

Chương 1

TỔNG QUAN VỀ ARM

1.1 GIỚI THIỆU VỀ VI XỬ LÝ ARM

Cấu trúc ARM (viết tắt từ tên gốc là **Acorn RISC Machine**) là một loại cấu trúc vi xử lý 32 bit kiểu RISC(thuộc kiến trúc Harvard, có tập lệnh rút gọn) được sử dụng rộng rãi trong các thiết kế nhúng. Do có đặc điểm tiết kiệm năng lượng, các bộ CPU ARM chiếm ưu thế trong các sản phẩm điện tử di động, mà với các sản phẩm này việc tiêu tán công suất thấp là một mục tiêu thiết kế quan trọng hàng đầu.

Ngày nay ARM được ứng dụng và rộng rãi trên mọi lĩnh vực của đời sống: Robot, máy tính, điện thoại, xe hơi, máy giặt...



Hình 1.1 Một số ứng dụng của ARM

1.2 LỊCH SỬ PHÁT TRIỂN ARM

Việc thiết kế ARM được bắt đầu từ năm 1983 trong một dự án phát triển của công ty máy tính Acorn. Nhóm thiết kế hoàn thành việc phát triển mẫu gọi là **ARM1** vào năm 1985, và vào năm sau, nhóm hoàn thành sản phẩm “thực” gọi là **ARM2** với thiết kế đơn giản chỉ gồm 30.000 transistor. ARM2 có tuyến dữ liệu 32 bit, không gian địa chỉ 26 bit tức cho phép quản lý đến 64 Mbyte địa chỉ và 16 thanh ghi 32 bit. Thế hệ sau, **ARM3**, được tạo ra với 4KB cache và có chức năng được cải thiện tốt hơn nữa.

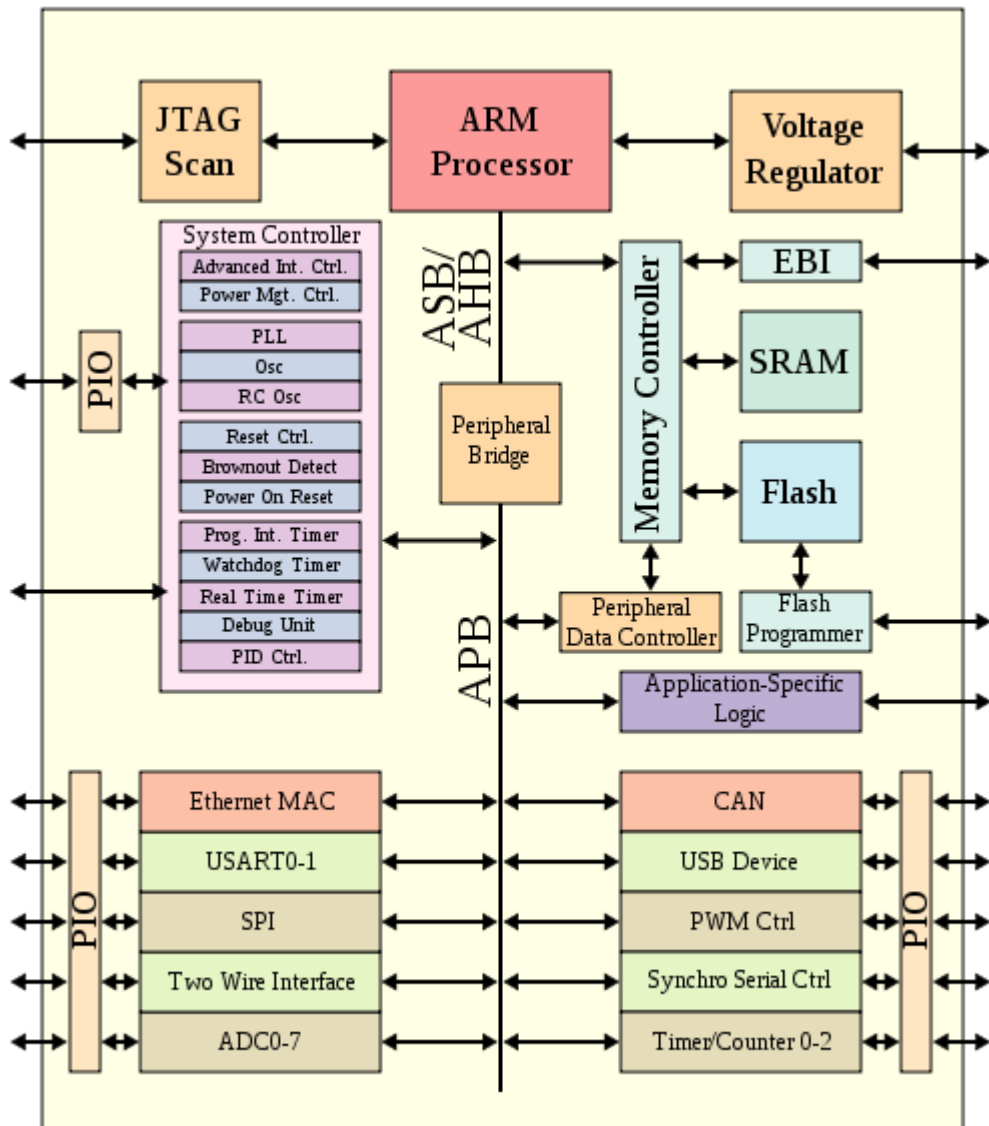
Vào những năm cuối thập niên 80, hãng máy tính Apple Computer và hãng VLSI Technology bắt đầu hợp tác với Acorn để phát triển các thế hệ lõi ARM mới. Kết quả sự hợp tác này là **ARM6**. Mẫu đầu tiên được công bố vào năm 1992 và Apple đã sử dụng bộ vi xử lý ARM 610 dựa trên ARM6 làm cơ sở cho PDA hiệu Apple Newton. Vào năm 1994, Acorn dùng ARM 610 làm CPU trong các máy vi tính RiscPC của họ.

Trải qua nhiều thế hệ nhưng lõi ARM gần như không thay đổi kích thước. ARM2 có 30.000 transistors trong khi ARM6 chỉ tăng lên đến 35.000. Ý tưởng của nhà sản xuất lõi ARM là sao cho người sử dụng có thể ghép lõi ARM với một số bộ phận tùy chọn nào đó để tạo ra một CPU hoàn chỉnh, một loại CPU mà có thể tạo ra trên những nhà máy sản xuất bán dẫn cũ và vẫn tiếp tục tạo ra được sản phẩm với nhiều tính năng mà giá thành vẫn thấp.

Bảng 1.1 Các dòng phát triển của ARM

Kiến Trúc	Số Bit	Tên Lõi
ARMv1	32/26	ARM1
ARMv2	32/26	ARM2, ARM3
ARMv3	32	ARM6, ARM7
ARMv4	32	ARM8
ARMv4T	32	ARM7TDMI, ARM9TDMI
ARMv5	32	ARM7EJ, ARM9E, ARM10E
ARMv6	32	ARM11
ARMv6-M	32	ARM Cortex-M0, ARM Cortex-M0+, ARM Cortex-M1
ARMv7-M	32	ARM Cortex-M3
ARMv7E-M	32	ARM Cortex-M4
ARMv7-R	32	ARM Cortex-R4, ARM Cortex-R5, ARM Cortex-R7
ARMv7-A	32	ARM Cortex-A5, ARM Cortex-A7, ARM Cortex-A8, ARM Cortex-A9, ARM Cortex-A12, ARM Cortex-A15, ARM Cortex-A17
ARMv8-A	64/32	ARM Cortex-A53, ARM Cortex-A57

1.3 KIẾN TRÚC CỦA ARM



Hình 1.2 Kiến trúc của ARM

1.4 GIỚI THIỆU VỀ ARM CORTEX

Như chúng ta đã thấy trong phần giới thiệu lịch sử phát triển các dòng ARM thì lõi Cortex là lõi nhúng kế thừa các ưu điểm từ các thế hệ lõi ARM11 về trước đó. Để phù hợp với nhu cầu sử dụng, ARM Cortex được chia làm 3 dòng:

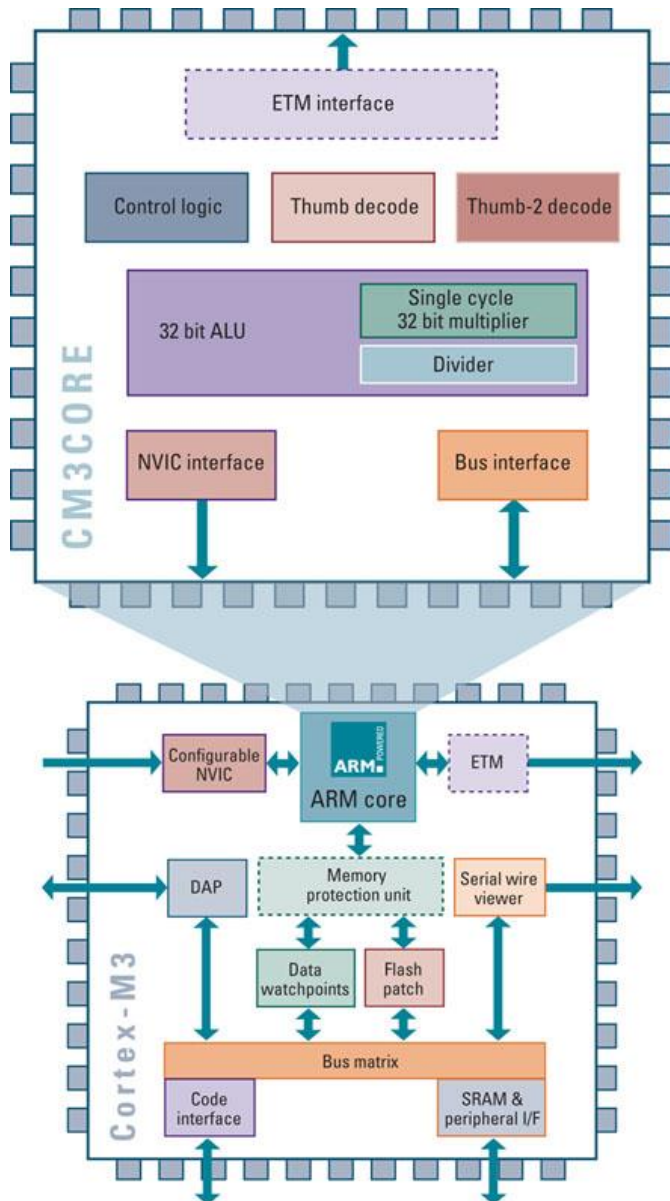
- Cortex-A: Bộ xử lý dành cho hệ điều hành và các ứng dụng phức tạp. Hỗ trợ tập lệnh ARM, thumb, và thumb-2.
- Cortex-R: Bộ xử lý dành cho hệ thống đòi hỏi khắc khe về đáp ứng thời gian thực. Hỗ trợ tập lệnh ARM, thumb, và thumb-2.
- **Cortex-M:** Bộ xử lý dành cho dòng vi điều khiển, được thiết kế để tối ưu về giá thành. Hỗ trợ tập lệnh Thumb-2. (Dòng ARM STM32 có lõi Cortex-M)

Giá trị số nằm cuối tên của 1 dòng ARM (ví dụ ARM Cortex-M3) cho biết về mức độ hiệu suất tương đối của dòng đó. Theo đó dòng ARM mang số 0 sẽ có hiệu suất thấp nhất.

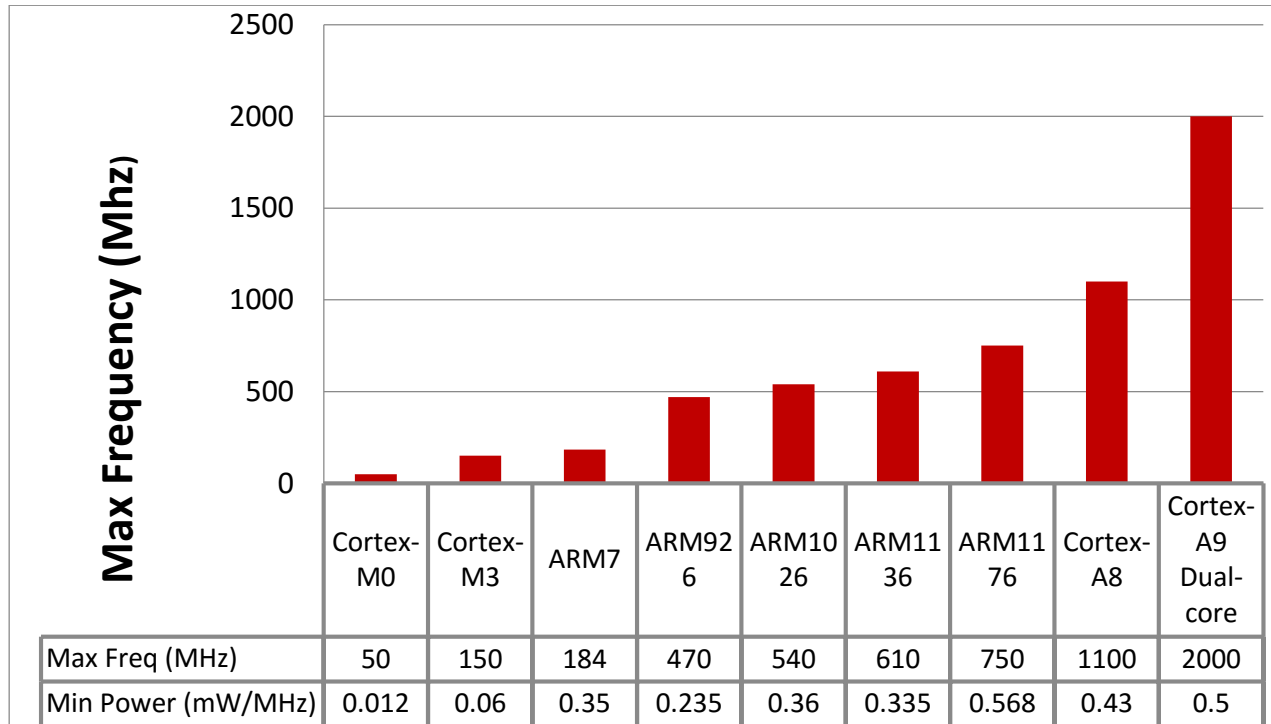
1.5 GIỚI THIỆU VỀ ARM CORTEX-M3

ARM Cortex-M3 được thiết kế đặc biệt để nâng cao hiệu suất hệ thống kết hợp với tiêu thụ năng lượng thấp. ARM Cortex-M3 được thiết kế trên nền kiến trúc mới do đó chi phí sản xuất đủ thấp để cạnh tranh với các dòng vi điều khiển 8 và 16 bit truyền thống. ARM Cortex-M3 có những đặc điểm sau:

- ARM Cortex - M3 được xây dựng dựa trên kiến trúc ARMv7-M 32 bit
- Kiến trúc Harvard tách biệt Bus dữ liệu và lệnh
- Đơn vị bảo vệ bộ nhớ (MPU- Memory Protection Unit) : hỗ trợ bảo vệ bộ nhớ thông qua việc phân quyền thực thi và truy xuất.
- Bộ vi xử lý Cortex-M3 hỗ trợ kiến trúc tập lệnh Thumb-2.
- Hỗ trợ kỹ thuật Bit Band giúp cho phép truy xuất dữ liệu theo bit đồng thời giảm thời gian truy xuất.
- Cho phép truy cập dữ liệu không xếp hàng (unaligned data accesses) đặc điểm này cho phép sử dụng hiệu quả SRAM nội.
- SysTick timer 24 bit hỗ trợ cho việc chạy hệ điều hành thời gian thực.
- Hỗ trợ lập trình và gỡ rối qua cổng JTAG truyền thống cũng như chuẩn 2 dây nhỏ gọn SWD (Serial Wire Debug).
- Khối quản lý vector ngắt lồng nhau (NVIC- Nested Vectored Interrupt Controller) cho phép rút ngắn thời gian đáp ứng yêu cầu ngắt.



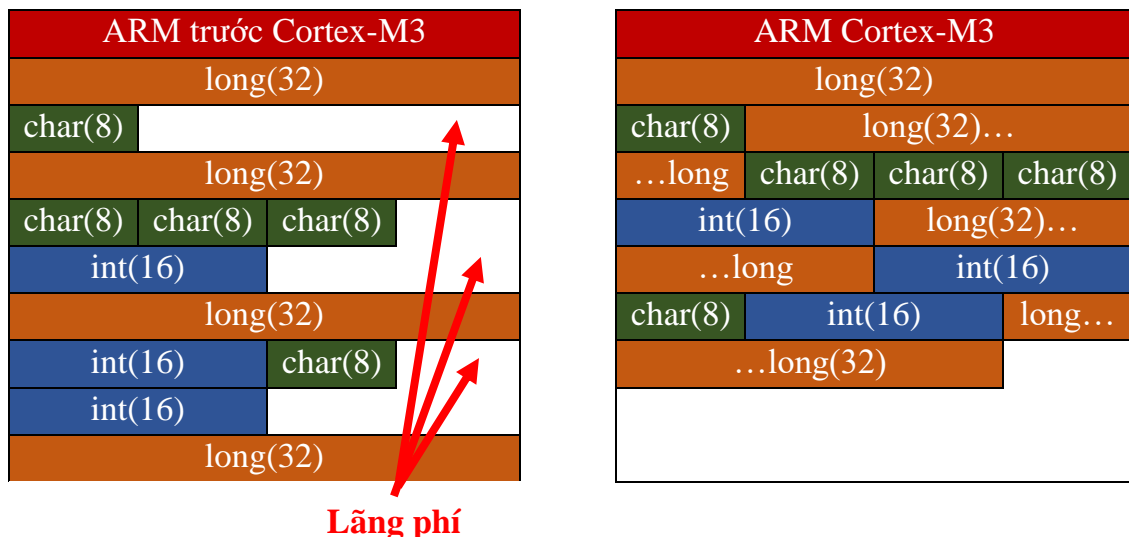
Hình 1.3 Sơ đồ khối ARM Cortex – M3



Hình 1.4 So sánh tốc độ và công suất tiêu thụ của các dòng ARM

1.6 TRUY XUẤT DỮ LIỆU KHÔNG XẾP HÀNG (UNALIGNED DATA ACCESSES)

Hình 1.5 minh họa khả năng truy xuất dữ liệu không xếp hàng giúp sử dụng bộ nhớ SRAM hiệu quả hơn

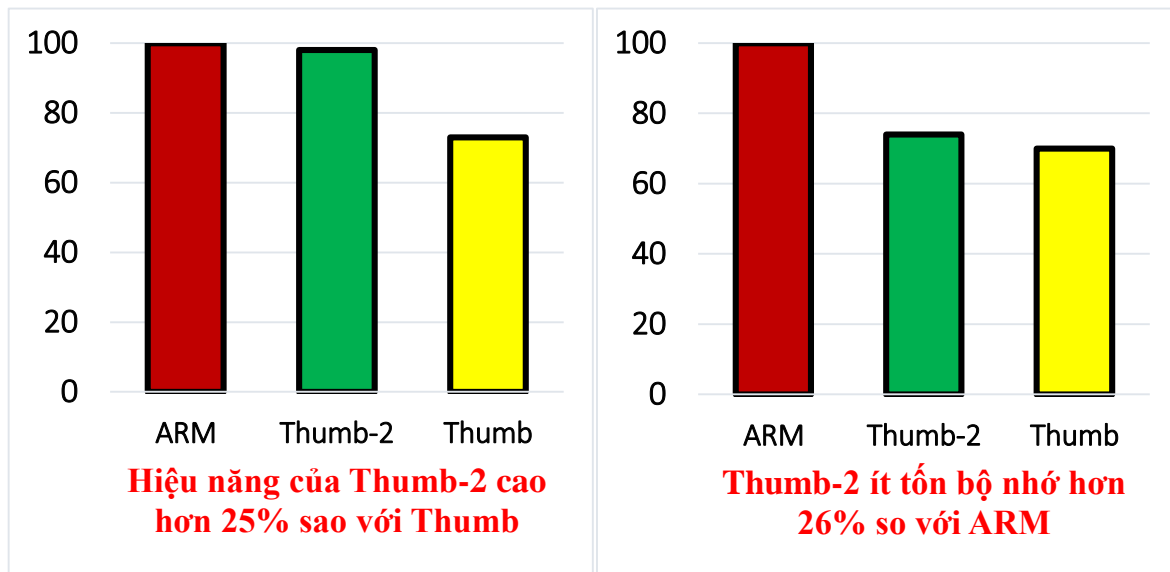


Hình 1.5 Truy xuất dữ liệu không xếp hàng của ARM Cortex-M3 so với thế hệ trước.

1.7 TẬP LỆNH THUMB-2 (THUMB-2 INSTRUCTIONS)

ARM Cortex-M3 là cấu hình vi điều khiển của kiến trúc ARMV7 và khác với các kiến trúc ARM trước đó ở chỗ nó chỉ hỗ trợ tập lệnh Thumb-2. Tập lệnh Thumb-2 là sự pha trộn giữa

tập lệnh 16 và 32 bit, đạt được hiệu suất của các lệnh ARM 32 bit, đồng thời chiếm dung lượng bộ nhớ ít cũng như tương thích ngược với tập lệnh gốc Thumb 16 bit.



Hình 1.6 So sánh hiệu năng và mức độ tổn hao bộ nhớ của 3 tập lệnh ARM, Thumb-2 và Thumb

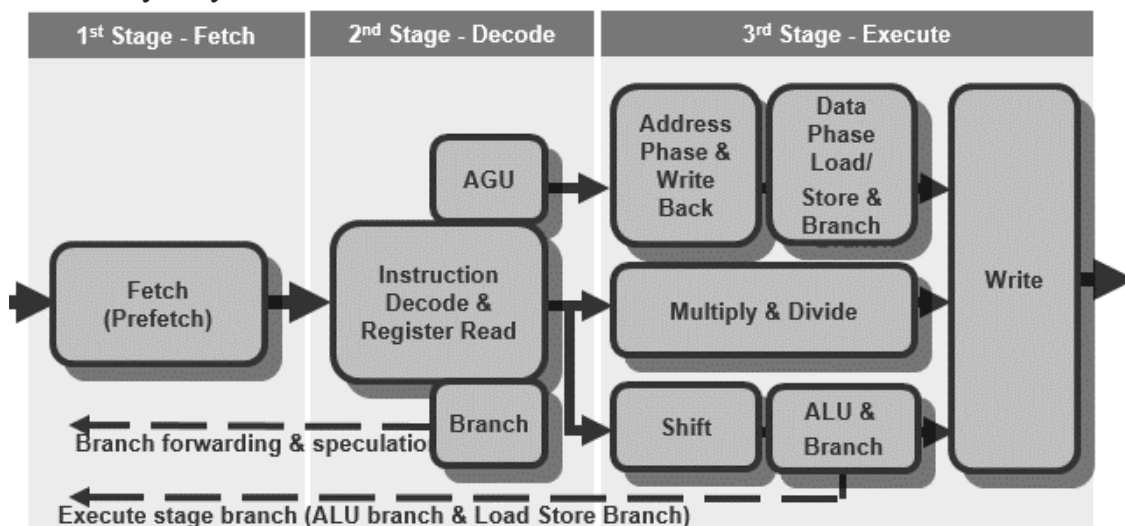
1.8 CẤU TRÚC ĐƯỜNG ỐNG 3 TẦNG (3-STAGE PIPELINE)

Lõi Cortex có cấu trúc đường ống gồm 3 tầng: **Instruction Fetch**(nhận lệnh), **Instruction Decode** (giải mã) và **Instruction Execute**(thực thi).

Cấu trúc đường ống 3 tầng của ARM Cortex-M3 tương tự như của ARM7 nhưng ở M3 cấu trúc này làm việc hiệu quả hơn giúp tăng tốc độ xử lý và tối ưu cho hoạt động ở tần số thấp.

Khối nhận lệnh có thể chứa đến 3 lệnh 32 bit hoặc 6 lệnh 16 bit hoặc là kết hợp cả 2.

Trong giai đoạn giải mã khối nhận lệnh sẽ đón các lệnh có khả năng sẽ được thực thi kế tiếp và khi đến giai đoạn thực thi thì lệnh tiếp theo đã có sẵn không làm tốn thời gian chờ điều này làm cho M3 có khả năng thực hiện 1 lệnh trong 1 chu kỳ máy. Trong ARM7 công việc này thường tốn 3 chu kỳ máy.



Hình 1.7 Cấu trúc đường ống 3 tầng của ARM Cortex-M3