hoặc viết và một đôi cổng đảo măc chéo nhau để giữ trạng thái. Các thông tin phản hồi tích cực hiệu chỉnh nhiễu gây ra do dò rỉ hoặc tiếng ồn.

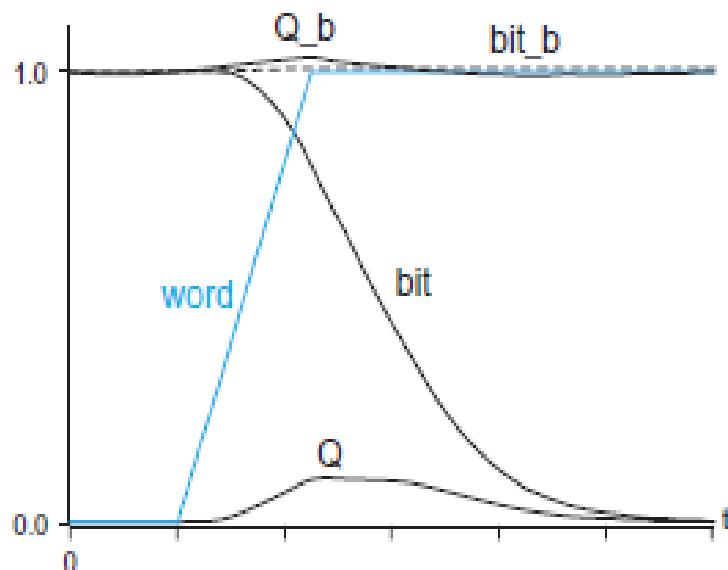


**Hình 2.4: Cấu trúc SRAM**

### 2.2.2. Hoạt động đọc của SRAM



**Hình 2.5: Tế bào SRAM 6T**

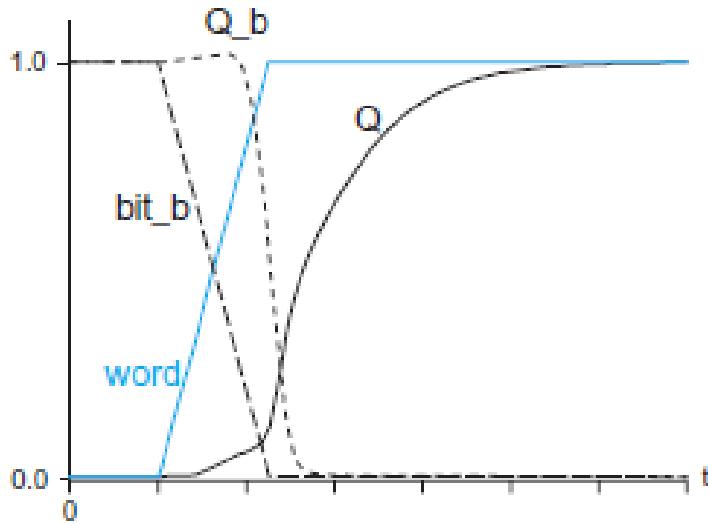


**Hình 2.6: Hoạt động đọc của tế bào SRAM 6T**

Hình 2.6 cho thấy tế bào SRAM được đọc. Ban đầu các bitlines ở mức cao (1). Giả sử  $Q$  ban đầu ở mức thấp (0) và do đó  $Q_b$  ở mức 1. Khi wordline được nâng lên mức 1 thì bit đang ở mức 1 sẽ được kéo xuống thông qua quá trình điều khiển và truy cập các transistor D1 và A1. Tại thời gian bit kéo xuống,  $Q$  có xu hướng tăng,  $Q$  được giữ thấp bởi D1, nhưng có xu hướng tăng bởi dòng chảy từ A1. Do đó, cần phải điều khiển D1 thông hơn transistor truy cập A1, cụ thể là các transistor phải được mắc theo kiểu ratioed để nút  $Q$  vẫn thấp hơn ngưỡng chuyển mạch của bộ đảo P2/D2 và việc này được gọi là đọc ổn định. Dạng sóng của hoạt động được thể hiện trong hình 2.6 quan sát thấy  $Q$  tăng lên trong giây lát nhưng không đủ để lật các tế bào.

Trong giai đoạn 2, các bitlines được nạp ở mức cao. Các wordline chỉ tăng lên trong một giai đoạn. Nhiều tế bào SRAM cùng sử dụng chung đôi bitline, hoạt động như một đường kép đa năng. Điện dung của toàn bộ bitline được đưa ra ngoài nhờ các transistor truy cập. Đầu ra có thể được cảm nhận bởi một cặp dò sai.

### 2.2.3. Hoạt động ghi của SRAM



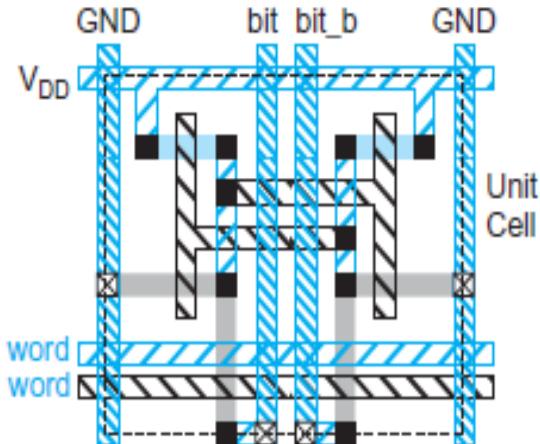
**Hình 2.7: Hoạt động ghi của tế bào SRAM 6T**

Hình 2.7 cho thấy tế bào SRAM đang được ghi. Một lần nữa, giả sử  $Q$  ban đầu ở mức 0 và chúng ta muốn viết 1 vào các tế bào. Bit được nạp ở mức cao,  $bit\_b$  được kéo xuống thấp bởi một trình điều khiển ghi. Chúng ta biết rằng việc tính toán các hạn chế của việc đọc ổn định là bit bắt buộc  $Q$  ở mức thấp thông qua A1. Do đó, các tế bào phải được ghi bằng cách đưa  $Q\_b$  xuống thấp thông qua A2. P2 sẽ chống lại việc kéo  $Q\_b$  xuống thấp do đó P2 phải thông yêu hơn A2 để  $Q\_b$  có thể được kéo xuống thấp. Hạn chế này được gọi là writability. Khi  $Q\_b$  ở mức thấp thì D1 tắt và P1 thông kéo  $Q$  lên mức cao như mong muốn.

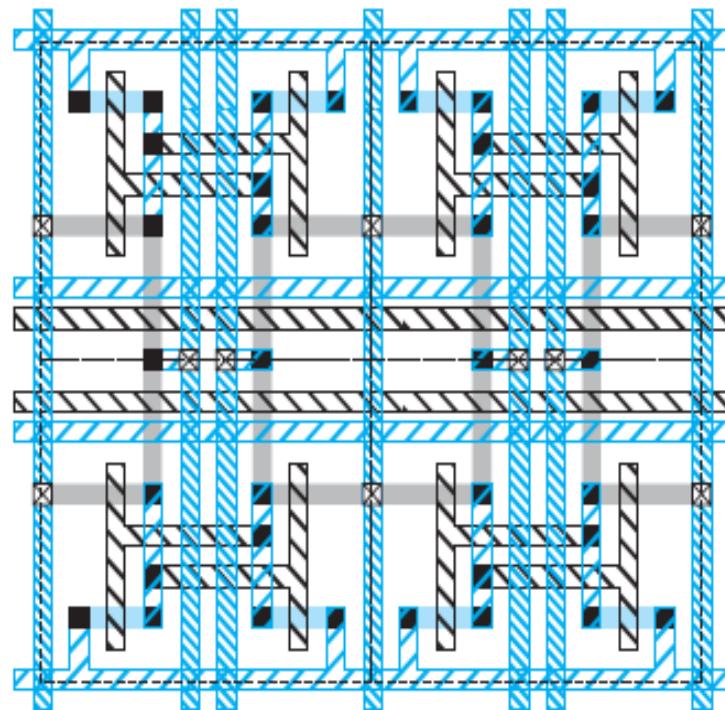
Trong giai đoạn 2, bitlines được nạp ở mức cao. Điều khiển ghi sẽ kéo bitline hoặc phần bù của nó thấp trong giai đoạn 1 tới việc ghi tế bào. Các điều khiển ghi có thể bao gồm một cặp transistor trên mỗi bitline để dữ liệu cho phép ghi hoặc một transistor đơn.

### 2.3. Thiết kế vật lý

Tế bào SRAM yêu cầu sắp đặt thông minh để đặt được mật độ tốt. Một thiết kế truyền thống đã được sử dụng cho đến khi thế hệ 90nm ra đời.



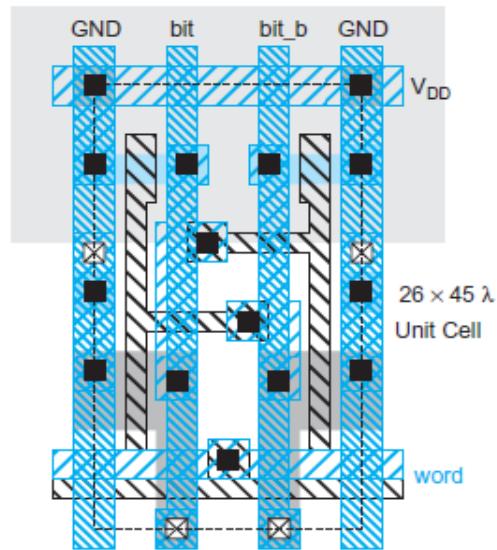
Hình 2.8: Sơ đồ thanh của tế bào SRAM 6T



Hình 2.9: Nhiều tế bào SRAM 6T

Hình 2.8 thể hiện sơ đồ thanh của một tế bào SRAM 6T truyền thông. Các tế bào được thiết kế để được nhân đôi và chồng chéo nhau để chia sẻ VDD và GND (hình 2.9). Lưu ý rằng một số liên kết khuếch tán duy nhất để các dòng bit được dùng chung giữa một đôi của các tế bào. Điều này làm giảm một nửa điện dung

khuếch tán, và do đó làm giảm sự trễ xả bitline. Các wordline được chạy trong cả metal1 và polysilicon.

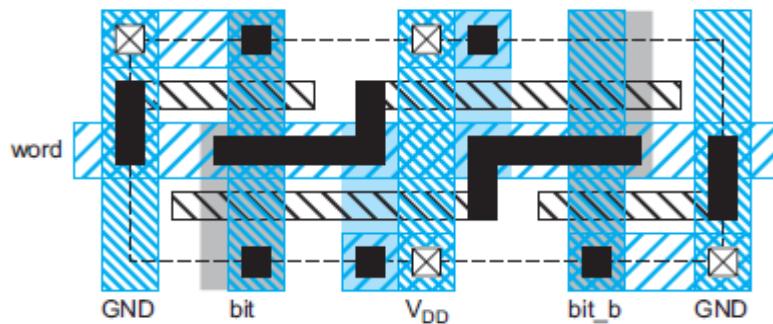


**Hình 2.10: Layout tế bào SRAM 6T**

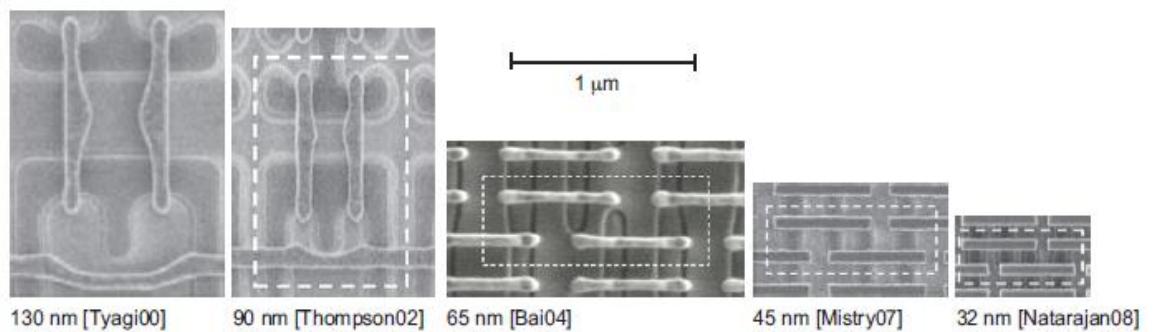
Hình 2.10 cho thấy một tế bào có định 26x45 λ, tuân theo quy tắc thiết kế MOSIS sub-micron. Trong cách sắp đặt này, wordlines được liên kết giữa metal1 và polysilicon trong mỗi tế bào. Nẽ cũng được liên kết trong mỗi tế bào.

Các gấp khúc trong silic đa tinh thể và khuếch tán là khó chế tạo chính xác khi kích thước là nhỏ hơn bước sóng của ánh sáng. Như vậy quá trình sử dụng quang khắc nanomet của trê bào 6T được thể hiện trong hình 2.11. Sự khuếch tán chạy đúng theo hướng thẳng đứng và polysilicon chạy theo hướng ngang. Sắp đặt hai dây ngang metal1 và sáu dây dọc metal2. Nó sử dụng kết nối cụ bộ hoặc liên kết rãnh làm cầu nối giữa cực máng pMOS và transistor nMOS và cực polysilicon.

Sự khuếch tán NMOS với độ rộng không bằng nhau để đạt được một tỉ lệ beta lớn hơn 1. Các khe khuếch tán có xu hướng lan rộng ra vì những hạn chế của quá trình quang khắc. Do đó, sự không thẳng hàng của polysilicon dẫn đến sự khuếch tán có thể thay đổi hiệu quả của các transistor. Một layout khác sử dụng khuếch tán nhỏ nhất cho hai transistor NMOS và tỉ lệ beta bằng 1.



**Hình 2.11: Quang khắc của tế bào SRAM 6T**



**Hình 2.12: Thế hệ thu nhỏ SRAM**

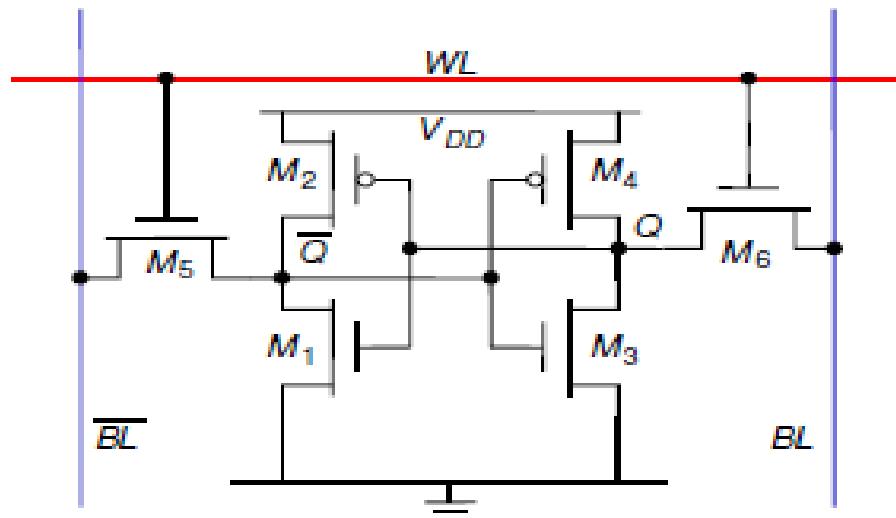
Hình 2.12 cho thấy năm thế hệ thu nhỏ SRAM, các ảnh hiển vi hiển thị các vùng khuếch tán và polysilicon.

## 2.4. Thực hiện layout tế bào SRAM

### 2.4.1. Vẽ sơ đồ nguyên lý

Theo sự tìm hiểu ở phần trên về SRAM thì nhóm thực hiện vẽ tế bào SRAM 6T (6 transistor).

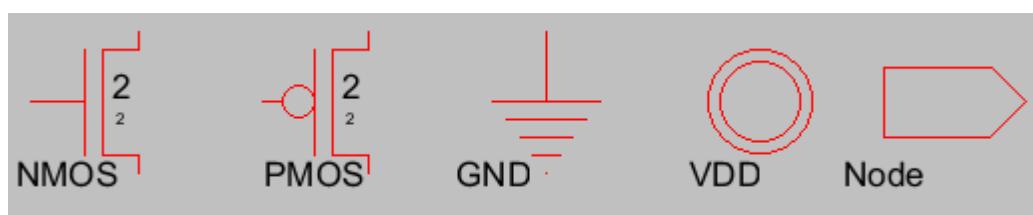
Ta có mạch nguyên lý của tế bào SRAM 6T như sau:



**Hình 2.13: Cell SRAM 6T**

Mạch sử dụng 2 transistor PMOS và 4 transistor NMOS, thực hiện các bước như sau:

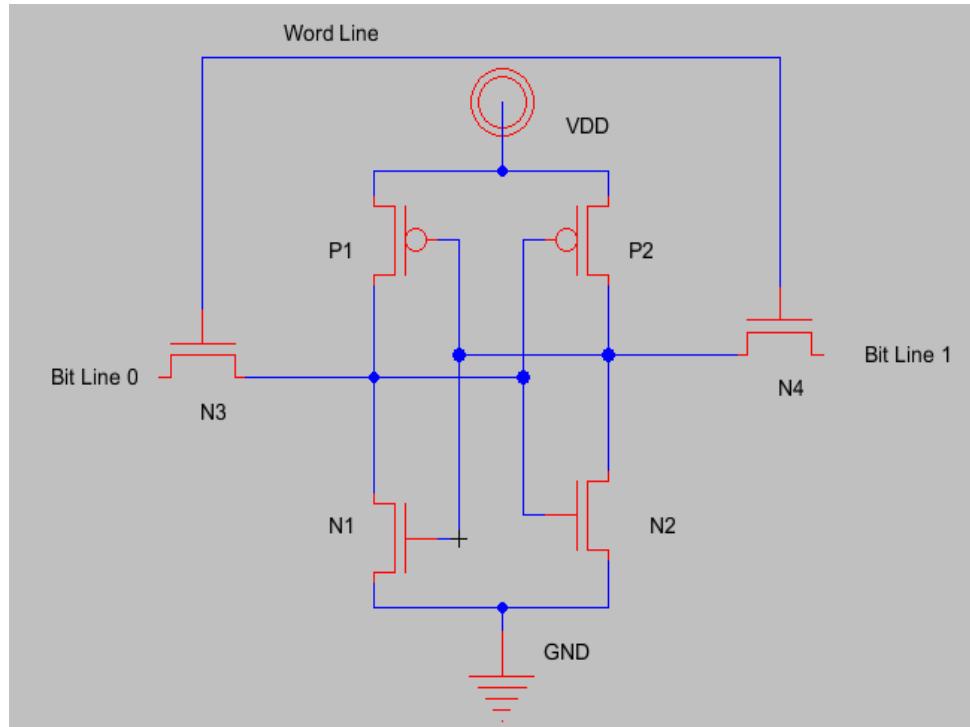
- B1: Vào file -> new library -> tên thư viện cần tạo (VD: SRAM6T\_2).
- B2: Vào Cell -> new cell -> chọn schematic và điền tên: SRAM6T\_2.
- B3: Vào component để chọn các thành phần phù hợp với thiết kế. Với mạch này, các thành phần cần có là NMOS, PMOS, GND, VDD và NODE lần lượt như sau:



**Hình 2.14: Các thành phần trong component của schemactic**

- B4: Đặt vị trí các thành phần đúng vị trí và chuẩn bị nối dây.
  - Cách nối dây: Bấm chuột trái vào thành phần đầu tiên (điểm đầu) để hiện lên ô vuông màu trắng bao quanh, sau đó bấm và giữ chuột phải, di chuyển đến điểm cần nối.

- Các thao tác khác có thể sử dụng bằng phím tắt như: Ctrl+ G: xem ở chế độ lưới, Ctrl + J để xoay linh kiện, Ctrl + M để copy,...
- Sau khi thực hiện thao tác nối dây, ta có sơ đồ như sau:



Hình 2.14: Sơ đồ cell SRAM 6T trong Electric

#### 2.4.2. Thiết kế và layout cell SRAM 6T

Khi đã hoàn tất công việc thiết kế schematic, bây giờ là thời điểm để đến bước tiếp theo là thiết kế layout.

Ta thực hiện các bước layout cell SRAM6T như sau:

B1: Ctrl + N hoặc new cell để thêm cell mới -> chọn layout

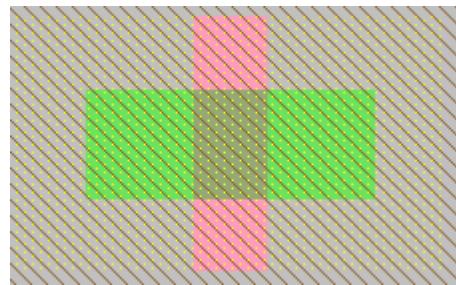
B2: Viết tên layout cần tạo: SRAM6T

B3: Bấm vào component để thấy các thành phần của mạch

- Trong Electric, với component của moscoss hỗ trợ các thành phần có sẵn như nmos, pmos, nwell, pwell, metal-active,...

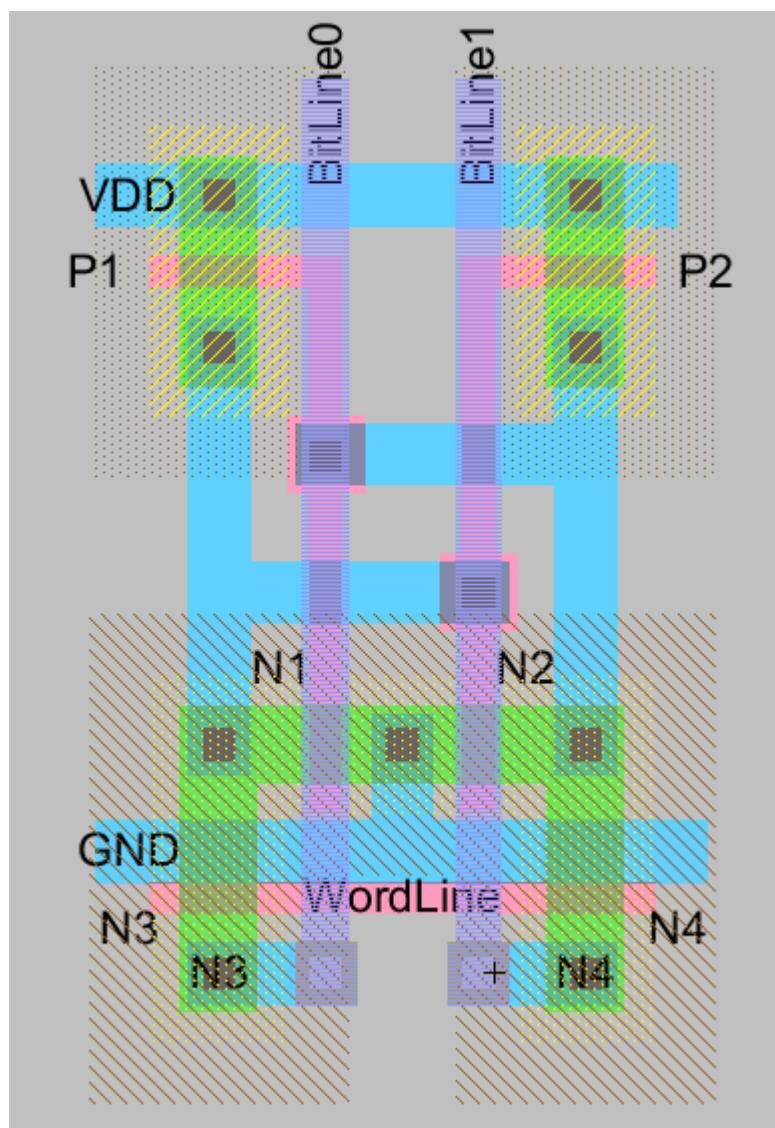
- Nmos: có ký hiệu là hình polisilicon (màu hồng) ngang qua n-diffusion (màu xanh lá cây) và bao quanh là các dấu chấm màu vàng.

- Layout cơ bản của 1 transistor nmos như sau:

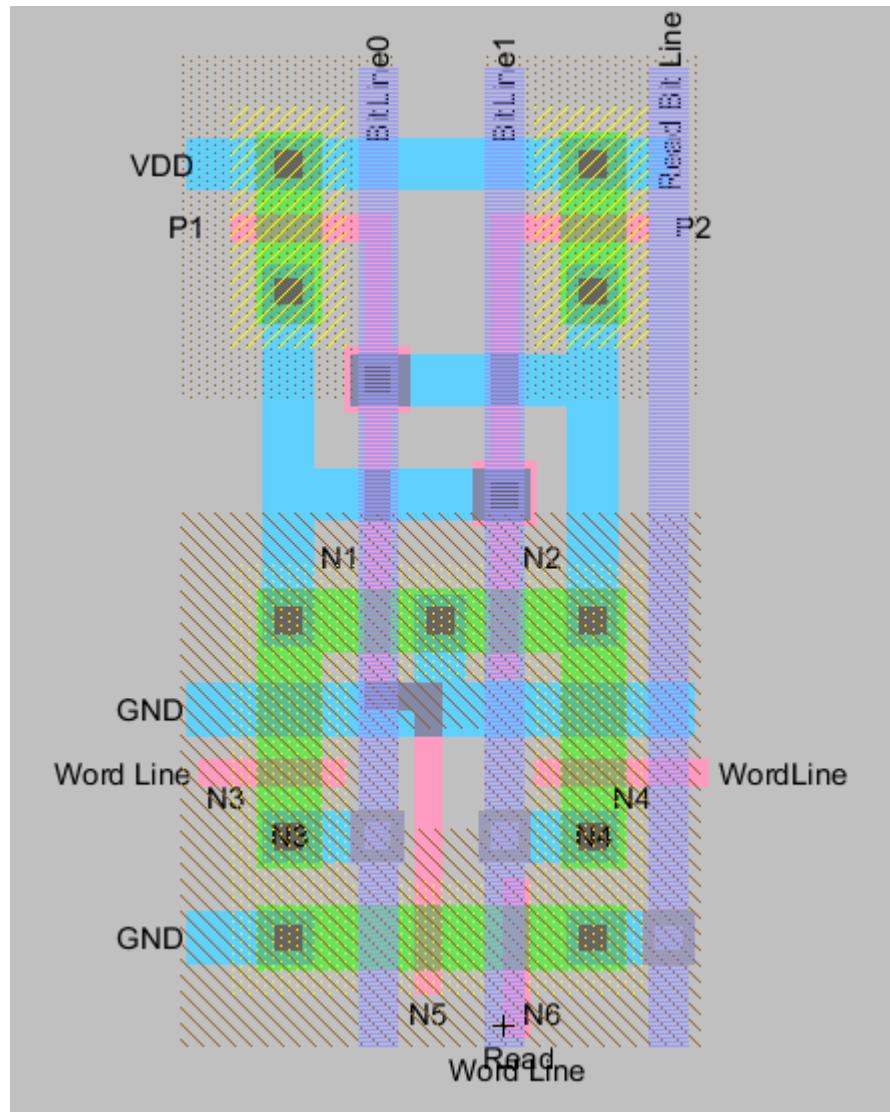


**Hình 2.15: Layout cơ bản của 1 tranzistor nmos**

- Chú ý là khi vẽ các nwell thì Electric đã hỗ trợ việc các lớp phải tuân theo quy luật thiết kế, nếu khoảng cách giữa các lớp nhỏ hơn thì sẽ có thông báo.
- Từ schematic và các hiểu biết về cmos, ta có thể thiết kế được layout cho cell SRAM6T như hình dưới :



**Hình 2.16: Sơ đồ layout cell SRAM 6T**



Hình 2.17: Sơ đồ layout cell SRAM 8T

### CHƯƠNG III: KẾT LUẬN

Electric là phần mềm mã nguồn mở dùng để thiết kế layout cho các mạch tích hợp. Trong quá trình tìm hiểu về phần mềm, ta thấy nó có những ưu điểm và khuyết điểm như sau:

- **Ưu điểm:**

+ Dễ cài đặt: có thể cài đặt một cách đơn giản trên cả hệ điều hành windows và linux.

+ Dễ sử dụng: Phần mềm có giao diện tương đối thân thiện với người sử dụng, các thao tác trên phần mềm cũng đơn giản với cách nối dây thông minh, tự động check DRC,..

+ Hỗ trợ nhiều công nghệ thiết kế: Trong electric hỗ trợ nhiều công cụ thiết kế cho phép người sử dụng có thể lựa chọn những công nghệ thích hợp, tuy nhiên công nghệ thường được lựa chọn nhiều nhất là mocmos vì nó hỗ trợ khá đầy đủ các component cho việc thiết kế.

- **Nhược điểm:** Nhược điểm lớn nhất của electric so với các phần mềm thiết kế VLSI khác như cadence, synopsys,.. là không hỗ trợ về phần mô phỏng chưa được hỗ trợ nhiều, nên không có khả năng phân tích triệt để hoạt động của hệ thống thiết kế.

Sau khi hoàn thành bài tập lớn, nhóm đã tìm hiểu được về SRAM như: các khái niệm liên quan, kiến trúc của SRAM, cấu trúc một cell SRAM (6T),.... Kết quả đạt được là nhóm đã hoàn thành được mục tiêu của bài tập lớn, thiết kế và layout một cell SRAM cụ thể là cell SRAM 6T và 8T.

Do thời gian tìm hiểu, kiến thức còn hạn chế nên vẫn còn nhiều sai sót trong quá trình thực hiện vì thế rất mong được sự quan tâm đóng góp ý kiến của thầy và các bạn để chúng em có thể hoàn thành tốt được bài tập cũng như hoàn thiện được kiến thức của mình.

Xin chân thành cảm ơn !

Hà Nội, ngày 15 tháng 4 năm 2013.

Nhóm sinh viên thực hiện.

## TÀI LIỆU THAM KHẢO

[1]- Neil H. E. Weste and David Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 3rd edition, Pearson Education, Inc., 2005.

[2]- <http://www.sciencedirect.com/>

[3]- <https://www.google.com.vn/>