**Bài tiểu luận**

LỊCH SỬ VÀ KIẾN TRÚC

MÁY TÍNH

***NHÓM 6***

***Giáo viên hướng dẫn: PGS.TS. Nguyễn Ái Việt***

***Học viên: Trần Nguyễn Hải Nam (Nhóm trưởng)***

***Vũ Hoàng Phong***

***Lưu Trung Nghĩa***

***Nguyễn Duy Phương***

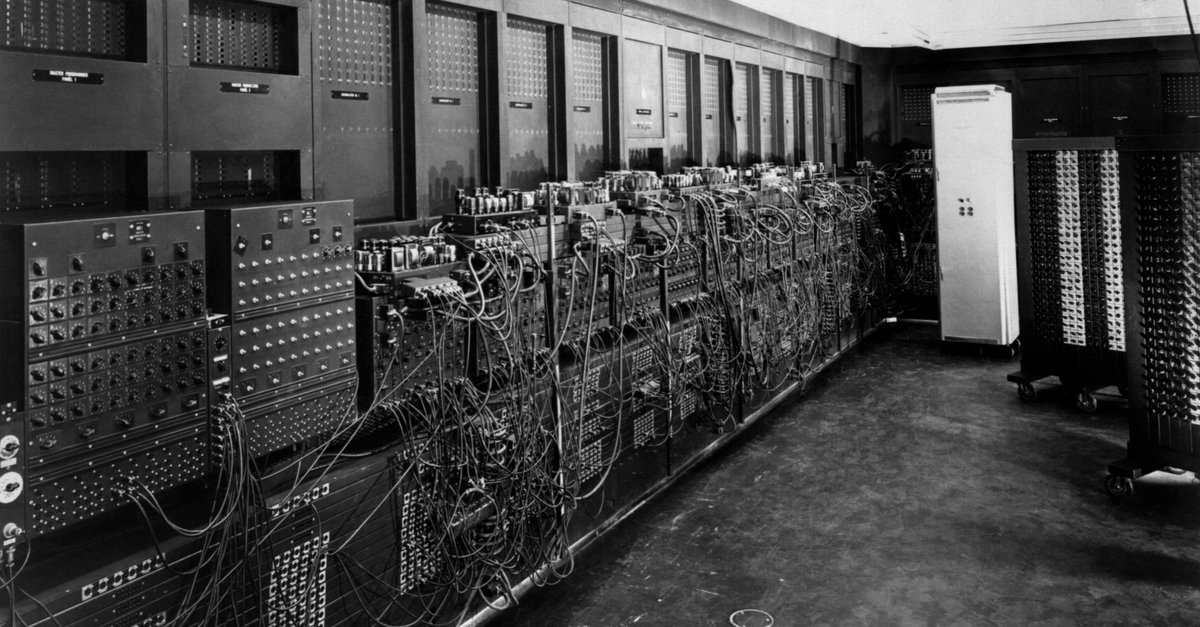
***Lớp CNTT K15 - 05***

**LỜI MỞ ĐẦU.**

Trong lịch sử văn minh nhân loại, con người đã chứng kiến những phát minh, phát kiến có tầm vóc hết sức to lớn, thay đổi hầu như hoàn toàn diện mạo cuộc sống của chúng ta: Công cụ đồ đá đầu tiên, việc sử dụng lửa, các máy cơ học, .... cho đến những phát minh vĩ đại thời gian gần đây như vệ tinh nhân tạo, các nhà máy điện hạt nhân hoặc các tiến bộ trong lĩnh vực sinh học (nhân bản vô tính, sơ đồ gen loài người) ... Từng bước, con người ngày càng cải thiện cuộc sống của mình nhờ vào các công cụ, thiết bị "hiện đại dần theo thời gian".

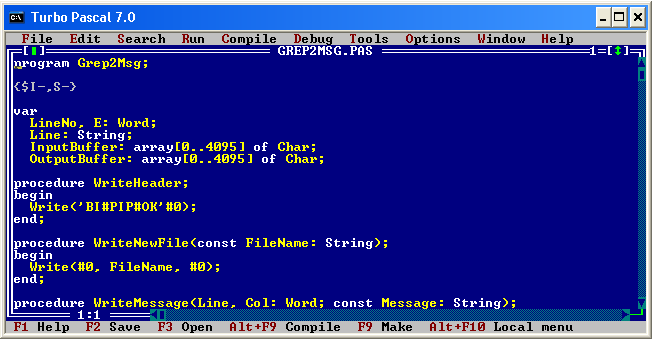
Trong kỉ nguyên "công nghệ - thông tin" hiện nay, máy vi tính (computer) đã trở thành một khái niệm phổ cập, mà sự tồn tại của nó là không thể tách rời với sự tiến bộ của con người. Bản thân sự hình thành và phát triển của máy tính đã là một quá trình chứa đựng rất nhiều nỗ lực cũng như óc sáng tạo của nhiều thế hệ con người. Trong bài tiểu luận này, ta sẽ điểm qua một số cột mốc quan trọng trong lịch sử phát triển máy tính và đưa ra những phân tích dưới góc nhìn của một người làm sáng tạo khoa học.

***A- Chiếc máy tính đầu tiên.***

***1.*** **ENIAC (**viết tắt của cụm từ **Electronic Numerical Intergrator and Computer -** Máy tích hợp [điện tử](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_t%E1%BB%AD) và máy tính) là máy tính điện tử đầu tiên được sử dụng cho mục đích chung, chẳng hạn như giải quyết các vấn đề về số. Nó được phát minh bởi John Presper Eckert và John Mauchly tại Đại học Pennsylvania, để phục vụ cho Phòng thí nghiệm nghiên cứu đạn đạo của quân đội Hoa Kỳ.

Việc xây dựng cỗ máy này bắt đầu vào năm 1943 và nó chính thức hoàn thành vào năm 1946. Mặc dù chưa hoàn thành cho đến khi kết thúc Thế chiến II, nhưng ENIAC đã giúp ích rất nhiều trong cuộc chiến chống lại phát xít Đức.

Năm 1953, Tập đoàn Burroughs đã xây dựng bộ nhớ lõi từ để thêm vào ENIAC. Đến năm 1956, ENIAC chiếm diện tích khoảng 167m² và bao gồm gần 20.000 ống chân không, 1.500 rơle, 10.000 tụ điện và 70.000 điện trở. Hệ thống sử dụng 200 kilowatt điện, nặng hơn 30 tấn và có giá khoảng $487.000.

***2.*** **Pascal** là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) cho [máy tính](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh) thuộc dạng [mệnh lệnh](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_m%E1%BB%87nh_l%E1%BB%87nh) và [thủ tục](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_th%E1%BB%A7_t%E1%BB%A5c), được [Niklaus Wirth](https://vi.wikipedia.org/wiki/Niklaus_Wirth) phát triển vào năm [1970](https://vi.wikipedia.org/wiki/1970). Pascal là ngôn ngữ lập trình đặc biệt thích hợp với kiểu [lập trình cấu trúc](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_c%E1%BA%A5u_tr%C3%BAc) và [cấu trúc dữ liệu](https://vi.wikipedia.org/wiki/C%E1%BA%A5u_tr%C3%BAc_d%E1%BB%AF_li%E1%BB%87u), ngôn ngữ lập trình này được đặt theo tên của nhà toán học, triết gia và nhà vật lí người Pháp [Blaise Pascal](https://vi.wikipedia.org/wiki/Blaise_Pascal).

Pascal cho phép xác định các kiểu dữ liệu phức tạp và xây dựng các cấu trúc dữ liệu động và đệ quy như [danh sách](https://vi.wikipedia.org/w/index.php?title=Danh_s%C3%A1ch_(c%E1%BA%A5u_tr%C3%BAc_d%E1%BB%AF_li%E1%BB%87u)&action=edit&redlink=1), [cây](https://vi.wikipedia.org/wiki/C%C3%A2y_(c%E1%BA%A5u_tr%C3%BAc_d%E1%BB%AF_li%E1%BB%87u)) và [đồ thị](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%93_th%E1%BB%8B_(ki%E1%BB%83u_d%E1%BB%AF_li%E1%BB%87u_tr%E1%BB%ABu_t%C6%B0%E1%BB%A3ng)&action=edit&redlink=1). Pascal có [khả năng xếp kiểu mạnh](https://vi.wikipedia.org/wiki/Ki%E1%BB%83u_m%E1%BA%A1nh_v%C3%A0_ki%E1%BB%83u_y%E1%BA%BFu) trên tất cả các đối tượng, có nghĩa là một loại dữ liệu không thể được chuyển đổi hoặc hiểu như một loại dữ liệu khác nếu không có các chuyển đổi rõ ràng. Không giống như C (và hầu hết các ngôn ngữ trong họ [C](https://vi.wikipedia.org/w/index.php?title=H%E1%BB%8D_C&action=edit&redlink=1)), Pascal cho phép các định nghĩa [thủ tục lồng nhau](https://vi.wikipedia.org/w/index.php?title=Th%E1%BB%A7_t%E1%BB%A5c_l%E1%BB%93ng_nhau&action=edit&redlink=1) ở bất kỳ độ sâu nào, và cũng cho phép hầu hết các loại định nghĩa và khai báo bên trong [chương trình con](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_con) (thủ tục và hàm). Do đó, về mặt cú pháp, một chương trình tương tự như một thủ tục hoặc một hàm. Nó đã được sử dụng rộng rãi như một ngôn ngữ giảng dạy trong các khóa học lập trình cấp [đại](https://vi.wikipedia.org/wiki/Vi%E1%BB%87n_%C4%91%E1%BA%A1i_h%E1%BB%8Dc) học vào những năm 1980, và cũng được sử dụng trong cài đặt sản xuất để viết phần mềm thương mại trong cùng thời kỳ.

Mô tả ngắn gọn Pascal có nguồn gốc từ ngôn ngữ ALGOL 60 cho phép các lập trình viên xác định các kiểu dữ liệu phức tạp và cũng dễ dàng hơn trong việc xây dựng cấu trúc dữ liệu động và đệ quy. Chẳng hạn như danh sách, cây cối và đồ thị. Các tính năng quan trọng được đưa vào đây là các bản ghi, liệt kê, phân nhóm, các biến phân bổ động với con trỏ liên quan và tập hợp. Để làm cho điều này có thể và có ý nghĩa, Pascal có cách [kiểu mạnh](https://vi.wikipedia.org/wiki/Ki%E1%BB%83u_m%E1%BA%A1nh_v%C3%A0_ki%E1%BB%83u_y%E1%BA%BFu) vào tất cả các đối tượng, có nghĩa là một loại dữ liệu không thể được chuyển đổi hoặc được hiểu là một loại dữ liệu khác mà không thể chuyển đổi rõ ràng. Pascal, giống như nhiều ngôn ngữ lập trình ngày nay (nhưng không giống như hầu hết các ngôn ngữ trong họ C), cho phép định nghĩa quy trình lồng nhau ở bất kỳ mức độ sâu nào. Điều này cho phép tạo một cú pháp rất đơn giản và mạch lạc trong đó một chương trình hoàn chỉnh là gần như giống hệt với một thủ tục hoặc hàm duy nhất.

Đặc điểm của Pascal ở dạng nguyên thủy của nó là một ngôn ngữ thuần túy và bao gồm các cấu trúc điều khiển giống như ALGOL truyền thống với các từ dành riêng như **if**, **then**, **else**, **while**, **for** và **case** khác nhau trên một câu lệnh khối lệnh. Pascal cũng có cấu trúc dữ liệu cấu trúc của ngôn ngữ lập trình ALGOL 60 như bản ghi, biến thể, con trỏ, liệt kê và bộ. Những cấu trúc như vậy được lấy cảm hứng từ Simula 67, [ALGOL 68](https://vi.wikipedia.org/wiki/ALGOL_68), ALGOL W của Niklaus Wirth và được đề xuất bởi C. A. R. Hoare.

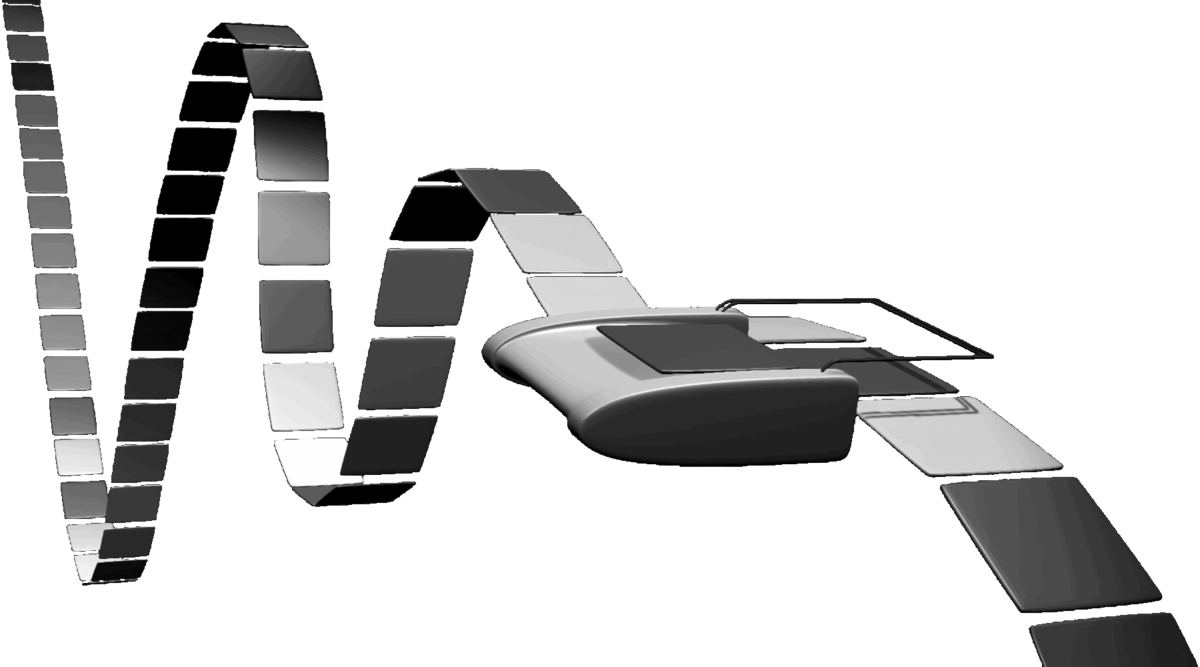
Mặc dù rất phổ biến (những năm 1980 và 1990 còn phổ biến hơn cả thời điểm bài viết này được thực hiện), các phiên bản Pascal trước đây đã bị phê phán rộng rãi vì không phù hợp cho việc sử dụng trong thực tế, ngoài việc dạy học. Mặt khác, rất nhiều nỗ lực phát triển lớn trong những năm 1980 lại phụ thuộc rất nhiều vào Pascal.

***3.*** **Alan Mathison Turing (23/06/1912 – 07/06/1954)** là một [nhà toán học](https://vi.wikipedia.org/wiki/Danh_s%C3%A1ch_nh%C3%A0_to%C3%A1n_h%E1%BB%8Dc), [logic học](https://vi.wikipedia.org/wiki/Logic) và [mật mã học](https://vi.wikipedia.org/wiki/M%E1%BA%ADt_m%C3%A3_h%E1%BB%8Dc) người [Anh](https://vi.wikipedia.org/wiki/Anh), được xem là cha đẻ của ngành [khoa học máy tính](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh). [Phép thử Turing](https://vi.wikipedia.org/wiki/Ph%C3%A9p_th%E1%BB%AD_Turing) (*Turing test*) là một trong những cống hiến của ông trong ngành [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o): thử thách này đặt ra câu hỏi rằng máy móc có khi nào đạt được [ý thức](https://vi.wikipedia.org/wiki/%C3%9D_th%E1%BB%A9c) và có thể [suy nghĩ](https://vi.wikipedia.org/w/index.php?title=Suy_ngh%C4%A9&action=edit&redlink=1) được hay không. Ông đã hình thức hóa khái niệm [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) và tính toán với [máy Turing](https://vi.wikipedia.org/wiki/M%C3%A1y_Turing), đồng thời đưa ra phiên bản của "Turing", mà ngày nay được đông đảo công chúng chấp nhận, về [luận đề Church-Turing](https://vi.wikipedia.org/w/index.php?title=Lu%E1%BA%ADn_%C4%91%E1%BB%81_Church-Turing&action=edit&redlink=1), một luận đề nói rằng tất cả những gì tính được bằng thuật toán đều có thể tính được bằng [máy Turing](https://vi.wikipedia.org/wiki/M%C3%A1y_Turing).

**Máy Turing** là một mô hình về thiết bị xử lý các ký tự, tuy đơn giản, nhưng có thể thực hiện được tất cả các [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) [máy tính](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh). Các máy Turing đã được [Alan Turing](https://vi.wikipedia.org/wiki/Alan_Turing) trình bày vào năm [1936](https://vi.wikipedia.org/wiki/1936). Các máy Turing được xây dựng không dành cho việc trực tiếp chế tạo ra máy tính, mà là dành cho các [thí nghiệm tưởng tượng](https://vi.wikipedia.org/wiki/Th%C3%AD_nghi%E1%BB%87m_t%C6%B0%E1%BB%9Fng_t%C6%B0%E1%BB%A3ng) để tìm hiểu về các giới hạn của việc tính toán trên máy móc. Việc nghiên cứu các tính chất của máy Turing cho biết nhiều kiến thức quan trọng trong lĩnh vực [khoa học máy tính](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh) và [lý thuyết về độ phức tạp tính toán](https://vi.wikipedia.org/wiki/L%C3%BD_thuy%E1%BA%BFt_%C4%91%E1%BB%99_ph%E1%BB%A9c_t%E1%BA%A1p_t%C3%ADnh_to%C3%A1n).

Máy Turing mà có khả năng mô phỏng lại hoạt động của tất cả các máy Turing khác được gọi là [máy Turing vạn năng](https://vi.wikipedia.org/w/index.php?title=M%C3%A1y_Turing_v%E1%BA%A1n_n%C4%83ng&action=edit&redlink=1) (hay đơn giản là *máy vạn năng*). Máy vạn năng cũng đã được [Alonzo Church](https://vi.wikipedia.org/w/index.php?title=Alonzo_Church&action=edit&redlink=1) mô tả, khi xây dựng các lý thuyết về [phép tính lambda](https://vi.wikipedia.org/wiki/Ph%C3%A9p_t%C3%ADnh_lambda). Lý thuyết của Church và Turing được tổng kết lại trong [luận đề Church-Turing](https://vi.wikipedia.org/w/index.php?title=Lu%E1%BA%ADn_%C4%91%E1%BB%81_Church-Turing&action=edit&redlink=1). Luận đề này khẳng định mọi [hàm toán học](https://vi.wikipedia.org/wiki/H%C3%A0m_s%E1%BB%91) tính được thì cũng có thể dùng các máy Turing để tính, và do đó cho phép định nghĩa các khái niệm như sự tính được của hàm hay [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n).

Trong lý thuyết về [ngôn ngữ hình thức](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_h%C3%ACnh_th%E1%BB%A9c), máy Turing có khả năng đoán nhận tất cả các ngôn ngữ sinh bởi [văn phạm cấu trúc câu](https://vi.wikipedia.org/w/index.php?title=V%C4%83n_ph%E1%BA%A1m_c%E1%BA%A5u_tr%C3%BAc_c%C3%A2u&action=edit&redlink=1).

**Máy turing**

Về mặt toán học, máy Turing có thể được định nghĩa bằng một bộ chứa các phần tử sau:

* [Tập](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p) *S* hữu hạn chứa các trạng thái của máy.
* Tập *V* hữu hạn chứa các ký tự ghi trên băng.
* [Hàm](https://vi.wikipedia.org/wiki/%C3%81nh_x%E1%BA%A1) chuyển trạng thái *f*: *S*[×](https://vi.wikipedia.org/wiki/T%C3%ADch_Descartes)*V* → *S*×*V*×{*Phải*, *Trái*}

Ngoài ra có thể định nghĩa thêm:

* Phần tử đặc biệt là *ký tự trống* *B* trong *V*, các ký tự khác trống trong *V* được gọi là các *ký tự đầu vào*.
* Trạng thái đặc biệt là *trạng thái ban đầu* *S*0 trong *S*.
* Các *trạng thái kết thúc* thuộc tập *F* là [tập con](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p_con) trong *S*.

***4.* John von Neumann (**[**28/12**](https://vi.wikipedia.org/wiki/28_th%C3%A1ng_12)**/**[**1903**](https://vi.wikipedia.org/wiki/1903)**–**[**8/2**](https://vi.wikipedia.org/wiki/8_th%C3%A1ng_2)**/**[**1957**](https://vi.wikipedia.org/wiki/1957)**)**  là một [nhà toán học](https://vi.wikipedia.org/wiki/Danh_s%C3%A1ch_nh%C3%A0_to%C3%A1n_h%E1%BB%8Dc) [người Mỹ gốc Hungary](https://vi.wikipedia.org/wiki/Ng%C6%B0%E1%BB%9Di_M%E1%BB%B9_g%E1%BB%91c_Hungary) và là một nhà bác học thông thạo nhiều lĩnh vực đã đóng góp vào [vật lý lượng tử](https://vi.wikipedia.org/wiki/V%E1%BA%ADt_l%C3%BD_l%C6%B0%E1%BB%A3ng_t%E1%BB%AD), [giải tích hàm](https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_t%C3%ADch_h%C3%A0m), [lý thuyết tập hợp](https://vi.wikipedia.org/wiki/L%C3%BD_thuy%E1%BA%BFt_t%E1%BA%ADp_h%E1%BB%A3p), [kinh tế](https://vi.wikipedia.org/wiki/Kinh_t%E1%BA%BF), [khoa học máy tính](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh), [giải tích số](https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_t%C3%ADch_s%E1%BB%91), [động lực học chất lưu](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99ng_l%E1%BB%B1c_h%E1%BB%8Dc_ch%E1%BA%A5t_l%C6%B0u), [thống kê](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_Th%E1%BB%91ng_k%C3%AA) và nhiều lĩnh vực toán học khác.

Đáng chú ý nhất, von Neumann là nhà tiên phong của [máy tính kỹ thuật số](https://vi.wikipedia.org/w/index.php?title=M%C3%A1y_t%C3%ADnh_k%E1%BB%B9_thu%E1%BA%ADt_s%E1%BB%91&action=edit&redlink=1) hiện đại và áp dụng của [lý thuyết toán tử](https://vi.wikipedia.org/wiki/L%C3%BD_thuy%E1%BA%BFt_to%C3%A1n_t%E1%BB%AD) (*operator theory*) vào [cơ học lượng tử](https://vi.wikipedia.org/wiki/C%C6%A1_h%E1%BB%8Dc_l%C6%B0%E1%BB%A3ng_t%E1%BB%AD) (xem [đại số Von Neumann](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BA%A1i_s%E1%BB%91_Von_Neumann&action=edit&redlink=1)), một thành viên của [Dự án Manhattan](https://vi.wikipedia.org/wiki/D%E1%BB%B1_%C3%A1n_Manhattan), người sáng lập ra [lý thuyết trò chơi](https://vi.wikipedia.org/wiki/L%C3%BD_thuy%E1%BA%BFt_tr%C3%B2_ch%C6%A1i) và khái niệm [cellular automata](https://vi.wikipedia.org/w/index.php?title=Cellular_automata&action=edit&redlink=1). Cùng với [Edward Teller](https://vi.wikipedia.org/wiki/Edward_Teller) và [Stanisław Ulam](https://vi.wikipedia.org/wiki/Stanis%C5%82aw_Ulam), von Neumann khám phá ra những bước quan trọng trong [vật lý hạt nhân](https://vi.wikipedia.org/wiki/V%E1%BA%ADt_l%C3%BD_h%E1%BA%A1t_nh%C3%A2n) liên quan đến [phản ứng nhiệt hạch](https://vi.wikipedia.org/wiki/Ph%E1%BA%A3n_%E1%BB%A9ng_t%E1%BB%95ng_h%E1%BB%A3p_h%E1%BA%A1t_nh%C3%A2n) (*thermonuclear*) và [bom khinh khí](https://vi.wikipedia.org/wiki/V%C5%A9_kh%C3%AD_nhi%E1%BB%87t_h%E1%BA%A1ch).

Tên của von Neumann được dùng trong [kiến trúc von Neumann](https://vi.wikipedia.org/wiki/Ki%E1%BA%BFn_tr%C3%BAc_von_Neumann) được dùng trong hầu hết các loại [máy tính](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh), bởi vì các tác phẩm của ông về khái niệm này; mặc dù nhiều người cảm thấy cách đặt tên này đã bỏ qua các đóng góp của [J. Presper Eckert](https://vi.wikipedia.org/w/index.php?title=J._Presper_Eckert&action=edit&redlink=1) và [John William Mauchly](https://vi.wikipedia.org/w/index.php?title=John_William_Mauchly&action=edit&redlink=1) cũng làm việc với các khái niệm đó trong công trình về máy [ENIAC](https://vi.wikipedia.org/wiki/ENIAC). Hầu hết các máy tính tại nhà, [microcomputer](https://vi.wikipedia.org/wiki/Microcomputer), [minicomputer](https://vi.wikipedia.org/wiki/Minicomputer) và máy tính [mainframe](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_l%E1%BB%9Bn) đều là [máy tính von Neumann](https://vi.wikipedia.org/w/index.php?title=M%C3%A1y_t%C3%ADnh_von_Neumann&action=edit&redlink=1). Ông cũng là người sáng lập ra ngành [cellular automata](https://vi.wikipedia.org/w/index.php?title=Cellular_automaton&action=edit&redlink=1) mà không cần dùng đến máy tính, xây dựng những ví dụ đầu tiên của automata có khả năng [tự nhân đôi](https://vi.wikipedia.org/w/index.php?title=T%E1%BB%B1_nh%C3%A2n_%C4%91%C3%B4i&action=edit&redlink=1) chỉ bằng bút chì và giấy vẽ. Khái niệm về một [máy xây dựng tổng quát](https://vi.wikipedia.org/w/index.php?title=M%C3%A1y_x%C3%A2y_d%E1%BB%B1ng_t%E1%BB%95ng_qu%C3%A1t&action=edit&redlink=1) được trích ra từ cuốn *Theory of Self Reproducing Automata* (Lý thuyết Automata tự sinh sản) xuất bản sau khi ông qua đời. Thuật ngữ "[máy von Neumann](https://vi.wikipedia.org/w/index.php?title=M%C3%A1y_von_Neumann&action=edit&redlink=1)" cũng được dùng để chỉ về những loại máy móc có thể tự nhân đôi. Von Neumann chứng minh rằng cách hiệu quả nhất cho các vụ khai mỏ cực lớn như khai mỏ toàn bộ [Mặt Trăng](https://vi.wikipedia.org/wiki/M%E1%BA%B7t_Tr%C4%83ng) hay [vanh đai tiểu hành tinh](https://vi.wikipedia.org/w/index.php?title=Vanh_%C4%91ai_ti%E1%BB%83u_h%C3%A0nh_tinh&action=edit&redlink=1) (*asteroid belt*) có thể đạt được bằng các máy móc tự nhân đôi được, để dựa vào [sự phát triển theo hàm mũ](https://vi.wikipedia.org/w/index.php?title=S%E1%BB%B1_ph%C3%A1t_tri%E1%BB%83n_theo_h%C3%A0m_m%C5%A9&action=edit&redlink=1) của những cơ chế như vậy.

Thêm vào các công trình về kiến trúc máy tính, ông được cho là có ít nhất một đóng góp trong việc nghiên cứu thuật toán. [Donald Knuth](https://vi.wikipedia.org/wiki/Donald_Knuth), vào năm [1945](https://vi.wikipedia.org/wiki/1945), đã dùng von Neumann như là người phát minh ra thuật toán được nhiều người biết đến của Knuth: thuật toán [sắp xếp trộn](https://vi.wikipedia.org/wiki/S%E1%BA%AFp_x%E1%BA%BFp_tr%E1%BB%99n) (*merge sort*), trong đó hai nửa của một mảng được sắp xếp một cách đệ quy rồi trộn lại với nhau.

Ông cũng bận rộn trong việc thăm dò các bài toán trong lãnh vực [thủy khí động lực diễn toán](https://vi.wikipedia.org/w/index.php?title=Th%E1%BB%A7y_kh%C3%AD_%C4%91%E1%BB%99ng_l%E1%BB%B1c_di%E1%BB%85n_to%C3%A1n&action=edit&redlink=1). Cùng với [R. D. Richtmyer](https://vi.wikipedia.org/w/index.php?title=R._D._Richtmyer&action=edit&redlink=1) ông phát triển khái niệm [độ nhớt nhân tạo](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%99_nh%E1%BB%9Bt_nh%C3%A2n_t%E1%BA%A1o&action=edit&redlink=1), sau đó trở thành cơ sở cho việc nghiên cứu [sóng chấn động](https://vi.wikipedia.org/wiki/S%C3%B3ng_xung_k%C3%ADch). Công bằng mà nói, chúng ta có thể sẽ không biết nhiều về vật lý thiên văn, và đã không phát triển được các loại [động cơ phản lực](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99ng_c%C6%A1_ph%E1%BA%A3n_l%E1%BB%B1c) và [tên lửa](https://vi.wikipedia.org/wiki/T%C3%AAn_l%E1%BB%ADa), nếu không có công trình đó. Vấn đề được giải quyết khi máy tính cố gắng giải các bài toán thủy khí động lực học, các chương trình đặt quá nhiều điểm lưới tại những điểm mà sóng chấn động là không liên tục. Khái niệm "độ nhớt nhân tạo" là một kĩ thuật tính toán nhằm làm trơn phần nào sự truyền chấn động mà không vi phạm nhiều về mặt cơ sở vật lý.



***B- Máy tính cá nhân ra đời (personal computer)***

Khi máy tính cá nhân lần đầu tiên được phát minh, bộ xử lý trung tâm (CPU) của chúng chỉ đứng một mình và chỉ có một lõi xử lý. Bản thân bộ xử lý là cốt lõi; ý tưởng về việc có một bộ xử lý đa lõi vẫn chưa từng nghe thấy. Ngày nay, không có gì lạ khi thấy máy tính, điện thoại và các thiết bị khác có nhiều lõi - thực sự, hầu như mọi máy tính có sẵn trên thị trường đều có nhiều lõi. Các lõi này nằm trong cùng, Đơn vị, CPU hoặc Bộ xử lý trung tâm.

Có nhiều lõi là một lợi thế lớn. Chỉ với một lõi, một máy tính chỉ có thể hoạt động trên một nhiệm vụ tại một thời điểm, phải hoàn thành một nhiệm vụ trước khi nó chuyển sang một nhiệm vụ khác. Tuy nhiên, với nhiều lõi hơn, một máy tính có thể hoạt động trên nhiều tác vụ cùng một lúc, điều này đặc biệt hữu ích cho những người thực hiện nhiều thao tác đa nhiệm.

**Bộ xử lý đa lõi**

Có nhiều lõi cho phép thực hiện nhiều việc cùng một lúc. Ví dụ: nếu bạn đang làm việc trên email, mở trình duyệt Internet, đang làm việc trên bảng tính excel và đang nghe nhạc trong iTunes, thì bộ xử lý lõi tứ có thể hoạt động trên tất cả những điều này cùng một lúc. Hoặc, nếu người dùng có một nhiệm vụ cần phải hoàn thành ngay lập tức, nó có thể được chia thành các nhiệm vụ nhỏ hơn, dễ dàng hơn để xử lý.

Sử dụng nhiều lõi cũng không chỉ giới hạn ở nhiều chương trình. Ví dụ: Google Chrome kết xuất mỗi trang mới với một quy trình khác nhau, có nghĩa là nó có thể tận dụng nhiều lõi cùng một lúc. Tuy nhiên, một số chương trình là những gì được gọi là đơn luồng, có nghĩa là chúng không được viết để có thể sử dụng nhiều lõi và vì vậy không thể làm như vậy. Siêu phân luồng một lần nữa xuất hiện ở đây, cho phép Chrome gửi nhiều trang đến hai lõi logic logic trên một lõi thực tế.

Đi đôi với bộ xử lý đa lõi và siêu phân luồng là một khái niệm gọi là đa luồng. Đa luồng về cơ bản là khả năng cho một hệ điều hành tận dụng nhiều lõi bằng cách tách mã thành dạng cơ bản nhất hoặc các luồng và đưa nó vào các lõi khác nhau cùng một lúc. Điều này, tất nhiên, quan trọng trong bộ xử lý đa cũng như bộ xử lý đa lõi. Đa luồng phức tạp hơn một chút so với âm thanh của nó, vì nó yêu cầu các hệ điều hành phải đặt mã đúng theo cách mà chương trình có thể tiếp tục chạy hiệu quả.

## Làm thế nào để bộ xử lý đa lõi làm việc?

Đầu tiên, bo mạch chủ và hệ điều hành cần nhận ra bộ xử lý và có nhiều lõi. Các máy tính cũ chỉ có một lõi, vì vậy một hệ điều hành cũ hơn có thể không hoạt động tốt nếu người dùng cố gắng cài đặt nó trên một máy tính mới hơn có nhiều lõi. Windows 95, ví dụ, không hỗ trợ siêu phân luồng hoặc nhiều lõi. Tất cả các hệ điều hành gần đây đều hỗ trợ bộ xử lý đa lõi, bao gồm cả Windows 7, 8, 10 mới được phát hành và OS X 10.10 của Apple.

Về cơ bản, hệ điều hành sau đó nói với bo mạch chủ rằng một quy trình cần phải được thực hiện. Các bo mạch chủ sau đó nói với bộ xử lý. Trong bộ xử lý đa lõi, hệ điều hành có thể yêu cầu bộ xử lý thực hiện nhiều việc cùng một lúc. Về cơ bản, thông qua sự chỉ đạo của hệ điều hành, dữ liệu được chuyển từ ổ cứng hoặc RAM, qua bo mạch chủ, đến bộ xử lý.

**Định luật Moore** được xây dựng bởi [Gordon Moore](https://vi.wikipedia.org/wiki/Gordon_Moore) - một trong những sáng lập viên của tập đoàn sản xuất [chip](https://vi.wikipedia.org/wiki/Chip) máy tính nổi tiếng [Intel](https://vi.wikipedia.org/wiki/Intel). Định luật ban đầu được phát biểu như sau:

"Số lượng [transistor](https://vi.wikipedia.org/wiki/Transistor) trên mỗi đơn vị [inch vuông](https://vi.wikipedia.org/w/index.php?title=Inch_vu%C3%B4ng&action=edit&redlink=1) sẽ tăng lên gấp đôi sau mỗi 24 tháng." (1 inch vuông xấp xỉ 6,45 cm²). Năm [2000](https://vi.wikipedia.org/wiki/2000) định luật được sửa đổi và công nhận là sau mỗi chu kỳ 18 tháng. Tuy nhiên, có một số thông tin cho rằng Gordon Moore đã công bố sửa đổi định luật của ông là 24 tháng nhưng báo chí tại thời điểm đó đã viết là 18 tháng.

**Tác dụng của Định luật Moore**

Định luật Moore là một bước ngoặt lớn trong ngành công nghệ điện tử, giải thích tại sao nhà sản xuất có thể giảm giá thành trong khi vẫn tiếp tục nâng cao hiệu suất của phần cứng.

Định luật Moore cũng là một động lực kích thích cho ngành công nghiệp điện tử duy trì sự phát triển mạnh mẽ trong hàng thập kỷ qua.

Thực tế, từ "định luật" trong tên gọi "Định luật Moore" không mang cùng một ý nghĩa với từ "định luật" trong tên gọi "định luật Newton" hay "định luật Gauss". Vào năm 1965, trong một bài báo có tiêu đề "Đưa thêm nhiều thành phần vào mạch tích hợp" cho tạp chí Electronics, Gordon Moore đưa ra tiên đoán rằng số lượng thành phần (bóng bán dẫn) trong các mạch tích hợp – bộ não của máy tính – sẽ tăng gấp đôi sau mỗi năm. Khoảng 1 thập kỷ sau, tiến sĩ Moore thay đổi lại kết luận của mình, theo đó số lượng bóng bán dẫn sẽ phải mất 2 năm để tăng gấp đôi - con số "18 tháng" được nhiều người đưa ra chưa bao giờ tồn tại trên Định luật Moore.

### Ý nghĩa của định luật Moore đối với ngành công nghiệp điện tử

Định luật Moore được xuất bản lần đầu trên tạp chí Điện tử vào năm 1965. Trong khi Moore không thực sự chú ý nhiều đến dự đoán của riêng mình, lời tuyên bố này đã trở thành một chuẩn mực công nghệ cho ngành công nghiệp bán dẫn.

Tầm quan trọng của định luật này đối với các nhà sản xuất chất bán dẫn là rất rõ ràng. Ngành công nghiệp sản xuất bán dẫn đã tạo ra một lộ trình dự đoán kéo dài gần năm thập kỷ từ năm 1971 đến năm 2020. Bộ tài liệu này có tiêu đề "Lộ trình công nghệ quốc tế cho bóng bán dẫn".

### **Ý nghĩa kinh tế của định luật Moore**

Một trong những tác động kinh tế của định luật Moore là các thiết bị điện toán tiếp tục cho thấy sự tăng trưởng theo cấp số nhân về độ phức tạp và sức mạnh tính toán trong khi giảm chi phí tương đương cho nhà sản xuất và người tiêu dùng.

Do đó, chi phí sản xuất thấp hơn và độ tin cậy của các công nghệ mớin đã dẫn đến sự cải thiện đáng kể về vốn chủ sở hữu và lợi nhuận hoạt động của ngành công nghiệp bán dẫn và cả ngành điện tử.

Định luật Moore có vai trò lớn trong sự phát triển của công nghệ điện toán đám mây và mạng xã hội ngày càng đòi hỏi khả năng tính toán tăng lên. Mối quan hệ kinh tế giữa nhà sản xuất thiết bị hỗ trợ ngành công nghiệp điện tử, nhà sản xuất chip và thị trường tiêu dùng tiếp tục chịu ảnh hưởng bởi khả năng ngành công nghiệp điện tử theo kịp các điều kiện của định luật Moore.

***C- Vi xử lí và bộ nhớ máy tính***

1:Vi điều khiển

   a; Khái niệm vi điều khiển và vi xử lý

* Vi điều khiểnlà một máy tính được tích hợp trên một chip, nó thường được sử dụng để điều khiển các thiết bị điện tử. Vi điều khiển, thực chất, là một hệ thống bao gồm một vi xử lý có hiệu suất đủ dùng và giá thành thấp (khác với các bộ vi xử lý đa năng dùng trong máy tính) kết hợp với các khối ngoại vi như bộ nhớ, các module vào/ra, các module biến đổi số sang tương tự và tương tự sang số,... Ở máy tính thì các module thường được xây dựng bởi các chip và mạch ngoài.
* Vi xử lý (viết tắt là µP hay uP), đôi khi còn được gọi là bộ vi xử lý, là một linh kiện điện tử máy tính được chế tạo từ các transistor thu nhỏ tích hợp lên trên một vi mạch tích hợp đơn. ... Do đó, một CPU có thể là một bảng mạch gồm hàng ngàn hay hàng triệu vi mạch tích hợp.

   b; Lịch sử,các họ vi điều khiển phổ biến

Lịch sử:

* Trong những thập niên cuối thế kỉ XX, bước đột phá lớn nhất trong công nghệ điện tử Intel cho ra đời bộ vi xử lý đầu tiên.
* Phát minh đầu tiên này đã cho phép tạo ra các ứng dụng theo một công thức khác : phần cứng + phần mền = ứng dụng.
* Bộ xử lý sau đó phát triển theo hai hướng tăng tốc độ xử lý , tích hợp trên bộ nhớ.

Các vi điều khiển thông dụng

* Dòng 8051 (8031, 8051, 8751, 8951, 8032, 8052, 8752, 8952)
* Dòng Atmel AT91 (Kiến trúc ARM THUMB)
* Dòng AT90, Tiny & Mega – AVR (Atmel Norway design)
* Dòng Atmel AT89 (Kiến trúc Intel 8051/MCS51)
* Dòng MARC4.

 c; So sánh vi điều khiển và vi xử lý

|  |  |  |
| --- | --- | --- |
| Cơ sở để so sánh | Bộ vi xử lý | Vi điều khiển |
| Căn bản | Được tạo thành từ một chip silicon duy nhất bao gồm ALU, CU và các thanh ghi. | Bao gồm bộ vi xử lý, bộ nhớ, cổng I / O, bộ điều khiển ngắt, v.v. |
| Đặc điểm | Đơn vị phụ thuộc | Đơn vị độc lập |
| Cổng I / O | Không chứa cổng I / O tích hợp | Cổng I / O tích hợp có mặt |
| Loại hoạt động  được thực hiện | Mục đích chung trong  thiết kế và vận hành. | Ứng dụng theo định hướng hoặc tên miền cụ thể. |
| Nhắm mục tiêu cho | Thị trường cao cấp | Thị trường nhúng |
| Sự tiêu thụ  năng lượng | Cung cấp ít tùy chọn  tiết kiệm năng lượng | Bao gồm nhiều tùy chọn tiết kiệm năng lượng hơn |

2:CPU

a;CPU là gì

     CPU là chữ viết tắt của Central Processing Unit hay còn gọi là bộ xử lý trung tâm. CPU đóng vai trò như não bộ của một chiếc Laptop, tại đó mọi thông tin, thao tác, dữ liệu sẽ được tính toán kỹ lưỡng và đưa ra lệnh điều khiển mọi hoạt động của Laptop.

b; Lịch sử phát triển CPU

     1958 – Đầu tiên mạch tích hợp lần đầu tiên được phát triển bởi Robert Noyce. IC đầu tiên được công bố chính thức vào ngày 12 tháng 9 năm 1958. ... Intel giới thiệu bộ xử lý 8008 vào ngày 1 tháng 4 năm 1972, chip cải tiến được giới thiệu vào 2 năm sau. 2 năm sau vào năm 1976 bộ xử lý 8085 được giới thiệu.

 c; CPU đơn nhân,CPU đa nhân

* Đơn nhân – một bộ xử lý trên một die. Kể từ năm 2012, thậm chí hầu hết CPU điện thoại thông minh được bán đều không còn là đơn nhân; vi điều khiển vẫn còn đơn nhân trừ một vài ngoại lệ.
* Đa nhân – một "vài" bộ xử lý trên một die, chẳng hạn 2, 4, 8.

3: GPU

  a;GPU là gì, lịch sử phát triển GPU

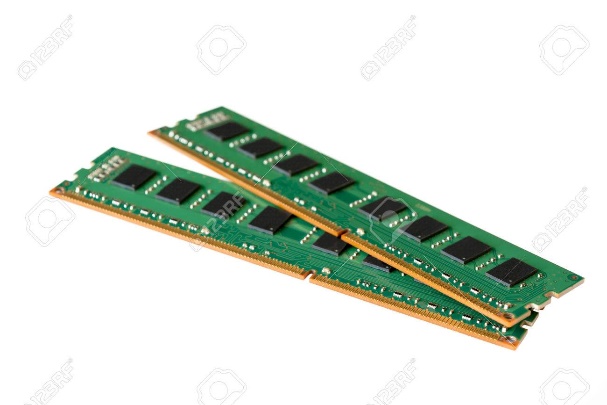
* GPU: Bộ phận xử lý đồ họa (GPU, graphics processing unit), đôi khi được gọi là bộ phận xử lý hình ảnh (VPU, visual processing unit), là một mạch điện tử tích hợp chuyên dụng được thiết kế để thao tác và truy cập bộ nhớ đồ họa một cách nhanh chóng, để tăng tốc việc tạo ra các hình ảnh trong bộ đệm khung hình dành cho ngõ ra ...
* Lịch sử: Thuật ngữ bộ xử lý đồ họa GPU được Nvidia đưa ra năm 1999, đánh dấu bằng sự ra đời của GeForce 256 là "GPU đầu tiên trên thế giới". Nó được trình bày như là một "bộ vi xử lý chip đơn với biến đổi tích hợp, nhẹ, thiết lập tam giác / clipping, và rendering engines".

 b; Khác biệt của CPU và GPU

* [GPU](https://fptshop.com.vn/tin-tuc/danh-gia/tat-tan-tatve-chip-xu-ly-do-hoa-gpu-tren-dien-thoai-129564) (đơn vị xử lý đồ họa) thường được thiết kế để xử lý nhiều tác vụ nhỏ đồng thời. Vì các hiệu ứng đồ họa thường yêu cầu nhiều tác vụ chạy đồng thời, lý tưởng nhất là xử lý đồ họa cho một GPU chuyên dụng, giải phóng CPU để thực hiện các lệnh khác.
* Điểm mấu chốt là CPU có thể làm nhiều việc nhưng phải làm từng việc một trong khi GPU tập trung vào một công việc cụ thể nhưng có thể thực hiện đồng thời nhiều khía cạnh của công việc đó.

  c; Ứng dụng chính của GPU

     Xử lý các tác vụ về hình ảnh, tính toán đồ họa, trong khi một chiếc máy tính cần xử lý nhiều hơn thế, do đó GPU chưa thể thay thế cho CPU. Tuy nhiên, GPU thì khác, nó không chỉ đảm nhận vai trì xử lý hình ảnh, mà còn thực hiện những tính toán phức tạp và thông minh không khác gì CPU.

4: RAM

  a; Khái niệm của RAM, lịch sử RAM

* RAM được viết tắt từ Random Access Memory - một trong những yếu tố hết sức quan trọng bên cạnh vi xử lý. RAM là bộ nhớ tạm của máy giúp lưu trữ thông tin hiện hành để CPU có thể truy xuất và xử lý. ... Nếu như thiết bị bị mất nguồn, tắt máy thì dữ liệu trên RAM sẽ bị xóa.
* Lịch sử của RAM: Chip RAM có mặt trên thị trường những năm 1960, thập niên 90 RAM SDR với tốc độ khá chậm và bộ nhớ rất ít được sản xuất,2000 nghiên cứu sản xuất RAM DDR, năm 2000 đến cuối 2004 DDR SD RAM được sử dụng rộng rãi, Vào năm [2007](https://wiki.edu.vn/wiki/index.php?title=2007&action=edit&redlink=1) cùng với sự ra đời của thế hệ HĐH mới như [Windows Vista](https://wiki.edu.vn/wiki/index.php?title=Windows_Vista), [Mac OS X Leopard](https://wiki.edu.vn/wiki/index.php?title=Mac_OS_X_Leopard&action=edit&redlink=1)người ta bắt đầu sản xuất thế hệ RAM tiếp theo của DDR2 là DDR3 với tốc độ rất nhanh và bộ nhớ lớn đồng thời tiết kiệm năng lượng hơn 30% so với thế hệ DDR2.2009 thì DDR3 mới bắt đầu xuất hiện rộng rãi trên Laptop. Hiện tại, DDR3 đang được giới hạn chủ yếu ở 4 mức xung nhịp 1333, 1600, 1866 MHz. Mức 2133 MHz đang là mức giới hạn xung nhịp về lý thuyết cho DDR3, trong khi các mức 800 MHz và 1066 MHz giờ đã không còn được tiếp tục sản xuất.

  b; Đặc trưng, phân loại RAM

Bộ nhớ RAM có 4 đặc trưng sau:

* Dung lượng bộ nhớ: Tổng số byte của bộ nhớ (nếu tính theo byte) hoặc là tổng số bit trong bộ nhớ nếu tính theo bit.
* Tổ chức bộ nhớ: Số ô nhớ và số bit cho mỗi ô nhớ
* Thời gian thâm nhập: Thời gian từ lúc đưa ra địa chỉ của ô nhớ đến lúc đọc được nội dung của ô nhớ đó.
* Chu kỳ bộ nhớ: Thời gian giữa hai lần liên tiếp thâm nhập bộ nhớ

Phân loại: Tùy theo công nghệ chế tạo của các nước trên thế giới, được chia làm hai loại.

* RAM tĩnh [nhỏ|phải|267px| 6 transistor trong một ô nhớ của RAM tĩnh](https://wiki.edu.vn/wiki/index.php?title=T%E1%BA%ADp_tin:6t-SRAM-cell.png&action=edit&redlink=1) RAM tĩnh - SRAM (Static Random Access Memory) được chế tạo theo công nghệ ECL (dùng trong CMOS và BiCMOS). Mỗi bit nhớ gồm có các cổng logic với 6 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. nhưng sram là một nơi lưu trữ các tập tin của CMOS dùng cho việc khởi động máy
* RAM động [nhỏ|phải|267px| 1 transistor và 1 tụ điện trong một ô nhớ của RAM động](https://wiki.edu.vn/wiki/index.php?title=T%E1%BA%ADp_tin:DRAM-Cell.png&action=edit&redlink=1) RAM động - DRAM (Dynamic Random Access Memory) 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.

c; Các thương hiệu RAM chính, xu thế phát triển RAM.

Các thương hiệu RAM chính:

* 2.1.Corsair Vengeance LED.
* 2.2.G.Skill Trident Z RGB.
* 2.3.Kingston - thương hiệu phổ biến.
* 2.4.Corsair Dominator Platinum.
* 2.5.Team T-Force Night Hawk.
* 2.6.HyperX.

Xu thế triển RAM: Với sự tiến bộ của công nghệ ngày nay, MTXT đã có cấu hình mạnh xấp xỉ với máy tính để bàn, chúng ta dễ dàng mua được một laptop có BXL 4 nhân, 8GB hoặc 16GB [RAM](https://tinhte.vn/tag/ram), ổ cứng hàng Terabyte và card đồ họa cực mạnh (ví dụ nVIDIA GTX 680M hoặc AMD HD 7970M). Tuy nhiên, với các MTXT đời cũ, nếu chủ nhân của nó không có điều kiện sắm máy tính mới, thì nâng cấp cũng là một trong các giải pháp để tăng hiệu năng cho chúng, ví dụ như thay SSD, hoặc cách đơn giản nhất là thêm RAM. Bài viết này sẽ giới thiệu đến bạn các bước cơ bản để nâng cấp RAM cho chiếc laptop của mình.



5: SSD

a; Khái niệm của SSD và lịch sử của SSD

SSD (Solid State Drive) là một loại phương tiện lưu trữ dữ liệu liên tục trên bộ nhớ flash trạng thái rắn. ... Ổ cứng SSD không chỉ cải thiện về sức mạnh tốc độ so với phiên bản ổ cứng HDD truyền thống. SSD còn có giúp người dùng cải thiện nhiệt độ, độ an toàn dữ liệu và cả về điện năng tiêu thụ.

Lịch sử của SSD: Năm 1983, phiên bản Sharp PC-5000 được giới thiệu với ổ lưu trữ SSD 128Kb. Đến năm 1996, ổ cứng SSD với công nghệ bộ nhớ flash được ra đời. ... Bởi nó giải quyết được những vấn đề của HDD truyền thống. Năm 2009, SSD dần được sử dụng nhiều trong laptop mặc cho giá cả của SSD lúc này tương đối cao hơn so với HDD.

b; Khác biệt giữa SSD và HDD

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Giá cả** | **Tốc độ** | **Độ bền** | **Công suất**  **cao nhất** | **Hiệu suất (truy cập dữ liệu, khởi động,**  **chuyển tập tin)** | **Chống phân mảnh** | **Hiệu quả năng**  **lượng** |
| **Ổ cứng**  **(HDD)** | Giá rẻ hơn | Chậm hơn | Ít bền hơn | 10TB | Thấp hơn | Cần thiết | Sử dụng nhiều năng lượng hơn |
| **Ổ đĩa trạng**  **thái rắn (SSD)** | Đắt hơn | Nhanh hơn | Bền hơn | 4TB | Cao hơn | Không cần  thiết | Sử dụng ít năng  lượng |

c; Thương hiệu SSD chính, xu thế SSD

Thương hiệu SSD chính:

* Ổ cứng SSD Sata III 2.5 inch 500GB Samsung 860 Evo. ...
* Ổ cứng SSD Sata III 2.5 Inch 120GB Apacer AS340. ...
* Ổ cứng SSD SanDisk Plus 120GB Up to 530/400 MB/s. ...
* Ổ cứng SSD Sata III 2.5 inch 250GB Samsung 860 Evo. ...
* Ổ cứng SSD WD Green 120GB 3D Nand WDS120G2G0A. ...

**Xu thế** SSD: Ổ cứng SSD không chỉ cải thiện về sức mạnh tốc độ so với phiên bản [ổ cứng HDD](https://www.dienmayxanh.com/kinh-nghiem-hay/o-cung-hdd-la-gi-co-bao-nhieu-loai-co-nen-mua-khon-1207380)

truyền thống. SSD còn có giúp người dùng cải thiện nhiệt độ, độ an toàn dữ liệu và cả về điện năng tiêu thụ.

**Lời kết**

Qua bài tiểu luận, ta có thể thấy, sự hình thành và phát triển của máy tính là cả một quá trình lâu dài và hàm chứa rất nhiều sự sáng tạo, đổi mới. Dưới góc nhìn của một người làm khoa học tài liệu đã phân tích, đồng thời chỉ rõ các phương pháp sáng tạo được áp dụng cho máy tính, từ đó giúp người đọc hiểu rõ hơn về các phương pháp sáng tạo trong khoa học cũng như sự linh hoạt, uyển chuyển khi áp dụng chúng.