

특집 : 전력전자제어용 마이크로프로세서의 기술동향

ARM Processor의 특성 및 최신동향

김 형 태
(펌웨어뱅크 대표이사)

1. 서론

i8051은 회로도에 해당하는 코어를 전 세계 반도체 회사들이 인텔로부터 사들여 자사가 구현하고자 하는 특정 소자를 붙여 8비트의 대명사 마이크로프로세서로 만들어 졌다. 이로 인해 MPU(Micro Processor Unit)를 변경하더라도 명령어 처리 엔진의 호환성이 뛰어났고 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 추가 되었다. [그림:Public ARM Processor Road map]에서 x축은 시간의 흐름을 나타내고 y축 위로의 진행 방향은 고속/고성능의 분포를 나타낸다. ARM7, ARM9 코어는 우리나라에서도 Silicon IP 형태로 구입한 회사가 상당히 있으나 몇몇 기업의 ARM9를 제외하고는 대중적인 IC 형태의 ARM 코어가 탑재된 국산 제품을 구입하여 Application으로 구현하기 쉽지 않다. 주로 화상, 음성, 멀티미디어 등의 신호처리와 Embedded 분야의 특정 용도로 사용되고 있는 추세이다.

ARM 코어는 11 이후로 계속적인 번호 증가로 성장 할 것으로 예상했으나 새로운 코어를 가지고 변신했다. Cortex라는 머리글자인데 다시 Cortex-M, Cortex-R, Cortex-A 시리즈

로 나눌 수 있다. 내부 엔진의 구성과 명령어 처리 방법, 디버깅 등의 기능을 강화하여 더욱 세련된 코어의 이름을 가진듯하다.

Cortex-M, Cortex-R, Cortex-A 시리즈는 모두 Thumb2의 Instruction 방식을 사용한다. Thumb2는 16비트 구성의 Thumb 명령어와 32비트 구성의 ARM 명령어를 혼용하여 사용하는 방식인데 코드 집적도를 높였다고 할 수 있다. 이런 방법은 우리나라에서도 이미 국제 특허를 가지고 있는 에이디칩스(www.adc.co.kr)의 EISC(Extendable Instruction Set Computing)와 비슷하다.

Cortex-M 시리즈 중에서 처음 나온 것은 Cortex-M3이며 마이크로프로세서 코어를 이용하여 제어 등에 활용하기 적합

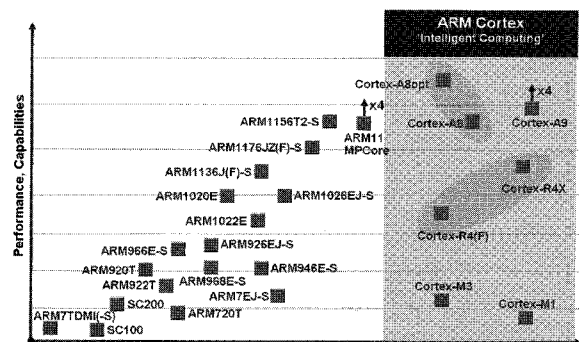


그림 1 Public ARM Processor Roadmap

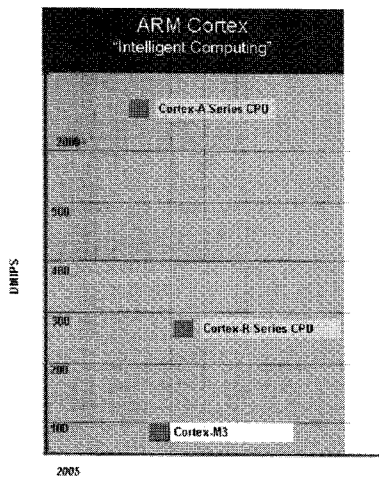


그림 2 ARM Cortex Family

하고 Thumb2 전용 명령어를 처리한다. 가격에 민감한 Product를 만들거나 코드집적도가 높고 다수의 인터럽트 처리가 필요한 응용개발에 알맞다고 할 수 있다.

Cortex-R 시리즈 중에서 처음 나온 것은 Cortex-R4이며 Real-Time 환경에 적용하기 위해 만들어진 Embedded 프로세서라고 할 수 있고 ARM, Thumb2 명령어를 처리한다.

Cortex-A 시리즈 중에서 처음 나온 것은 Cortex-A8이며 복잡한 OS와 User Application을 처리하기 위한 CPU라고 할 수 있고 ARM, Thumb2 명령어를 처리한다.

본 특집원고에서는 전력전자 분야에서 제어용으로 활용하기 편리하고 비교적 다루기 쉬운 Cortex-M3의 코어 특징, 구조, 성능 응용분야 그리고 개발 환경에 대해 알아보겠다. 현재 급속도로 다양한 제품군과 응용범위가 넓은 제품이 계속 출시 되고 있는 시점에서 검토 대상과 흐름을 파악하고자 하는 취지로 조사하게 되었음을 밝힌다.

2. ARM Cortex-M3

코텍스M3을 사용하면 좋은 편리한 이유가 있을까? 마이크로프로세서를 이용하여 제작되는 산업현장과 가전제품 Application은 32비트 추세로 현재 시장이 점점 Open되고 있다. 또한 빠르고 소비전력이 적으면서 많은 기능을 내포하기를 원한다. 이런 것을 수용하기 위해서는 32비트 급으로 가야하지만 구현되어야 하는 알고리즘의 비트 효율을 높이기 위해서는 32비트 자체만으로 개발 하기엔 무리가 있다. 예를 들어 AD(Analog Digital) 변환을 32비트로 하기도 어렵고 어드레스의 범위도 232의 4G영역을 전부 사용하기위한 알고리즘을 발굴하기도 힘들어 것이다. 무엇보다 명령어 처리를 32비트로만 한다면 플래시메모리와 RAM 영역의 비트 효율

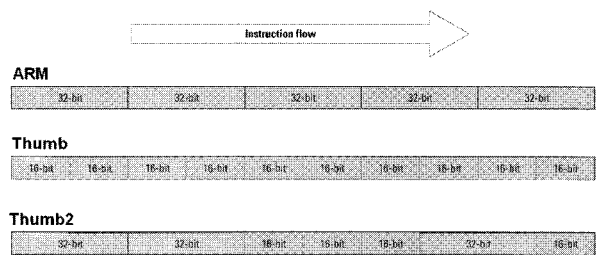


그림 3 Thumb2 명령의 처리 개념

은 더욱 떨어질 것이 당연하다. 그래서 대부분의 응용 Embedded 개발은 16비트 처리방법으로 효율을 높이고 32비트 처리로 속도를 높이는 추세를 이룰 것인데 이때 사용되는 것이 Thumb2 명령어 처리이다. 여기에 다이의 집적도가 높아 동일한 웨이퍼 위에 다수의 마이크로프로세서를 양산할 수 있다면 더욱 저렴해 지고 코어의 처리속도를 개선하므로 소비전력이 작아져 더욱 매력 있게 되는데 이 제품이 Cortex-M3가 된다.

2.1 특징

Cortex-M3의 Thumb2 core의 기술은 ARM사의 ARMv7의 특징과 Harvard Architecture를 가지고 있으며 명령어는 16비트 명령어 처리와 32비트 명령어 처리를 혼용으로 응대할 수 있어 메모리와 처리의 효율을 높이는 특징이 있다. (그림:Thumb2 명령의 처리 개념)에 보듯이 ARM 명령어와 Thumb를 혼합한 형태를 가지고 있다. 고속 32비트 명령어 처리는 ARM 처럼 32비트, 16비트 레지스터와 명령어 처리의 메모리 효율을 높이기 위한 명령어는 16비트로 처리한다.

2.2 구조

내부적으로 3-stage 파이프라인(pipeline) 처리와 C 언어로의 개발을 염두하여 설계된 독특한 구조를 가지고 있다. 예를 들어 C의 조건문 if()와 {}의 if~then에 해당하는 처리를 위한 IT 처리기, Table branch 전용의 TB{B|H} [Rbase, Rindex] 그리고 비트 field insert/extract/clear 처리를 하는 BFI, {S|U}BFX, BFC 비트반전 RBIT 처리를 할수 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 통하여 변경할 때 보다 효율적인 기계어 코드를 보장해 준다. 실제로 같은 프로그램을 ARM으로 컴파일 했을 때 보다 약 40%의 코드 절감 효과가 있다.

[그림:Cortex-M3 내부 구조]를 통하여 블록을 간단히 살펴보면 32Bit ALU(Arithmetic Logic Unit)에는 나누기와 32비트 곱셈기가 내장 되어 있고 NVIC(Integrated Nested

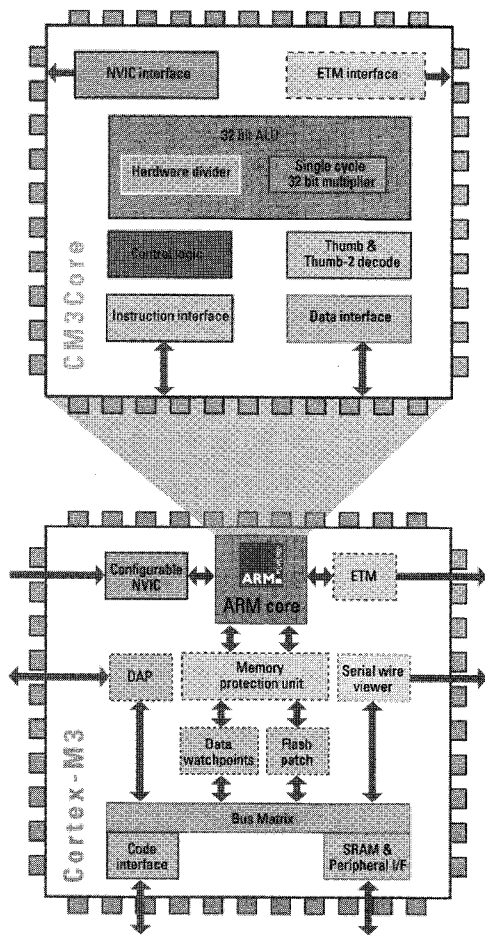


그림 4 Cortex-M3 내부 구조

Vector Interrupt Controller)가 있어 순서적으로 그리고 244개의 물리적인 인터럽트 처리가 가능하다. 또 Bus Matrix는 Bit Banding 기법을 사용하여 효율적인 Point 영역을 지정 할 수 있고 Bit Banding이 가능하다. 디버그를 위하여 DAP를 설치 SWD/JTAG의 연결이 용이하므로 보다 적은 비용으로 개발 장비를 구현 할 수 있게 한다.

Cortex-M3 Core를 어떻게 사용하고 디자인 하는가에 따라 다르겠지만 다음의 메모리 배치를 가지고 있다.

16 BIT THUMB [®] -2 INSTRUCTIONS			
ADC	ADD	AND	ASR
BIC	CMN	CMF	CPY
EOR	LSL	LSR	MOV
MUL	MVN	NEG	ORR
ROR	SBC	SUB	TST
REV	REVH	REVSH	SXTB
SXTH	UXTB	UXTH	
B	BL	BLX	CZB
IT			
LDR	LDRB	LDRH	LDRSB
LDRSH	STR	STRB	STRH
LDM	POP	PUSH	STM
BKPT	SVC	NOP	YIELD

그림 5 Thumb2 명령어

표 1 Cortex-M3의 메모리 배치

항목	크기(Byte)	설명
Vendor Specific	0.5GB	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	1GB	Intended for external devices and/or shared memory that needs ordering/non-buffered
External RAM	1GB	Intended for off chip memory
Peripheral	0.5G	Intended for normal peripherals. The bottom 1MB of the 32MB peripheral address space (0×40000000 - 0×400FFFFFF) is reserved for bit-band accesses. Accesses to the peripheral 32MB bit band alias region (0×42000000 - 0×43FFFFFF) are remapped to this 1MB address space.
SRAM	0.5G	Intended for on-chip SRAM. The bottom 1MB of the SRAM address space (0×20000000 - 0×200FFFFFF) is reserved for bit-band accesses. Accesses to the SRAM 32MB bit band alias region (0×22000000 - 0×23FFFFFF) are remapped to this 1MB address space.
Code	0.5GB	Reserved for code memory (flash, SRAM). This region is accessed via the Cortex-M3 I-Code and D-Code busses.

2.3 성능

Cortex-M3는 ARM7-S 코어를 설계할 때 보다 약 절반 정도의 Gate가 소용되어 구조가 간단하다. 따라서 회로 선폭이 가늘고 가까우므로 ARM7-S의 0.39mW/MHz에 비해 Cortex-M3 Core는 0.19mW/MHz로 소비전력이 적다.

인터럽트 처리는 [그림:Interrupt Response - Tail Chaining]에서 처럼 ARM7은 IRQ1에서 ISR1로 들어가고 32사이클의 Pop, Push 그리고 ISR2를 나오기까지 모두 84 cycle이 필요한 반면 Cortex-M3는 개선된 인터럽트 처리 방법(Tail Chaining)으로 34개의 Cycle이 필요하다. 인터럽트 처리도 ARM7 코어는 FIQ/IRQ 두 종류이지만 Cortex-M3는 200개 이상의 물리적인 인터럽트 처리가 가능하도록 설계되어 있다.

2.4 응용

Cortex-M3 코어를 이용하여 마이크로프로세서 IC 제품을 만드는 회사가 더욱 생겼지만 현 시점에서는 루미나리마이크로사(www.luminarymicro.com) ST사(www.st.com) 제품을 요약해 본다.

루미나리마이크로사는 AC, Step 모터제어, Ethernet, CAN 통신 등의 주로 제어기 개발 응용에 알맞다고 할 수 있으며 LM3S시리즈라고 하고 주요 특징으로는 28 SOIC, 48 LQFP, 100 LQFP의 104가지의 종류와 3가지 Package를 가지고 있어 메모리의 크기와 Peripheral의 기준에 따라 선행적으로 칩의 변경이 가능하다. 종류가 많아 지면을 통하여 나열하기 어려우므로 마이크로프로세서 선택은 [www.luminarymicro.com/products/product_selector_guide.html]를 참고하면 되겠다.

다음은 주요 IC에 탑재된 기능이다.

- 20, 25, 50MHz 동작
- Serial Wire Debug/Serial Wire Trace(SWD/SWT)
- 5V tolerant input/output

- 2mA, 4mA, 8mA 출력 선택 가능
- FIFO가 있는 UART
- Synchronous Serial Interface
- Ethernet Controller
- CAN(Controller Area Network) Module
- I2C(Inter-Integrated Circuit)
- QEI(Quadrature Encoder Input) Module
- PWM Module
- PLL
- 8KB ~ 256KB Flash
- 2KB ~ 64KB SRAM
- 4개의 Timer Modules
- Analog Comparators
- 3종류 ADC(FAST, FLEXIBLE, SMART)
- Watchdog Timer
- LDO Voltage Regulator

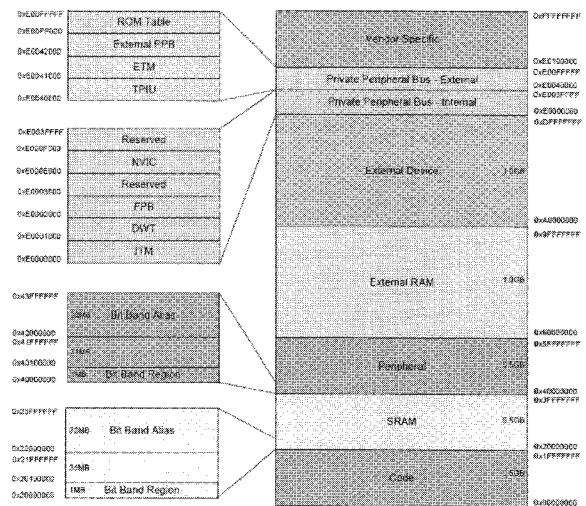


그림 6 Cortex-M3의 메모리 배치 블록

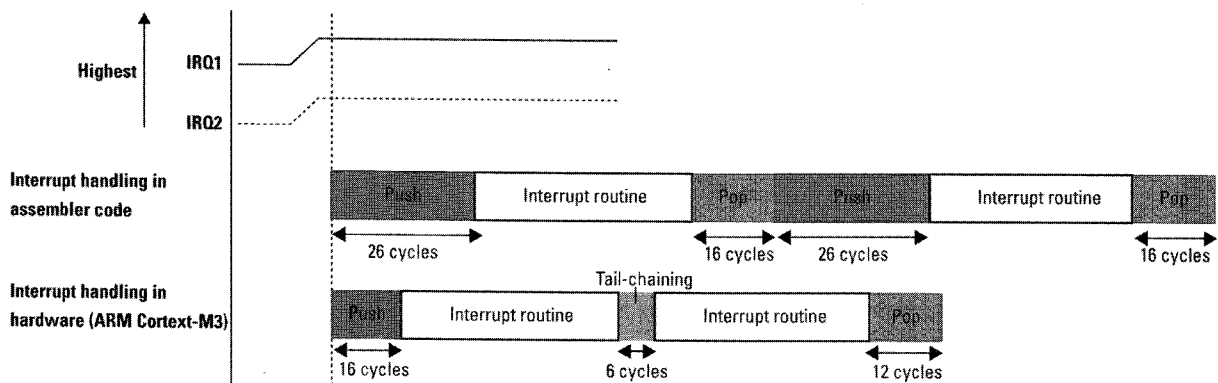


그림 7 Interrupt Response - Tail Chaining

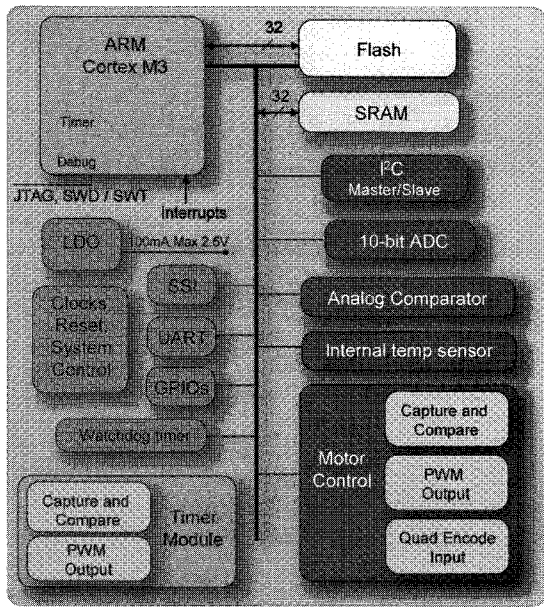


그림 8 LM 시리즈의 기본 구조

ST사는 STM32시리즈라고 하고 자동차, 제어, 가전의 Application에 알맞다. 다음은 주요 성능을 나타낸다.

- 2V-3.6V Supply
- 5V tolerant I/O
- Excellent safe clock modes
- Low-power modes with wake-up
- Internal RC
- ADC
- DAC
- 12 channels DMA
- PWM timers
- FSMC
- SDIO
- I2S
- Embedded reset
- -40/+105° C
- 36 QFN, 48/64 LQFP/QFN, 100 LQFP/BGA의 Package

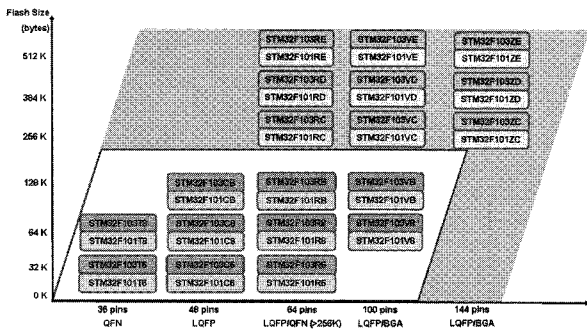


그림 9 STM32 Family

2.5 개발환경

코텍스M3 코어는 RISC processor이며 C/C++ 언어의 처리를 잘 할 수 있도록 하드웨어가 디자인 되어 있기 때문에 기계어 코드 기반인 Compiler 의존도가 높다. 또한 JTAG 등의 Simulator를 이용하여 개발 할 수 있도록 배려해 놓았기

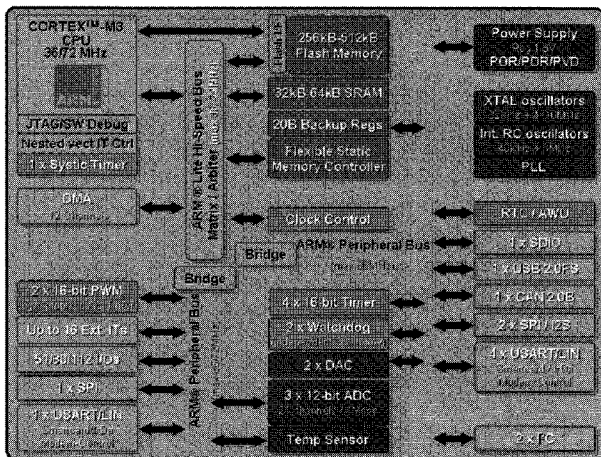


그림 10 STM32 블록도

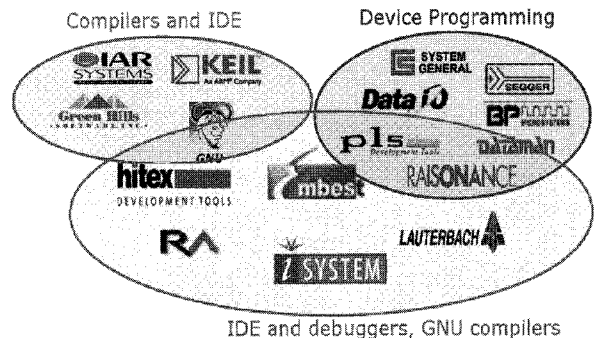


그림 11 주요 개발 Tool



그림 12 주요 RTOS 제공사

때문에 손에 익고 성능 좋은 전용 컴파일러를 사용하는 것이 좋다. 대표되는 컴파일러로는 ARM사의 KEIL MDK제품과 IAR사의 EWARM, GNU 등이 있다. 개발 툴과 장비 그리고 Cortex-M3에 적합하도록 설계된 RTOS 업체의 로고를 나타내었다.

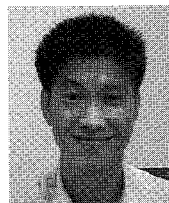
3. 결론

마이크로프로세서가 진화 하면서 같이 변화된 것이 있다. 8비트 MPU에서 데이터시트를 보고 비트 제어를 하는 원시적인 코딩을 주로 했다면 32비트 특히 Cortex-M3로 오면서 전달 인수와 리턴값과 변수, define 정의 등을 요약해 놓은 API(Application Program Interface)를 이용하여 프로그램 하는 것으로 바뀌었다. 제공된 API는 데이터 시트를 일일이 분석하여 코딩 하지 않아도 하드웨어 제어에 손쉽게 입문 할 수 있으며 코딩 적응을 빠르게 하는 장점이 있다. 또한 PC에서 운영되었던 C/C++ 언어로 프로그램 수정을 극히 적게 하여 Cortex-M3 알고리즘으로 옮겨 올 수 있다. ■

참고 문헌

- [1] ARM CPU Roadmap, EricSchorn, 2007, November
- [2] LM Cortex-M3 presentation, 2007
- [3] STM32 General marketing presentation, 2008
- [4] ARM presentation, 2007

< 필자 소개 >



김형태(金炯泰)

1965년 4월 9일생. 1992년 호서대 전자공학과 졸업. 1994년 동 대학원 졸업(석사). 1994년~1998년 대우전자 시스템사업부 근무. 2004년~현재 펌웨어뱅크주식회사 대표. 2006년~현재 산업기술대 메카트로닉스와 겸임교수. 강남대 컴퓨터미디어과 겸임교수.