(120-125)일 반 -ARM/전 자 -6 2008.5.27 4:3 PM페 이 지 120



# ARM® 사의 Cortex™-M3의 특성 및 최신 동향

**김형태**, 펌웨어뱅크(주)

i8051은 전 세계 반도체 회사들이 회로도에 해당하는 코어를 인텔로부터 사들인 후, 자사가 구현하고자 하는 특정 소자를 붙임으로써 8비트의 대표 마이크로프로세서로 만들어졌다. 이로 인해 MPU(Microprocessor Unit)를 변경하더라도 8051 코어로만 되어 있다면 펌웨어 이식의 호환성이 뛰어났고 IC를 만드는 회사에서는 특별한 Peripheral을 붙여 특성화하기도 편리했는데, 이제는 ARM사에서 설계한 32비트 마이크로프로세서 ARM Core가 비슷한 방법으로 급속도로 확산되고 있다.

ARM 코어를 이용하여 만들어지는 프로세서는 Family로 나눌 수 있는데 ARM7, ARM9, ARM11에 최근 Cortex Family가 추가되었다. 그림 1에서 x축은 시간의 흐름을 나타내고 y축 위로의 진행 방향은 고속/고성능의 분포를 나타낸다. ARM7, ARM9 코어는 우리나라에서도 Silicon IP 형태로 구입한 회사가 상당수 있지만, 몇몇 기업의 ARM9를 제외하고는 대중적인 IC 형태의 ARM 코어가 탑재된 국산 제품을 구입하여 애플리케이션으로 구현하기는 쉽지 않다. 주로 화상, 음성, 멀티미디어 등의 신호처리와 임베디드 분야의 특정 용도로 사용되고 있는 추세이다.

ARM 코어는 11 이후 계속적인 번호 증가로 성장할 것이라 예상됐으나 새로운 코어를 가지고 변신했다. Cortex라는 머리글자인데, Cortex-M, Cortex-R, Cortex-A 시리즈로 나눌 수 있다. 내부 엔진의 구성과 명령어 처리 방법, 디버깅 등의 기능을 강화하여 더욱 세련된 코어의 이름을 갖게 된 듯하다.

Cortex-M, Cortex-R, Cortex-A 시리즈는 모두 Thumb-2®의 Instruction 방식을 사용한다. Thumb-2는 16비트 구성의 Thumb 명령어와 32비트 구성의 ARM 명령어를 혼용하여 사용하는 방식으로, 이 둘을 혼용하여 사용함으로써 코드 효율성을 높였다. 이러한 방법은 우리나라에서도 이미 국제 특허를 갖고 있는 에이디칩스(www.adc.co.kr)의 EISC(Extendable Instruction Set Computing)가 있다.

Cortex-M 시리즈 중에서 처음 나온 것은 Cortex-M3이며

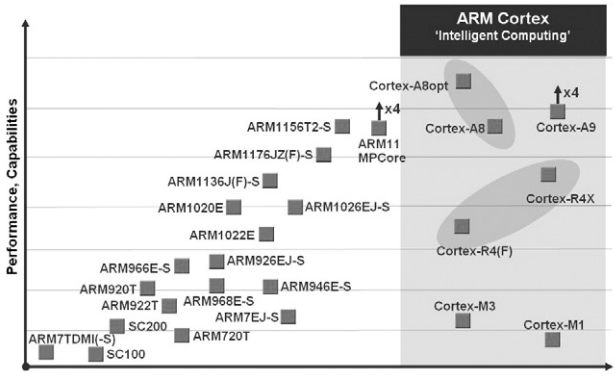


그림 1. Public ARM Processor Road map

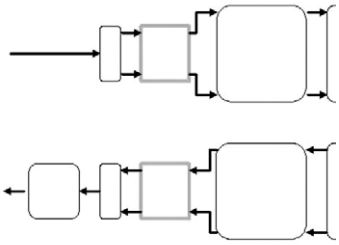

그림 2. ARM Cortex Family

(120-125)일 반 -ARM/전 자 -6 2008.5.27 4:3 PM페 이 지 121



마이크로프로세서 코어를 이용하여 제어 등에 활용하기 적합하고 Thumb-2 전용 명령어를 처리한다. 가격에 민감한 제품을 만들거나 코드집적도가 높고 다수의 인터럽트 처리가 필요한 응용 개발에 알맞다고 할 수 있다.

Cortex-R 시리즈 중에서 처음 나온 것은 Cortex-R4이며 Real-Time 환경에 적용하기 위해 만들어진 임베디드 프로세서라고 할 수 있고 Cortex-M과 마찬가지로 Thumb-2 명령어를 처리한다.

Cortex-A 시리즈 중에서 처음 나온 것은 Cortex-A8이며 복잡한 OS와 User Application을  처리하기 위한 CPU라고 할 수 있고 다른 Cortex 시리즈와 마찬가지로 Thumb-2 명령어를 처리한다.

여기서는 마이크로프로세서 응용 분야에서 제어용으로 활용하는데 편리하고 비교적 다루기 쉬운 Cortex-M3의 코어 특징, 구조, 성능, 응용 분야 및 개발 환경에 대해 알아본다. 현재 급속도로 다양한 제품군과 응용 범위가 넓은 제품이 계속 출시되고 있는 시점에서, 검토 대상과 흐름을 파악하고자 하는 취지로 조사하게 되었다.

## ARM®의 Cortex™-M3

Cortex-M3를 사용했을 때의 메리트는 무엇일까? 마이크로프로세서를 이용하여 제작되는 가전제품과 산업 현장의 애플리케이션은 32비트 추세로 현재 시장이 점점 오픈되고 있다. 또한 빠르고 소비전력이 적으면서 많은 기능을 내포하기를 원한다. 이와 같은 사항들을 수용하기 위해서는 32비트 급으로 가야 하지만, 구현되어야 하는 알고리즘의 비트 효율을 높이기 위해서는 32비트 자체만으로 개발하기엔 무리가 있다.

예를 들어 AD(Analog Digital) 변환을 32비트로 하기도 어렵고 어드레스의 범위도 $2^{32}$의 4G 영역을 전부 사용하기 위한 알고리즘을 개발하기도 힘겨울 것이다. 무엇보다 명령어 처리를 32비트로만 한다면 FLASH와 RAM 영역의 비트 효율이 더 떨어지는 것은 당연하다. 그래서 대부분의 응용 임베디드 개발은 16비트 처리 방법으로 효율을 높이고 32비트 처리로 속도를 높이는 추세가 되는데, 이 때 사용되는 것이 Thumb-2 명령어 처리이다. 여기에 칩의 집적도가 높아 동일한 웨이퍼 위에 다수의 마

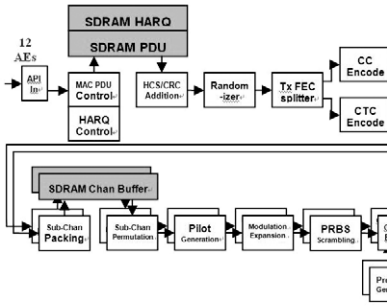


그림 4. Thumb-2 명령어

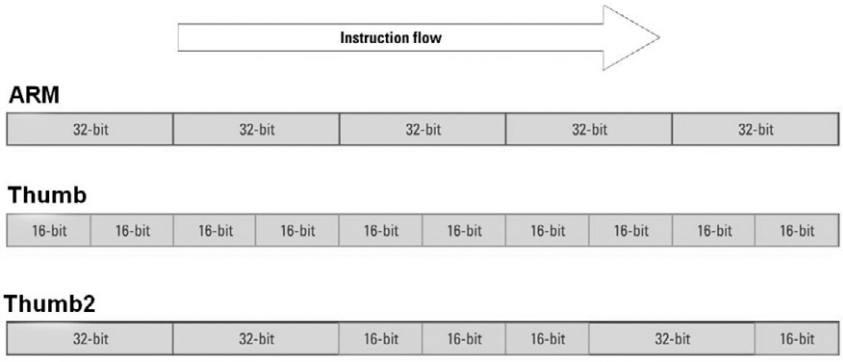


그림 3. Thumb-2 명령의 처리 개념

(120-125)일 반 -ARM/전 자 -6 2008.5.27 4:3 PM페 이 지 122



이크로프로세서를 양산할 수 있다면 더욱 저렴해지고 코어의 처리속도가 개선되므로 소비전력이 작아져 더욱 매력 있게 되는데, 그 중 대표적인 제품이 ARM(www.ARM.com)의 Cortex-M3이다.

## 1. 특징

Cortex-M3의 Thumb-2 core 기술은 ARM사의 ARMv7의 특징과 Harvard Architecture(Code Bus, Data Bus, System Bus가 각각 따로 존재함)를 갖고 있으며 명령어는 16비트 명령어와 32비트 명령어를 혼용으로 사용할 수 있어 메모리와 명령어 처리의 효율을 높이는 특징이 있다.

그림 3에 나타난 바와 같이, ARM 명령어와 Thumb을 혼용한 형태를 갖고 있다. 이것의 의미는 32비트 RISC에서 Cortex-M3의 모든 레지스터는 32비트로 되어 있기 때문에 16비트 명령은 32비트의 긴 필드가 필요 없는 간단한 처리를 16비트로 처리하여 메모리 효율을 높이기 위한다는 것이고, 여러 개의 레지스터를 처리하거나 긴 Immediate Value를 쓰는 경우에는 32비트 명령으로 처리하여 속도를 높이는 방법이라 할 수 있다. 이것을 Thumb-2라고 한다.

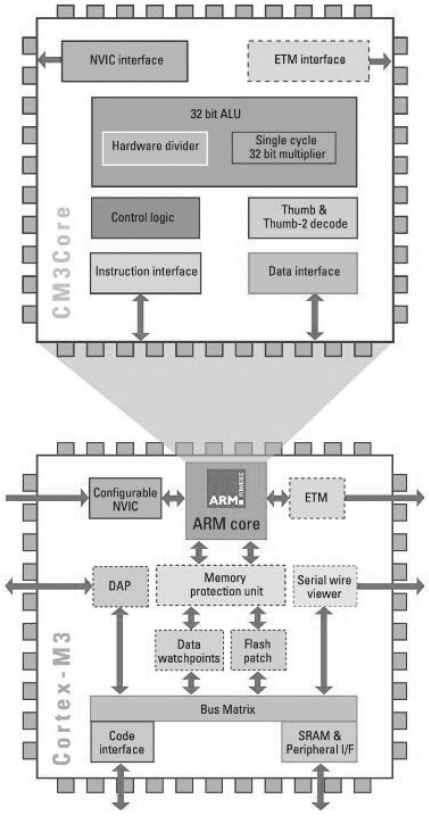


그림 5. Cortex-M3의 내부 구조

## 2. 구조

내부적으로 3-stage 파이프라인(pipeline) 처리와 C 언어로의 개발을 염두에 두고 설계된 독특한 구조를 갖고 있다. 3-state 파이프라인은 Fetch, Decode, Execute로 나눌 수 있다. Fetch란 실행할 코드를 가져오는 것을 의미하고, Decode란 가져온 명령어를 CPU가 알 수 있는 코드로 바꾸는 것을 의미하며, Execute는 당연히 이것을 실행하는 것을 말한다. 각 단계는 서로에게 영향을 미치지만 동작은 독립되어 있다. 예를 들면, Execute를 하면서 Decode도 동시에 할 수 있고, Decode를 하면서 동시에 Fetch를 할 수 있다는 뜻이다.

C 언어로의 개발을 염두에 둔 설계는 그림 4에 나타난 바와 같이, C의 조건문 if( )와 { }의 if~then에 해당하는 처리를 위한 IT 처리기, Table branch 전용 TB{B | H} [Rbase, Rindex] 그리고 비트 field insert/extract/clear 처리를 실행하는 BFI, {S | U}BFX, BFC 비트 반전 RBIT 처리를 실행할 수 있는 명령어가 있다.

Sensor 핸들링의 축약 명령어 DIV의 SDIV, UDIV 그리고 FLASH 처리를 위한 16비트 즉시 처리의 MOVW, MOVT 명령어가 있어 C/C++의 고급 언어를 컴파일러를 통해 변경할 때 보다 효율적인 기계어 코드를 보장해 준다. Cortex-M3는 실제로 같은 프로그램을 ARM으로 컴파일 했을 때보다 약 40%의 코드 절감 효과가 있다.

그림 5를 통해 블록을 살펴보면, 32Bit ALU(Arithmetic Logic Unit)에는 나누기와 32비트 곱셈기가 내장되어 있고 NVIC(Integrated Nested Vector Interrupt Controller)가 있어 1~255의 Priority 레벨로 최소 1개에서 최대 244개까지의 인터럽트를 처리하도록, 설계단계에서 칩 메이커가 설정할 수 있다.

(120-125)일 반 -ARM/전 자 -6 2008.5.27 4:3 PM페 이 지 123



또 Bus Matrix는 Bit Banding 기법을 사용하여 효율적인 포인트 영역을 지정할 수 있는데, Bit-Banding은 32비트의 Read/Writ로 효율적으로 비트를 처리(Bit Manipulation)하기 위한 것이다. 자세히 설명하면, Bit-band-alias를 Bit-band-region으로 어드레스 변환해서 읽기/쓰기를 실행하는데, 이 과정에서 Bit-band-region의 특정 어드레스에서 32비트 중 특정 비트값만 읽거나 Bit-band-region의 특정 어드레스의 32비트 중 특정 비트값만 변환할(atomic read-modify-write) 수 있다. 또한 디버그를 위해 DAP를 설치, SWD/JTAG의 연결이 용이하므로 보다 적은 비용으로 개발 장비를 구현할 수 있도록 해준다.

Cortex-M3 Core를 칩 벤더에서 어떻게 사용하고 디자인하는가에 따라 다르겠지만, 비교적 비슷하게 **표 1**과 같은 메모리

표 1. Cortex-M3의 메모리 배치

| 항목 | 크기[Byte] | 설명 |
|---|---|---|
| Vendor Specific | 0.5G | Set aside to enable vendors to implement peripheral compatibility with previous systems |
| Private Peripheral Bus | 1M | Address space for system components(Core Sight, NVIC etc.) |
| External Device | 1G | Intended for external devices and/or shared memory that needs ordering/non-buffered |
| External RAM | 1G | Intended for off chip memory |
| Peripheral | 0.5G | Intended for normal peripherals. The bottom 1MB of the 32MB peripheral address space(0x40000000 0x400FFFFF) is reserved for bit-band accesses. Accesses to the peripheral 32MB bit band alias region(0x42000000 0x43FFFFFF) are remapped to this 1MB address space. |
| SRAM | 0.5G | Intended for on-chip SRAM. The bottom 1MB of the SRAM address space(0x20000000-0x200FFFFF) is reserved for bit-band accesses. Accesses to the SRAM 32MB bit band alias region(0x22000000 0x23FFFFFF) are remapped to this 1MB address space. |
| Code | 0.5G | Reserved for code memory(flash, SRAM). This region is accessed via the Cortex-M3 I-Code and D-Code busses. |

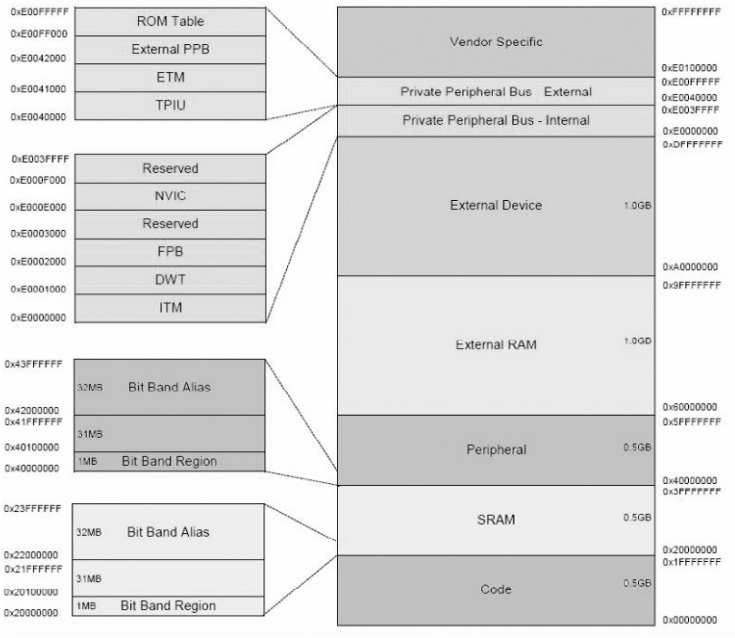


그림 6. Cortex-M3의 메모리 배치 블록

(120-125)일 반 -ARM/전 자 -6 2008.5.27 4:3 PM페 이 지 124



배치를 갖고 있다.

### 3. 성능

Cortex-M3에는 코어 외에 Bus Matrix와 NVIC등이 포함되어 있고 ARM7(TDMI)-S는 코어만 포함되어 있으므로 직접적인 비교는 어렵지만, 단지 Cortex-M3의 기본적인 최소 Gate Count가 39K~43K 게이트 정도 되므로 Core만 있는 ARM7(TDMI)-S보다 작은 사이즈라고 할 수 있다.

또 소비 전력은 Cortex-M3를 이상적인 조건(Core+Bus-matrix, +NVIC of 16 Interrupts, 0.13㎛ Metro™)에서 측정했을 때 0.107mW/MHz 정도인데 반해 ARM7TDMI-S(Core only, 0.13㎛ Metro™)는 0.152mW/MHz로 된다.

인터럽트 처리는 그림 7에 나타난 바와 같이, ARM7이 IRQ1에서 ISR1로 들어가고 42사이클의 Pop, Push, ISR2로 나오기까지 모두 84사이클이 필요한 반면, Cortex-M3는 개선된 인터럽트 처리 방법(Tail Chaining)으로 34개의 사이클 정도로 사이클 수가 줄고, 특히 Tail-chaining을 사용함으로써 ISR1에서 ISR2로 넘어가는데 필요한 사이클 수가 6사이클로 간소화된다.

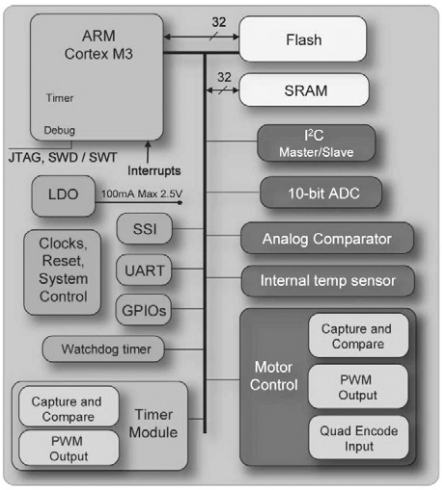


그림 8. LM 시리즈의 기본 구조

### 4. 응용

Cortex-M3 코어를 이용하여 마이크로프로서 IC 제품을 만드는 회사가 앞으로 더 많이 생기겠지만, 현 시점에서는 루미나리마이크로사(www.luminarymicro.com), ST사(www.st.com) 벤드의 제품이 출시되고 있다.

루미나리마이크로사의 제품은 AC, 스텝 모터 제어, 이더넷, CAN 통신, USB 등, 주로 제어기 개발 응용에 알맞다고 할 수 있다. LM3S시리즈의 주요 특징은 104개에 달하는 다양한 종류와 4가지 패키지(28 SOIC, 48 LQFP, 100 LQFP, BGA)가 있어 메모리의 크기와 Peripheral의 기준에 따라 선형적으로 칩을 변경할 수 있다는 것이다. 종류가 많아 지면을 통해 나열하기 어려우므로 마이크로프로세서 선택은 www.luminarymicro.com/products/product_selector_guide.html를 참고하도록 한다.

다음은 주요 IC에 탑재된 기능이다.

- 20, 25, 50MHz 동작
- Serial Wire Debug/Serial Wire Trace(SWD/SWT)

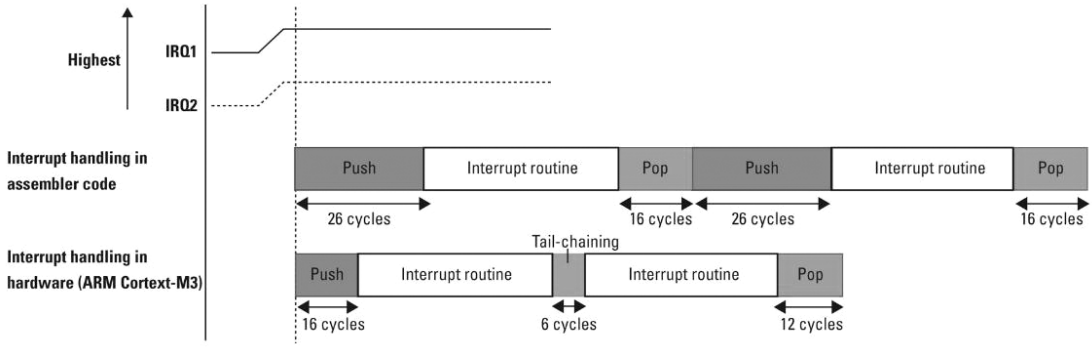


그림 7. Interrupt Response-Tail Chaining

(120-125)일 반 -ARM/전 자 -6 2008.5.27 4:3 PM페 이 지 125



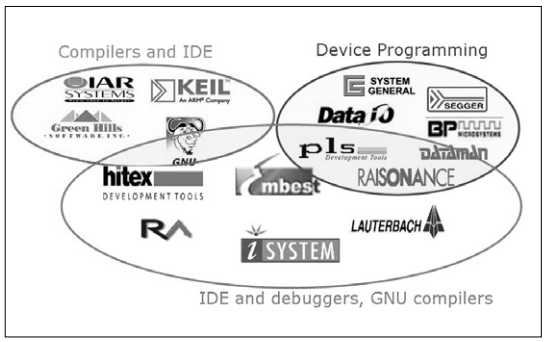


그림 9. 주요 개발 Tool


그림 10. 주요 RTOS 제공사

- 5V Tolerant Input/Output
- 2mA, 4mA, 8mA 출력 선택 가능
- FIFO가 있는 UART
- Synchronous Serial Interface
- 이더넷 컨트롤러
- CAN(Controller Area Network) 모듈
- I2C(Inter-Integrated Circuit)
- QEI(Quadrature Encoder Input) 모듈
- PWM 모듈
- PLL(Phase Lock Loop)
- 8KB~256KB Flash
- 2KB~64KB SRAM
- 4개의 타이머 모듈
- 아날로그 콤퍼레이터
- 3종류 ADC(FAST, FLEXIBLE, SMART)
- Watchdog Timer
- LDO Voltage Regulator

## 5. 개발 환경

Cortex-M3 코어는 RISC Processor이며 C/C++ 언어를 잘 처리할 수 있도록 하드웨어가 디자인되어 있기 때문에 컴파일러의 의존도가 무엇보다 높다. 또한 SWD/JTAG 등의 에뮬레이터를 이용하여 개발할 수 있도록 배려해 놓았기 때문에 손에 익고 성능 좋은 전용 컴파일러를 사용하는 것이 좋다. 대표적인 컴파일러로는 ARM社의 KEIL MDK 제품과 IAR社의 EWARM, Code Red Technologies社의 Code-Red, GNU 등이 있다. 개발 툴과 장비 및 Cortex-M3에 적합하도록 설계된 RTOS 업체의 로고를 그림 9, 그림 10에 나타낸다.

☆

마이크로프로세서가 진화하면서 같이 변화된 것이 있다. 8비트 MPU에서 데이터시트를 보고 비트를 제어하는 원시적인 코딩을 주로 했다면, 32비트 특히 Cortex-M3로 오면서 ANSI C/C++ 언어를 이용하여 하드웨어와 레지스터를 제어하는 방법이 보다 손쉬워진 것이다.

그러나 반대로 RISC 구조를 갖고 있는 32비트 마이크로프로세서는 컴파일러의 옵션/기능 선택 그리고 고급 언어를 이용하여 알고리즘을 표현, 해당 프로젝트의 성능과 처리속도, 생성 코드의 크기를 좌우할 수도 있다 그래서 툴 사용 기술과 논리적 사고의 언어 표현 방법, 정확한 알고리즘 구현을 개선할 필요가 있다. 다음에는 미국 Luminary Micro사의 Cortex™-M3 Code로 디자인된 LM3S 시리즈의 펌웨어 설계와 구현에 대해 기술 자료를 소개할 예정이다.

참 고 문 헌


- ARM CPU Roadmap, November, 2007
- LM Cortex-M3 presentation, 2008
- ARM presentation, 2007