2. Kiến trúc và tính năng vi xử lý Cortex-M3

Bộ vi xử lý Cortex-M3 dựa trên kiến trúc ARMv7-M có cấu trúc thứ bậc. Nó tích hợp lõi xử lý trung tâm, gọi là CM3Core, với các thiết bị ngoại vi hệ thống tiên tiến để tạo ra các khả năng như kiểm soát ngắt, bảo vệ bộ nhớ, gỡ lỗi và theo vết hệ thống. Các thiết bị ngoại vi có thể được cấu hình một cách thích hợp, cho phép bộ vi xử lý Cortex-M3 đáp ứng được rất nhiều ứng dụng và yêu cầu khắt khe của hệ thống. Lõi của bộ vi xử lý Cortex-M3 và các thành phần tích hợp (hình 3) đã được thiết kế đặc biệt để đáp ứng yêu cầu bộ nhớ tối thiểu, năng lượng tiêu thụ thấp và thiết kế nhỏ gọn.

2.1 Lõi Cortex-M3

Lõi trung tâm Cortex-M3 dựa trên kiến trúc Harvard, được đặc trưng bằng sự tách biệt giữa vùng nhớ chứa dữ liệu và chương trình do đó có các bus riêng để truy cập (hình 3). Đặc tính này khác với dòng ARM7 dựa trên kiến trúc Von Neumann sử dụng chung vùng nhớ để chứa dữ liệu và chương trình, do đó dùng chung bus cho việc truy xuất. Vì có thể đọc cùng lúc lệnh và dữ liệu từ bộ nhớ, bộ vi xử lý Cortex-M3 có thể thực hiện nhiều hoạt động song song, tăng tốc thực thi ứng dụng.

Lõi Cortex có cấu trúc đường ống gồm 3 tầng: **Instruction Fetch**, **Instruction Decode** và **Instruction Execute**. Khi gặp một lệnh nhánh, tầng decode chứa một chỉ thị nạp lệnh suy đoán có thể dẫn đến việc thực thi nhanh hơn. Bộ xử lý nạp lệnh dự định rẽ nhánh trong giai đoạn giải mã. Sau đó, trong giai đoạn thực thi, việc rẽ nhánh được giải quyết và bộ vi xử lý sẽ phân tích xem đâu là lệnh thực thi kế tiếp. Nếu việc rẽ nhánh không được chọn thì lệnh tiếp theo đã sẵn sàng. Còn nếu việc rẽ nhánh được chọn thì lệnh rẽ nhánh đó cũng đã sẵn sàng ngay lập tức, hạn chế thời gian rỗi chỉ còn một chu kỳ.

Lõi Cortex-M3 chứa một bộ giải mã cho tập lệnh Thumb truyền thống và Thumb-2 mới, một ALU tiên tiến hỗ trợ nhân chia phần cứng, điều khiển logic, và các giao tiếp với các thành phần khác của bộ xử lý.

Bộ vi xử lý Cortex-M3 là một bộ vi xử lý 32-bit, với độ rộng của đường dẫn dữ liệu 32 bit, các dải thanh ghi và giao tiếp bộ nhớ. Có 13 thanh ghi đa dụng, hai con trỏ ngăn xếp, một thanh ghi liên kết, một bộ đếm chương trình và một số thanh ghi đặc biệt trong đó có một thanh ghi trạng thái chương trình.

Bộ vi xử lý Cortex-M3 hỗ trợ hai chế độ hoạt động (Thread và Handler) và hai mức truy cập tài nguyên của lõi xử lí (đặc quyền và không đặc quyền), tạo điều kiện cho việc cài đặt các hệ thống mở và phức tạp nhưng vẫn bảo mật. Những dòng mã không đặc quyền bị giới hạn hoặc không cho phép truy cập vào một số tài nguyên quan trọng (một số lệnh đặc biệt và các vùng nhớ nhất định). Chế độ Thread là chế độ hoạt động tiêu biểu hỗ trợ cả mã đặc quyền và không đặc quyền. Bộ vi xử lý sẽ vào chế độ Handler khi một ngoại lệ (exception) xảy ra và tất cả các mã là đặc quyền trong chế độ này. Ngoài ra, tất cả các hoạt động trong bộ vi xử lý đều thuộc một trong hai trạng thái hoạt động: Thumb cho chế độ thực thi bình thường và Debug cho việc gỡ lỗi.

Bộ vi xử lý Cortex-M3 là một hệ thống ánh xạ bộ nhớ đơn giản, quản lí vùng nhớ cố định lên tới 4 gigabyte với các địa chỉ định nghĩa sẵn, dành riêng cho mã lệnh (vùng mã lệnh), SRAM (vùng nhớ), bộ nhớ/thiết bị bên ngoài, thiết bị ngoại vi bên trong và bên ngoài. Ngoài ra còn có một vùng nhớ đặc biệt dành riêng cho nhà cung cấp.

Bộ vi xử lý Cortex-M3 cho phép truy cập trực tiếp đến từng bit dữ liệu trong các hệ thống đơn giản bằng cách thực thi một kỹ thuật được gọi là bit-banding (hình 5). Bộ nhớ bao gồm hai vùng bit-band (mỗi vùng 1MB) trong SRAM và vùng bí danh 32MB của vùng không gian ngoại vi (Mỗi byte trong vùng bí danh sẽ tương ứng với một bit trong vùng bit-band). Mỗi hoạt động nạp/lưu tại một địa chỉ trong khu vực bí danh (alias region) sẽ trực tiếp tương ứng với hoạt động trên bit được đại diện bởi bí danh đó. Cụ thể, khi ghi giá trị 0x01 vào một địa chỉ trên vùng bí danh thì có nghĩa là xác định bit tương ứng sẽ có giá trị là 1, tương tự giá trị 0x00 sẽ xác định bit tương ứng có giá trị 0. Còn đọc giá trị tại một địa chỉ vùng bí danh có nghĩa là đọc được giá trị của bit tương ứng. Một vấn đề cần chú ý nữa là hoạt động này mang tính nguyên tử (không chia nhỏ được nữa), không thể bị gián đoạn bởi các hoạt động khác trên bus.

Các hệ thống cũ dựa trên ARM7 chỉ hỗ trợ truy xuất dữ liệu thẳng hàng, chỉ cho phép lưu trữ và truy xuất dữ liệu của một khối bộ nhớ mà mỗi phần tử có đơn vị là một word. Bộ vi xử lý Cortex-M3 hỗ trợ truy xuất dữ liệu không thẳng hàng, cho phép chuyển dữ liệu không thẳng hàng trong một truy xuất đơn. Thực tế, việc chuyển dữ liệu không thẳng hàng được biến thành việc chuyển nhiều lần dữ liệu thẳng hàng và có tính trong suốt đối với lập trình viên (nghĩa là lập trình viên hoàn toàn không cần quan tâm đến điều này). Ngoài ra bộ vi xử lý Cortex-M3 cũng hỗ trợ phép nhân 32-bit hoạt động trong một chu trình đơn và các phép chia có dấu, không dấu với các lệnh SDIV và UDIV, mất từ 2 đến 12 chu kỳ phụ thuộc vào kích thước của toán hạng. Phép chia được thực thi nhanh hơn nếu số chia và số bị chia có kích thước tương tự nhau. Những cải tiến trong khả năng toán học giúp Cortex-M3 trở thành bộ vi xử lý lý tưởng cho các ứng dụng thiên về tính toán như đọc cảm biến hoặc các hệ thống mô phỏng.

2.2 Kiến trúc tập lệnh Thumb-2

ARMv7-M 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 phù hợp với mật độ mã cũng như tương thích ngược với tập lệnh gốc Thumb 16 bit.

Trong một hệ thống dựa trên bộ vi xử lý ARM7, việc chuyển đổi nhân xử lý giữa chế độ Thumb (có lợi về mật độ mã) và ARM (có lợi về mặt hiệu suất) là cần thiết cho một số ứng dụng. Còn bộ vi xử lý Cortex-M3 có các lệnh 16 bit và 32 bit tồn tại trong cùng một chế độ, cho phép mật độ mã cũng như hiệu suất đều cao hơn mà không cần phải chuyển đổi phức tạp. Vì tập lệnh Thumb-2 là tập bao hàm của tập lệnh Thumb 16 bit nên bộ vi xử lý Cortex-M3 có thể thực thi các đoạn mã trước đây viết cho Thumb 16 bit. Do được cài đặt tập lệnh Thumb-2 nên bộ vi xử lý Cortex-M3 có khả năng tương thích với các thành viên khác của dòng ARM Cortex.

Tập lệnh Thumb-2 có các lệnh đặc biệt giúp lập trình viên dễ dàng viết mã cho nhiều ứng dụng khác nhau. Các lệnh BFI và BFC là các lệnh thao tác trên bit, rất có ích trong các ứng dụng xử lý gói tin mạng. Các lệnh SBFX và UBFX giúp việc chèn vào hoặc trích xuất một số bit trong thanh ghi được nhanh chóng. Lệnh RBIT đảo bit trong một WORD, có ích trong các thuật toán DSP như DFT. Các lệnh bảng rẽ nhánh TBB và TBH tạo sự cân bằng giữa mật độ mã và hiệu suất. Tập lệnh Thumb-2 cũng giới thiệu cấu trúc If-Then mới có thể xác định điều kiện thực hiện tối đa bốn lệnh tiếp theo.

2.3 Bộ điều khiển vector ngắt lồng nhau (NVIC)

NVIC (Nested Vectored Interrupt Controller) là thành phần tích hợp của bộ vi xử lý Cortex-M3 có khả năng xử lý ngắt rất linh hoạt và nhanh chóng. Trong cài đặt chuẩn, nó cung cấp một NMI (Non-Maskable Interrupt) và 32 ngắt vật lý đa dụng với 8 mức ưu tiên pre-emption. Nó có thể được cấu hình từ 1 đến 240 ngắt vật lý với tối đa 256 mức độ ưu tiên.

Bộ vi xử lý Cortex-M3 sử dụng một bảng vector có thể tái định vị được, dùng để chứa địa chỉ của hàm xử lý ngắt. Khi nhận một ngắt, bộ xử lý sẽ lấy địa chỉ từ bảng vector thông qua bus chương trình. Bảng vector ngắt được đặt ở địa chỉ 0 khi reset, nhưng có thể được di chyển đến vị trí khác bằng cách lập trình một thanh ghi điều khiển.

Để giảm bớt số cổng và tăng tính linh hoạt hệ thống, bộ vi xử lý Cortex-M3 đã chuyển từ mô hình ngoại lệ thanh ghi theo dõi của bộ vi xử lý ARM7 sang mô hình ngoại lệ dựa trên stack. Khi có một ngoại lệ xuất hiện thì bộ đếm chương trình (Program Counter), thanh ghi trạng thái chương trình (Program Status Register), thanh ghi liên kết (Link Register) và các thanh ghi đa dụng từ R0-R3, R12 bị đẩy vào ngăn xếp. Trong khi bus dữ liệu đẩy các thanh ghi lên vùng ngăn xếp thì bus chương trình xác định các vector ngoại lệ từ bảng vector và nạp lệnh đầu tiên của mã chương trình xử lí ngoại lệ. Sau khi hoàn tất việc lưu trữ dữ liệu trên ngăn xếp và nạp lệnh, chương trình phục vụ ngắt và xử lý lỗi được thực thi, tiếp theo đó các thanh ghi sẽ được phục hồi tự động để chương trình bị ngắt tiếp tục thực hiện bình thường. Vì thực hiện các hoạt động ngăn xếp bằng phần cứng nên ta không cần viết các đoạn hợp ngữ để thực hiện các thao tác trên ngăn xếp cho các hàm xử lý ngắt truyền thống dựa trên ngôn ngữ C, giúp việc phát triển ứng dụng dễ dàng hơn rất nhiều.

NVIC hỗ trợ ngắt lồng nhau, cho phép một ngắt được xử lý trước một ngắt khác dựa trên mức độ ưu tiên. Nó cũng hỗ trợ cấu hình mức ưu tiên động cho các ngắt. Độ ưu tiên có thể được thay đổi bằng phần mềm trong thời gian chạy (run time). Các ngắt đang được xử lý đều bị khóa cho đến khi hàm xử lý ngắt hoàn thành, do đó, độ ưu tiên của ngắt có thể thay đổi mà không cần lo đến chuyện trùng lặp.

Trong trường hợp các ngắt nối đuôi nhau, các hệ thống cũ sẽ lặp lại hai lần việc lưu trạng thái hoàn thành và khôi phục, dẫn đến độ trễ cao. Bộ vi xử lý Cortex-M3 đơn giản hóa việc chuyển đổi giữa các ngắt đang hoạt động và đang chờ bằng cách cài đặt công nghệ tail-chaining trong phần cứng NVIC. Tail-chaining đạt độ trễ thấp hơn nhiều bằng cách thay thế chuỗi các thao tác pop và push vốn mất hơn 30 chu kỳ xung nhịp bằng một thao tác nạp lệnh đơn giản chỉ mất 6 chu kỳ. Trạng thái bộ vi xử lý được tự động lưu khi ngắt bắt đầu được xử lý và phục hồi ngay khi kết thúc, ít chu kỳ hơn so với việc thực thi bằng phần mềm, nâng cao hiệu suất đáng kể ở hệ thống hoạt động dưới 100MHz.

NVIC cũng cài đặt cách thức quản lý năng lượng của bộ vi xử lý Cortex-M3 tích hợp chế độ ngủ. Chế độ Sleep Now được gọi bằng một trong hai lệnh WFI (Wait For Interrupt) hoặc WFE (Wait For Event) sẽ ngay lập tức đặt nhân bộ vi xử lý vào trạng thái năng lượng thấp và chờ một ngoại lệ (exception). Chế độ Sleep On Exit đặt hệ thống vào chế độ năng lượng thấp ngay khi nó thoát khỏi hàm xử lý ngắt có độ ưu tiên thấp nhất. Nhân bộ vi xử lý vẫn ở trạng thái ngủ cho đến khi gặp một ngoại lệ. Vì chỉ có thể thoát khỏi chế độ này bằng ngắt nên trạng thái hệ thống không được phục hồi. Bit SLEEPDEEP của thanh ghi điều khiển hệ thống nếu được thiết lập có thể được sử dụng để khoá cổng (clock gate) lõi bộ vi xử lý và các thành phần hệ thống khác để tiết kiệm điện năng.

NVIC cũng tích hợp một bộ đếm SysTick 24-bit đếm ngược (count-down timer) có thể được sử dụng để định thời tạo ra ngắt, cung cấp nhịp đập để một hệ điều hành thời gian thực hoạt động hoặc các tác vụ được lập lịch.

2.4 Đơn vị bảo vệ bộ nhớ (MPU)

MPU là một thành phần tùy chọn của bộ vi xử lý Cortex-M3, có thể nâng cao độ tin cậy của hệ thống nhúng bằng cách bảo vệ các dữ liệu quan trọng được hệ điều hành sử dụng khỏi các ứng dụng khác, tách biệt độc lập các tác vụ đang thực thi bằng cách không cho phép truy cập vào dữ liệu của nhau, vô hiệu hoá quyền truy cập vào một số vùng nhớ, cho phép các vùng nhớ được định nghĩa là chỉ đọc (read only) và phát hiện các truy cập bộ nhớ có thể phá vỡ hệ thống.

MPU cho phép một ứng dụng được chia nhỏ thành các tiến trình. Mỗi tiến trình sẽ có bộ nhớ (code, dữ liệu, ngăn xếp, heap) và thiết bị riêng, cũng như có quyền truy cập vào bộ nhớ và các thiết bị được chia sẻ. MPU cũng có các quy tắc (rule) truy cập của người dùng và đặc quyền bao gồm việc thực thi mã tại mức đặc quyền thích hợp cũng như quyền sở hữu bộ nhớ và các thiết bị của mã đặc quyền và mã người dùng.

MPU chia bộ nhớ thành các vùng riêng biệt và thực hiện việc bảo vệ bằng cách ngăn các truy cập trái phép. MPU có thể chia bộ nhớ thành tối đa 8 vùng trong đó mỗi vùng có thể được chia thành 8 vùng con. Kích thước vùng có thể bắt đầu từ 32 byte và tăng gấp đôi dần cho đến tối đa 4 gigabyte. Các vùng được đánh số thứ tự bắt đầu từ 0. Có thể xác định một bản đồ bộ nhớ (memory map) nền mặc định để truy cập đặc quyền. Việc truy cập đến các địa chỉ bộ nhớ không được xác định trong vùng MPU hoặc không được phép sẽ tạo ra ngoại lệ lỗi về quản lí bộ nhớ (Memory Management Fault Exception).

Quy tắc bảo vệ vùng nhớ được dựa trên vào loại tác vụ (đọc, viết hoặc thực thi) và đặc quyền của mã thực hiện việc truy cập. Mỗi vùng bao gồm một bộ bit quy định loại truy cập được phép và hành động nào được phép trên bus. MPU cũng hỗ trợ các vùng chồng lên nhau (overlapping regions), tức là có sự giao nhau cùng một vùng địa chỉ. Vì kích thước mỗi vùng là bội số của 2 nên nếu 2 vùng chồng lên nhau thì sẽ có thể có một vùng nằm hoàn toàn trong vùng kia. Do đó, hoàn toàn có khả năng xảy ra trường hợp nhiều vùng nằm trọn trong một vùng hoặc trường hợp chồng lồng nhau. Trong trường hợp địa chỉ tra cứu nằm trong vùng chồng nhau thì kết quả trả về sẽ là vùng có số thứ tự cao nhất.

2.5 Gỡ lỗi (Debug) và theo vết (Trace)

Việc gỡ lỗi hệ thống dựa trên bộ vi xử lý Cortex-M3 được thực hiện thông qua DAP (Debug Access Port), có thể là một cổng SWD (Serial Wire Debug) sử dụng 2 đường tín hiệu hoặc một cổng SWJ-D (Serial Wire JTAG Debug) sử dụng giao thức JTAG hoặc SW. Các SWJ-DP mặc định để chế độ JTAG khi reset và có thể chuyển giao thức với một chuỗi điều khiển cụ thể được cung cấp bởi phần cứng gỡ lỗi bên ngoài.

Hành động debug có thể được kích hoạt bởi các sự kiện khác nhau như breakpoints, watchpoints, điều kiện lỗi hoặc yêu cầu debug từ bên ngoài. Khi một sự kiện debug xảy ra, bộ vi xử lý Cortex-M3 có thể vào chế độ tạm dừng (halt mode) hoặc chế độ theo dõi debug. Trong chế độ tạm dừng, bộ vi xử lí ngưng thực thi hoàn toàn các chương trình. Chế độ này hỗ trợ chạy từng bước. Lúc này, một ngắt phát sinh có thể bị trì hoãn đáp ứng, có thể được thực thi từng bước, hoặc bị che (masked) nên ngắt bên ngoài có thể bị bỏ qua trong quá trình debug. Trong chế độ theo dõi debug, một hàm xử lý ngoại lệ được thực thi để thực hiện việc gỡ lỗi trong khi vẫn cho phép các exception có độ ưu tiên cao hơn diễn ra. Chế độ này cũng hỗ trợ chạy từng bước.

Bộ FPB (Patch Flash and Breakpoint) có 6 breakpoint trong chương trình và 2 breakpoint nạp dữ liệu, hoặc chuyển lệnh/dữ liệu từ bộ nhớ mã đến bộ nhớ hệ thống. Bộ FPB này có sáu

comparator để so sánh các lệnh được lấy từ bộ nhớ mã. Mỗi comparator có thể được kích hoạt để định vị lại mã chương trình đến một vùng trong bộ nhớ hệ thống, hoặc thực hiện một breakpoint phần cứng bằng cách trả về một lệnh breakpoint cho bộ vi xử lý. Nó cũng có hai comparator với nhiệm vụ tương tự cho dữ liệu.

Bộ DWT (Data Watchpoint and Trace) có bốn comparator có thể được cấu hình thành watchpoint phần cứng. Khi được sử dụng trong cấu hình này, comparator có thể được lập trình để so sánh địa chỉ truy cập dữ liệu hoặc bộ đếm chương trình. Các DWT comparator cũng có thể được cấu hình để kích hoạt các sự kiện lấy mẫu PC, sự kiện lấy mẫu địa chỉ dữ liệu và làm cho ETM (Embedded Trace Macrocell) phát ra các gói kích hoạt trong dòng lệnh đang được truy vết.

ETM là một thành phần tùy chọn để hỗ trợ việc theo vết lệnh để đảm bảo rằng có thể tái cấu trúc lại việc thực hiện chương trình mà chỉ ảnh hưởng một cách tối thiểu đến bộ nhớ. ETM cho phép truy vết theo thời gian thực về việc thực thi lệnh và truyền dữ liệu bằng cách nén thông tin truy vết từ nhân bộ xử lý để giảm thiểu yêu cầu băng thông.

Bộ vi xử lý Cortex-M3 thực hiện việc theo vết dữ liệu bằng DWT và ITM (Trace Instrumentation Macrocell). DWT cung cấp số liệu thống kê về việc thực hiện lệnh và có thể tạo ra sự kiện watchpoint để gọi debug hoặc ETM trên các sự kiện hệ thống. ITM là công cụ truy vết ứng dụng hỗ trợ cách gỡ lỗi kiểu "printf" cho hệ điều hành và theo vết các sự kiện ứng dụng. Nó chấp nhận các gói truy vết phần cứng từ DWT và phần mềm theo dõi sự kích thích từ lõi bộ vi xử lý và phát ra thông tin chẩn đoán hệ thống định kỳ. Bộ TPIU (Trace Port Interface Unit) chấp nhận các thông tin truy vết từ ETM và ITM, sau đó hòa trộn chúng, định dạng lại và phát ra thông qua SWV (Serial Wire Viewer) đến các bộ phân tích truy vết bên ngoài. SWV cho phép tạo ra profile cho các sự kiện hệ thống một cách đơn giản và hiệu quả bằng cách xuất dòng dữ liệu thông qua một pin duy nhất. Mã hóa Manchester và UART là các định dạng được hỗ trợ cho SWV.

2.6 Ma trận bus và các giao diện liên kết

Ma trận bus của bộ vi xử lý Cortex-M3 kết nối bộ xử lý và giao diện debug đến các bus bên ngoài, ICode, DCode và giao diện hệ thống dựa trên AMBA AHB-Lite 32 bit, và bus cho các ngoại vi (Private Peripheral Bus) dựa trên AMBA APB 32 bit. Ma trận bus cũng đảm nhiệm việc truy cập dữ liệu không thẳng hàng và các vùng bit-banding.

Giao diện ICode 32 bit lấy các lệnh từ vùng nhớ chương trình và chỉ có thể truy cập bởi CM3Core. Tất cả các lần nạp lệnh đều có độ rộng là một từ (WORD), với số lượng lệnh được lấy trên mỗi từ tùy thuộc vào loại mã thực hiện và vị trí của nó trong bộ nhớ. Giao diện DCode 32 bit truy cập dữ liệu từ vùng nhớ mã chương trình và có thể được truy cập bởi CM3Core và DAP. Giao diện hệ thống 32 bit lấy các lệnh và truy cập dữ liệu trong vùng bộ nhớ hệ thống và giống như bus DCode, có thể được truy cập bởi CM3Core và DAP. PPB cho phép truy cập vào các thành phần bên ngoài của hệ thống Cortex-M3.

1. Giới thiệu:

Giải pháp Soc (System-on-chip) dựa trên bộ vi xử lý nhúng ARM được ứng dụng vào rất nhiều thị trường khác nhau bao gồm các ứng dụng doanh nghiệp, các hệ thống ô tô, mạng gia đình và công nghệ mạng không dây... Dòng vi xử lý ARM Cortex dựa trên một kiến trúc chuẩn đủ để đáp ứng hầu hết các yêu cầu

về hiệu năng làm việc trong tất cả các lĩnh vực trên. Dòng ARM Cortex bao gồm ba cấu hình khác nhau của kiến trúc ARMv7: cấu hình A cho các ứng dụng tinh vi, yêu cầu cao chạy trên các hệ điều hành mở và phức tạp như Linux, Android…; cấu hình R dành cho các hệ thống thời gian thực và cấu hình M được tối ưu cho các ứng dụng vi điều khiển, cần tiết kiệm chi phí.

Bộ vi xử lý Cortex-M3 là bộ vi xử lý ARM đầu tiên dựa trên kiến trúc ARMv7-M và được thiết kế đặc biệt để đạt được hiệu suất cao trong các ứng dụng nhúng cần tiết kiệm năng lượng và chi phí, chẳng hạn như các vi điều khiển, hệ thống cơ ô tô, hệ thống kiểm soát công nghiệp và hệ thống mạng không dây. Thêm vào đó là việc lập trình được đơn giản hóa đáng kể giúp kiến trúc ARM trở thành một lựa chọn tốt cho ngay cả những ứng dụng đơn giản nhất.

2. Kiến trúc và tính năng xử lý của lõi Cortex M3

Bộ vi xử lý Cortex-M3 dựa trên kiến trúc ARMv7-M có cấu trúc thứ bậc. Nó tích hợp lõi xử lý trung tâm, với các thiết bị ngoại vi hệ thống tiên tiến để tạo ra các khả năng như kiểm soát ngắt, bảo vệ bộ nhớ, gỡ lỗi và theo vết hệ thống.

Các thiết bị ngoại vi có thể được cấu hình một cách thích hợp, cho phép bộ vi xử lý Cortex-M3 đáp ứng được rất nhiều ứng dụng và yêu cầu khắt khe của hệ thống. Lõi của bộ vi xử lý Cortex-M3 và các thành phần tích hợp đã được thiết kế đặc biệt để đáp ứng yêu cầu bộ nhớ tối thiểu, năng lượng tiêu thụ thấp và thiết kế nhỏ gọn.

2.1 Kiến trúc lõi Cortex M3

Lõi Cortex M3 dựa trên cấu trúc Havard, được đặc trưng bằng sự tách biệt giữa vùng nhớ dữ liệu và chương trình. Vì có thể đọc cùng lúc lệnh và dữ liệu từ bộ nhớ, bộ vi xử lý Cortex-M3 có thể thực hiện nhiều hoạt động song song, tăng tốc thực thi ứng dụng.

Bộ vi xử lý Cortex M3

2.2 Kiến trúc 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 phù hợp với mật độ mã cũng như tương thích ngược với tập lệnh gốc Thumb 16 bit.

Quan hệ giữa tập lệnh Thumb-2 và tập lệnh Thumb

Trong một hệ thống dựa trên bộ vi xử lý ARM7, việc chuyển đổi nhân xử lý giữa chế độ Thumb (có lợi về mật độ mã) và ARM (có lợi về mặt hiệu suất) là cần thiết cho một số ứng dụng. Còn bộ vi xử lý Cortex-M3 có các lệnh 16 bit và 32 bit tồn tại trong cùng một chế độ, cho phép mật độ mã cũng như hiệu suất đều cao hơn mà không cần phải chuyển đổi phức tạp. Vì tập lệnh Thumb-2 là tập bao hàm của tập lệnh Thumb 16 bit nên bộ vi xử lý Cortex-M3 có thể thực thi các đoạn mã trước đây viết cho Thumb 16 bit. Do được cài đặt tập lệnh Thumb-2 nên bộ vi xử lý Cortex-M3 có khả năng tương thích với các thành viên khác của dòng ARM Cortex.

Tập lệnh Thumb-2 có các lệnh đặc biệt giúp lập trình viên dễ dàng viết mã cho nhiều ứng dụng khác nhau. Các lệnh BFI và BFC là các lệnh thao tác trên bit, rất có ích trong các ứng dụng xử lý gói tin mạng. Các lệnh SBFX và UBFX giúp việc chèn vào hoặc trích xuất một số bit trong thanh ghi được nhanh chóng. Lệnh RBIT đảo bit trong một WORD, có ích trong các thuật toán DSP như DFT. Các lệnh bảng rẽ nhánh TBB và TBH tạo sự cân bằng giữa mật độ mã và hiệu suất. Tập lệnh Thumb-2 cũng giới thiệu cấu trúc If-Then mới có thể xác định điều kiện thực hiện tối đa bốn lệnh tiếp theo.

Các tính năng chính mới trong tập lệnh Thumb-2 bao gồm việc thực hiện mã lệnh C một cách tự nhiên hơn, thao tác trực tiếp trên các bit, phép chia phần cứng và lệnh If/Then. Hơn nữa, nhìn từ góc độ phát triển ứng dụng, Thumb-2 tăng tốc độ phát triển, đơn giản hóa việc bảo trì, hỗ trợ các đối tượng biên dịch thông qua tối ưu hóa tự động cho cả hiệu suất và mật độ mã mà không cần quan tâm đến việc mã được biên dịch cho chế độ ARM hoặc Thumb. Kết quả là lập trình viên có thể để mã nguồn của họ trong ngôn ngữ C mà không cần tạo ra các thư viện đối tượng biên dịch sẵn, có nghĩa là khả năng tái sử dụng mã nguồn lớn hơn nhiều.

2.3 Cách tổ chức và thực thi tập lệnh

Cách tổ chức của nhân ARM là dòng chảy lệnh 3 tác vụ:
- Fetch (nhận lệnh).
- Decode (giải mã).
- Excute (thực thi).

Hình 1: Câu lệnh một chu kỳ máy sử dụng dòng chảy lệnh có 3 tác vụ.

Khi gặp một lệnh nhánh, tầng decode chứa một chỉ thị nạp lệnh suy đoán có thể dẫn đến việc thực thi nhanh hơn. Bộ xử lý nạp lệnh dự định rẽ nhánh trong giai đoạn giải mã. Sau đó, trong giai đoạn thực thi, việc rẽ nhánh được giải quyết và bộ vi xử lý sẽ phân tích xem đâu là lệnh thực thi kế tiếp. Nếu việc rẽ nhánh không được chọn thì lệnh tiếp theo đã sẵn sàng. Còn nếu việc rẽ nhánh được chọn thì lệnh rẽ nhánh đó cũng đã sẵn sàng ngay lập tức, hạn chế thời gian rỗi chỉ còn một chu kỳ.

2.4 Bộ nhớ:

Bộ vi xử lý Cortex-M3 quản lí vùng nhớ cố định lên tới 4 gigabyte với các địa chỉ định nghĩa sẵn, dành riêng cho mã lệnh (vùng mã lệnh), SRAM (vùng nhớ), bộ nhớ/thiết bị bên ngoài, thiết bị ngoại vi bên trong và bên ngoài. Ngoài ra còn có một vùng nhớ đặc biệt dành riêng cho nhà cung cấp.Code có thể lưu ở vùng mã lệnh, SRAM hoặc RAM ngoài, tuy nhiên khi lưu ở vùng mã lệnh, việc gọi lệnh và truy cập dữ liệu được xử lý đồng thời trên các bus.

Sơ đồ bộ nhớ

- Truy cập vùng SRAM thực hiện thông qua các bus. Trong vùng này, có hai thành phần: vùng bit-band 1MB và vùng bí danh (Bit-band Alias) 32 MB.

- Vùng nhớ thiêt bị ngoại vi 0.5GB tương tự như vùng SRAM với hai thành phần cơ bản. Tuy nhiên, các lệnh không được thực thi ở vùng này, kĩ thuật bit-band được sử dụng để thay đổi các trạng thái bit, dễ dàng điều khiển ngoại vi.
- Vùng bộ nhớ và thiết bị bên ngoài: mỗi vùng 1GB, sự khác biệt hai vùng này là lệnh không được thực hiện ở vùng thiết bị bên ngoài, ngoài ra có một số khác biệt về cách lưu trữ giữa hai vùng.