임베디드 시스템

최 민

ARM 프로세서 역사

- **ARM 프로세서**
  - RISC 구조를 가지는 일련의 32비트 프로세서군을 가르키는 용어
  - 내부 구성이 간단, 동작속도가 빠르며, 전력 소비가 작음
  - 1985년 ARM1 의 개발 이후 현재의 Cortex 까지 발전
  - 휴대폰, 게임기 등의 모바일 기기 분야 CPU 최고 강자

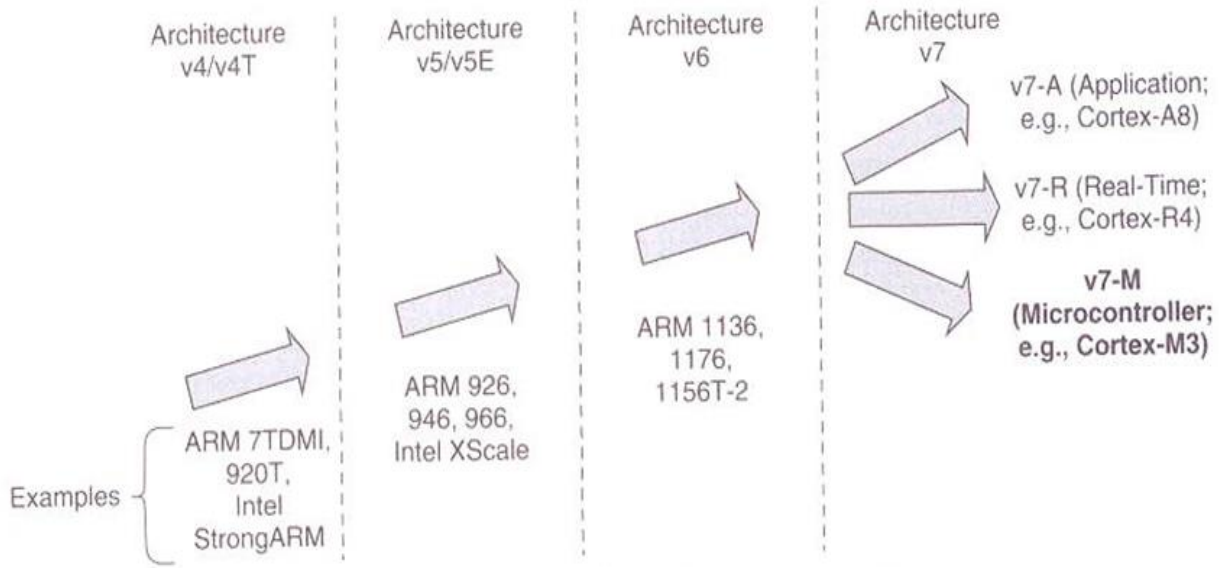


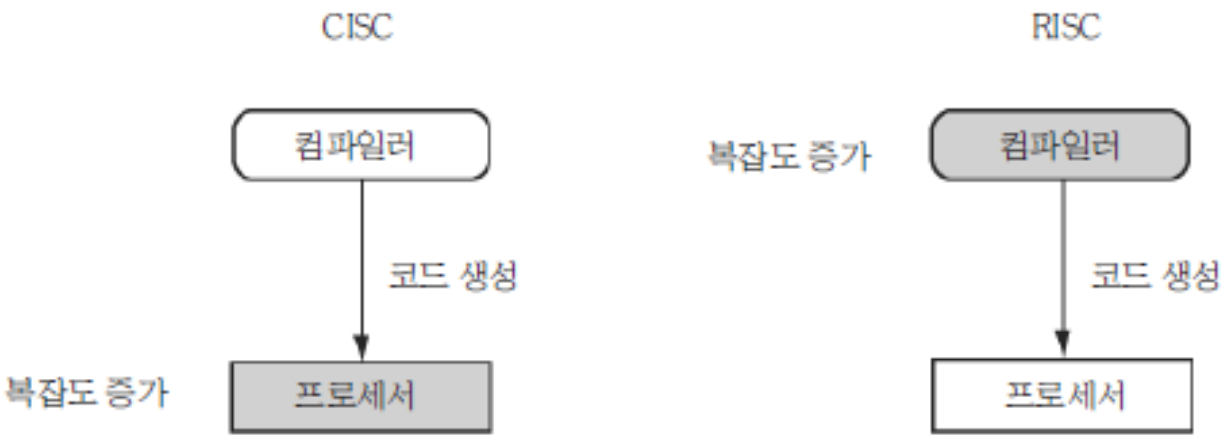
Figure 1.2 The Evolution of ARM Processor Architecture

Why we study ARM?

- **ARM vs. Intel**
  - 인텔 직원 수: 수십만명 내외
  - ARM 직원 수:1만명 내외
- **Intel**
  - MS (Windows/Windows CE)와 협력
- **ARM**
  - 반도체 설계 기술을 공급받는 모든 업체와 협력
  - Linux, Apple Iphone/IPad, Google Android, Nokia Symbian, 퀄컴, 삼성, LG, 엔비디아, HP, 델, 에이서, 레노보, Analog Devices, Atmel, Cirrus, Fujitsu, IBM, Infineon, Mitsubishi, Motorola, National Semiconductor, NEC, Philips, Sharp, ST Microelectronics, Texas Instruments, Toshiba , NEC, Philips, Sharp, ST Microelectronics, Texas Instruments, Toshiba 등 다수

RISC Architecture 특징

- **RISC** 구조
  - 매우 빠른 속도로 한 클럭 안에 실행될 수 있는 간결하면서도 막강한 명령어들을 가지고 있는 아키텍처
  - 하드웨어에 의해 수행되는 명령어들의 복잡도를 줄이는 것을 목표
  - 컴파일러 의존적
    - CISC는 하드웨어복잡도에 초점, RISC는 컴파일러 복잡도에 초점



ARM 프로세서 특징

- **Instructions**
  - Reduced instruction sets/single cycle/fixed length
- **Pipeline**
  - Decoded in one stage/no need for microcode
- **Registers**
  - Large set of general-purpose registers
- **Load/store architecture**
  - Data processing instructions applied to registers only
  - load/store instructions transfers data from memory to registers
- 명령어 파이프라인
  - 5-stage pipeline

ARM Processor 과거 사례

패밀리	Architecture Version	주요모델	비고
ARM7TDMI	ARMv4T	ARM7TDMI(-S)	3 Stage pipe line, Sony Game Boy Advance, Nitendo DS, Apple iPod, Lego MindStorm NXT
		ARM710T	MMU, 8KB cache
		ARM720T	
	ARMv5TEJ	ARM7EJ-S	
StrongARM	ARMv4	SA-110	DEC에서 개발 Apple Newton 2x00 PDA
ARM8	ARMv4	ARM810	별로 많이 사용되지 않음
ARM9TDMI	ARMv4T	ARM9TDMI	5 Stage pipe line (1997)
		ARM920T	16KB/16KB cache 휴대폰 : 삼성, LG, Nokia, Motorora, Hic, Sony Ericsson 등, Nintendo DS/DSi
		ARM922T	8KB/8KB cache
		ARM940T	4KB/4KB cache

ARM Processor 과거 사례

- **ARM7**
  - 폰 노이만 아키텍쳐에 기초
  - Code / Data 를 위한 공유된, 단일 메모리 공간
  - 선형 32-bit 주소 영
- **ARM9**
  - Harvard architecture 지원
  - Code / Data를 위한 개별적인 메모리 포트
  - Code / Data 에 대한 동시 액세스 제공

ARM Processor 과거 사례

패밀리	Architecture Version	주요모델	비고
XScale	ARMv5TE	PXA210	Intel사에서 개발 (2002,2004) (2006년 Marvell에 매각) PDA, 스마트폰, 네비게이션등에 사용
		PXA250	
		PXA271, 272	
ARM11	ARMv6	ARM1136J(F)-S	8 Stage pipe line(2001) 각종 휴대폰에 많이 사용됨
	ARMv6T2	ARM1156T2(F)-S	9 Stage pipe line
	ARMv6KZ	ARM1176JZ(F)-S	Apple iPhone(Original,3G),Apple iPod touch

ARM Cortex 프로세서

- **Cortex processor : ARMv7** 아키텍처의 코어를 가지는 프로세서(차세대 프로세서에는 또 따른 이름을 부여할 예정)
  - A (Application), R (Real Time), M (Microcontroller) Series
  - 각 시리즈 별로 1~10(또는 그 이상)의 숫자를 부여, 큰 숫자일 수록 고성능(ex. Cortex-A8, Cortex-A9, Cortex-M0, Cortex-M3)
- **Cortex-A series**
  - 가장 고성능의 프로세서
    - 복잡한 운영 환경, 고성능이 요구되는 어플리케이션의 구현에 적합
  - Architecture: ARMv7-A
  - Instruction set : ARM, Thumb-2
  - 프로세서 종류 : Cortex-A8 , Cortex-A9
  - PMP나 스마트폰, 넷북과 같은 기기에 많이 사용

ARM Cortex 프로세서

- **Cortex-R series**
  - Real time 동작이 요구되는 high-end embedded system용
  - Architecture: ARMv7-R
  - Instruction set : ARM, Thumb-2
  - 프로세서 종류 : Cortex-R4 , Cortex-R4F , Cortex-R4F
  - 임베디드 제품에서 복잡한 알고리즘 및 실시간 작업처리 가능
- **Cortex-M series**
  - 8bit, 16bit MCU 시장을 타겟으로한 프로세서
    - 마이크로컨트롤러 및 FPGA에 최적화된 프로세서
  - Architecture: ARMv7-M
  - Instruction set : Thumb-2 only
  - 프로세서 종류 : Cortex-M0 , Cortex-M1 , Cortex-M3
  - 가격에 민감한 임베디드 어플리케이션 분야에 사용

ARM Cortex 프로세서

패밀리	Architecture Version	주요모델	비고
Cortex	ARMv7-A	Cortex-A8	(2005) Apple iPhone 3GS 외
		Cortex-A9	(2007)
		Cortex-A9 MPCore	Multi Processor Core (2009) IPhone 4G 삼성 Galaxy S (Core 4개 사용)
	ARMv7-R	Cortex-R4(F)	Cortex-R 시리즈
	ARMv7-ME	Cortex-M4	(2010)
	ARMv7-M	Cortex-M3	(2004) ST STM32F시리즈, NXP 시리즈, TI Stellaris 시리즈
	ARMv6-M	Cortex-M0	(2009) NXP LPC1100
		Cortex-M1	(2007) Actel ProASIC3, IGL00

ARM Cortex 프로세서의 명령어 구조

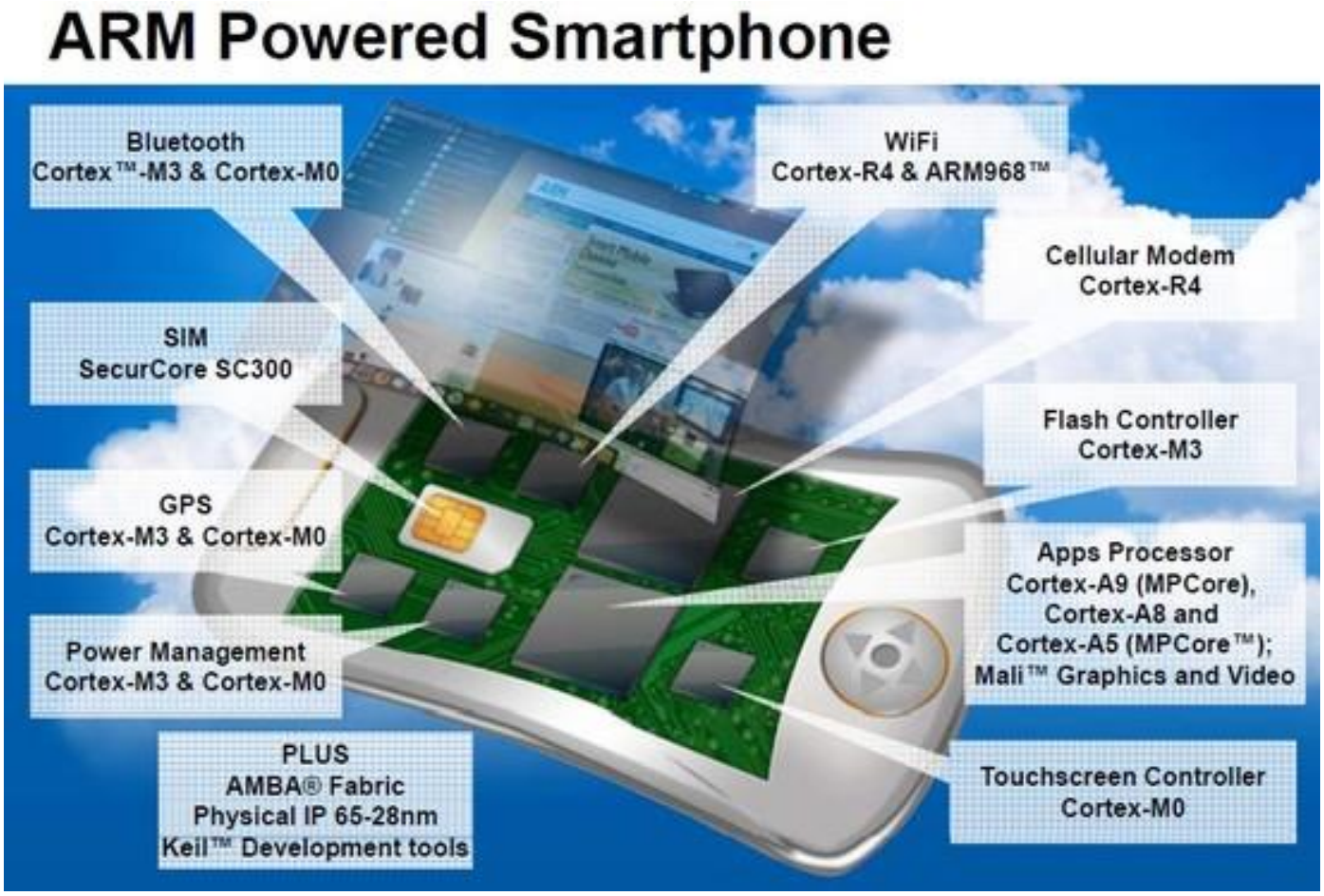
- **ARM Instruction Set의 발전**

아키텍쳐	명령어 세트	비고
ARMv4 까지	ARM	ARM : 32비트 명령어로 구성
ARMv4T ~ ARMv6	ARM + Thumb	Thumb : 16비트 명령어로 구성
ARMv7	ARM + Thumb2	Thumb2 : 16비트, 32비트 명령어로 구성

- **Thumb-2 명령어 세트**
  - 16비트와 32비트의 명령어를 모두 포함
  - 기존의 명령어 세트가 가지고 있던 문제점인 32비트 명령어의 코드 사이즈 증가와 16비트 명령어의 부족한 성능, 32비트와 16비트 간의 전환에 따른 성능 저하의 문제를 모두 해결

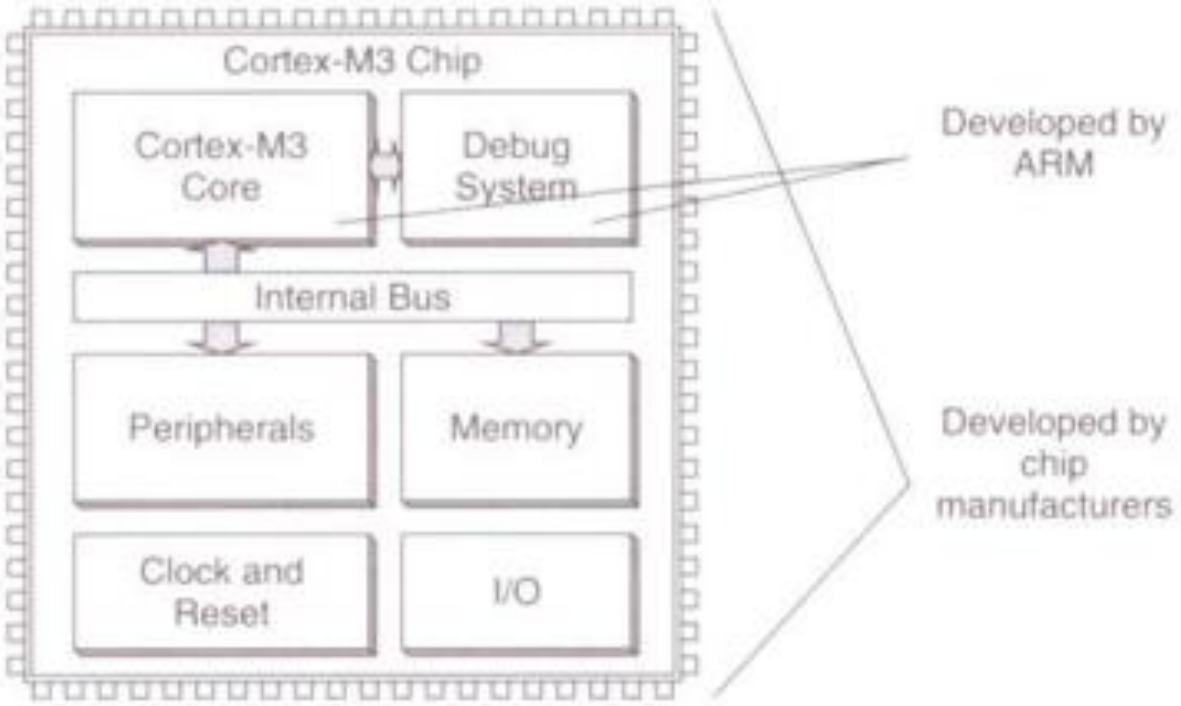
Why we study ARM?

- 스마트폰 한대에 다양한 종류의 **ARM** 프로세서 칩셋 사용



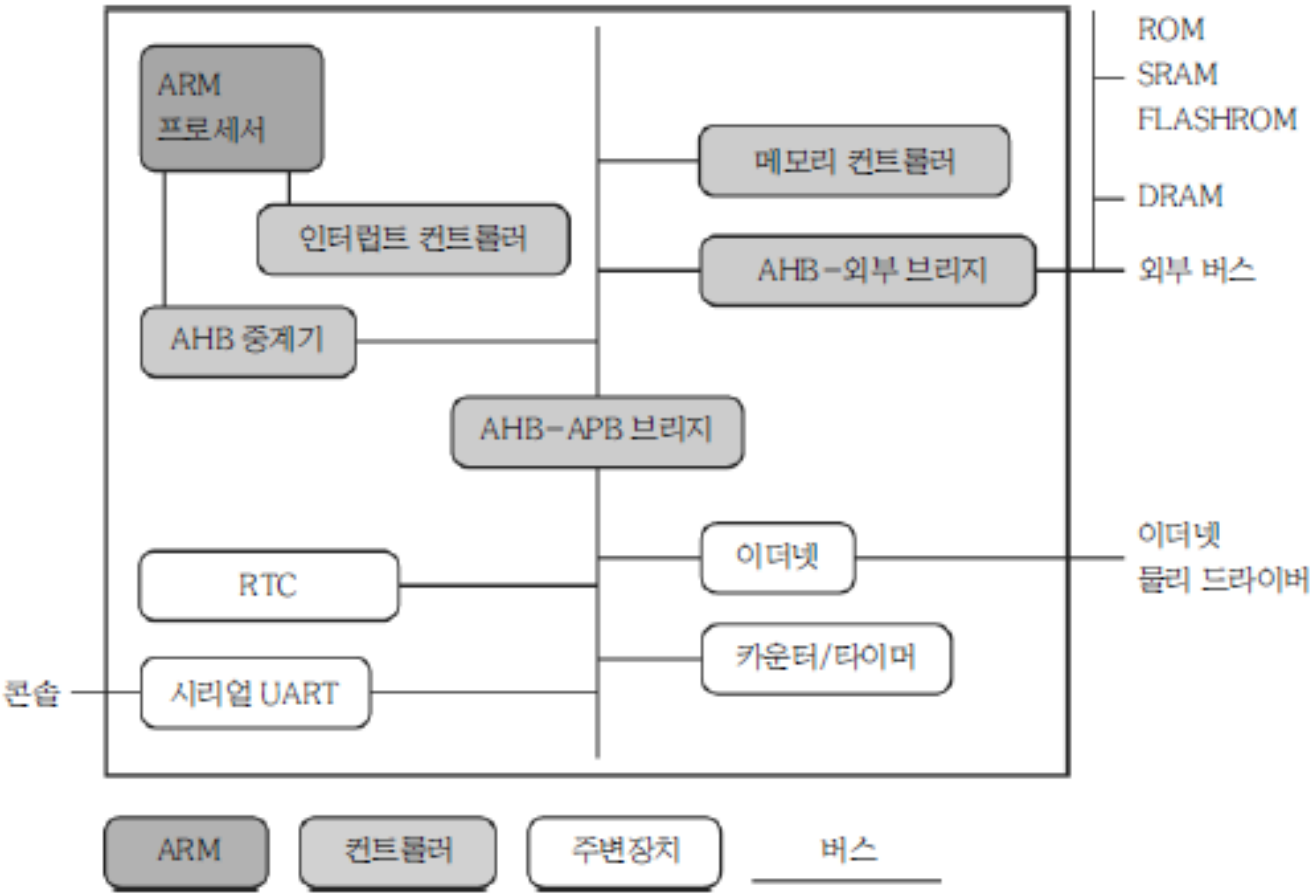
ARM-M3 프로세서

- **"Cortex-M3** 기반의 마이크로컨트롤러“
  - 프로세서 코어를 바탕으로 여러 가지 주변 장치를 추가하여 완성된 마이크로 컨트롤러



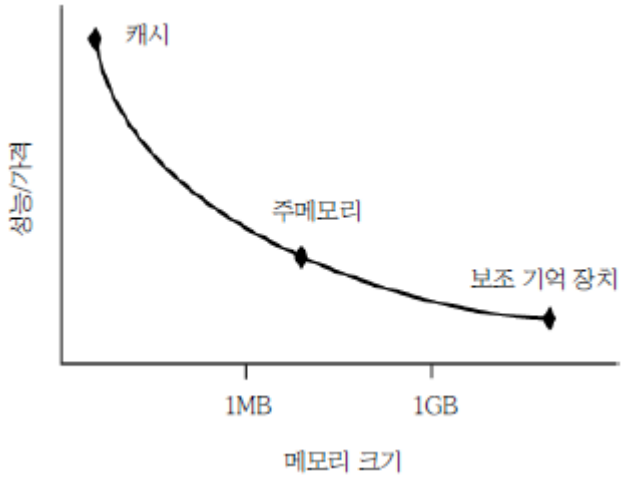
ARM SoC(System on a chip) 아키텍처

- **ARM SoC** 컨트롤러 아키텍쳐



메모리 구성

- 메모리 계층구조
  - 캐시 : 가장 빠른 메모리, 물리적으로 ARM 프로세서 코어에 가장 가까이 위치
  - 보조 기억 장치 : 가장 느린 메모리, 프로세서에서 가장 멀리 떨어져 있다.
- 메모리가 프로세서에 가까울수록 : 비용↑ 용량↓



메모리의 트레이드오프(trade-off)

ARM Cortex-M3 프로세서 특징

- 전력 소모, 뛰어난 연산 성능, 빠른 인터럽트 응답
  - ARMv7-M architecture의 Cortex-M3 프로세서 코어
  - Thumb-2 명령어 세트(16비트와 32비트 명령어가 혼합)
  - Harvard Architecture(명령 버스와 데이터 버스가 분리)
  - 빠른 인터럽트 처리 : 12사이클에 처리 가능
- **No ARM instruction set support**
  - Thumb-2 only
- **No Cache , No MMU(Memory Management Unit)**

ARM Cortex-M3 vs. ARM7TDMI

Features	ARM7TDMI-S	Cortex-M3
Architecture	ARMv4T (von Neumann)	ARMv7-M (Harvard)
ISA Support	Thumb / ARM	Thumb / Thumb-2
Pipeline	3-Stage	3-Stage + branch speculation
Interrupts	FIQ / IRQ	NMI + 1 to 240 Physical Interrupts
Interrupt Latency	24-42 Cycles	12 Cycles
Sleep Modes	None	Integrated
Memory Protection	None	8 region Memory Protection Unit
Dhrystone	0.95 DMIPS/MHz (ARM mode)	1.25 DMIPS/MHz
Power Consumption	0.28mW/MHz	0.19mW/MHz
Area	0.62mm2 (Core Only)	0.86mm2 (Core & Peripherals)*

ARM Cortex-M3 Core

- **ARM Cortex-M3 Core : Architecture**

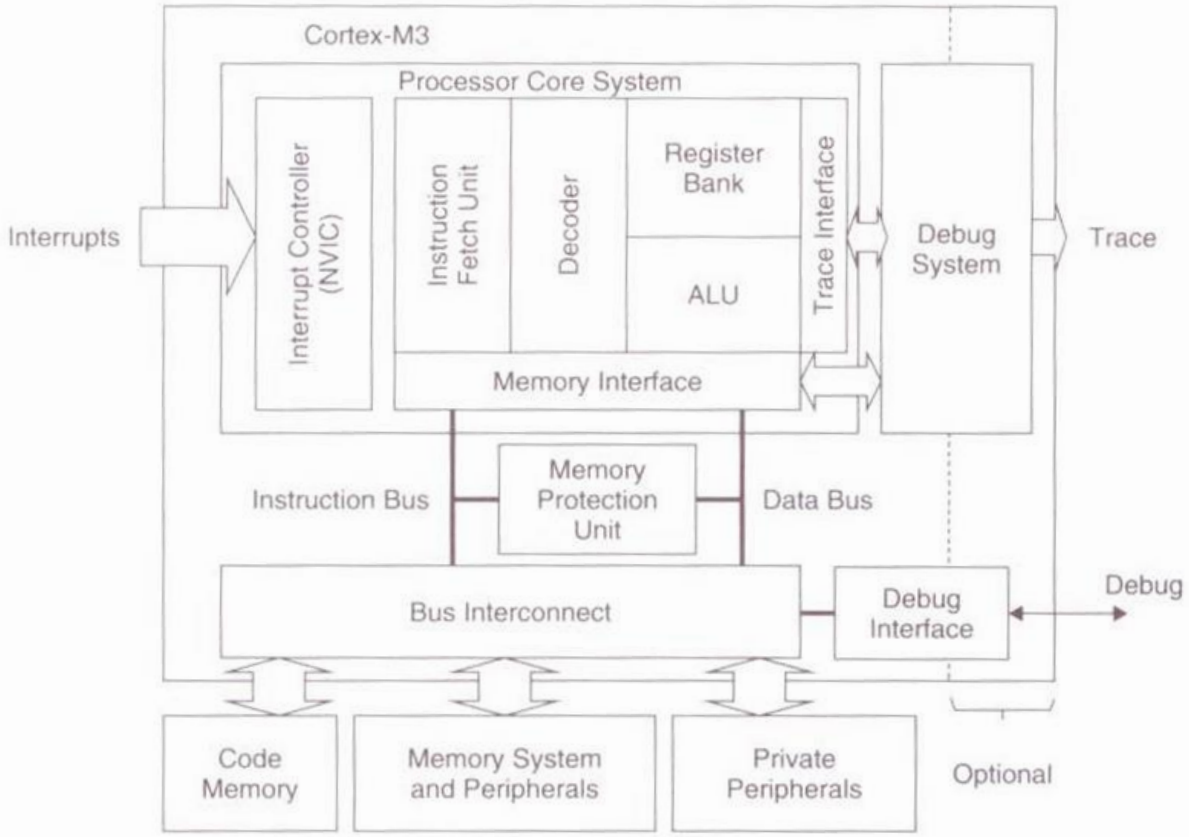
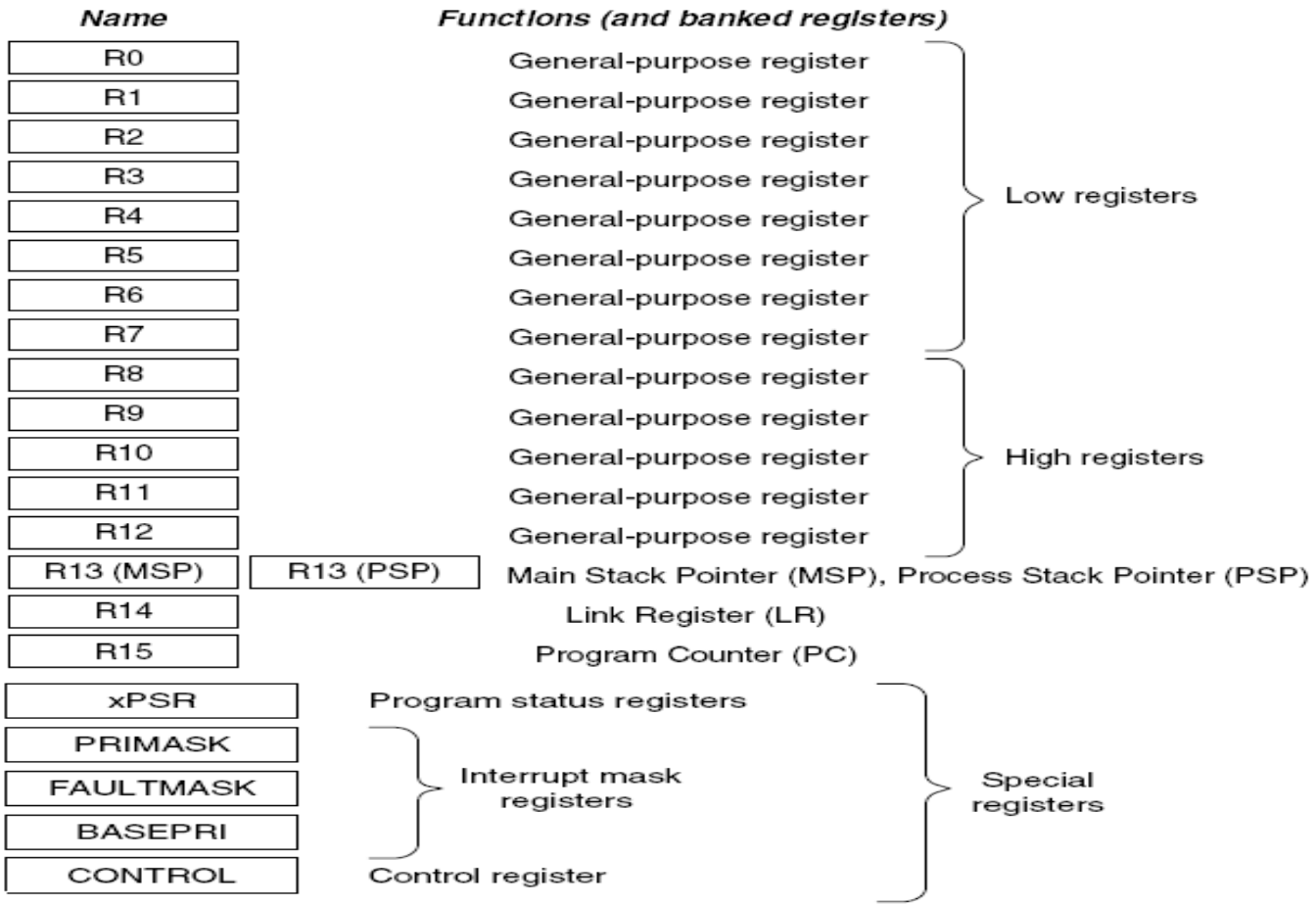


Figure 2.1 A Simplified View of the Cortex-M3

ARM Cortex-M3 Core

- **ARM Cortex-M3 Core : Registers**



ARM Cortex-M3 Core

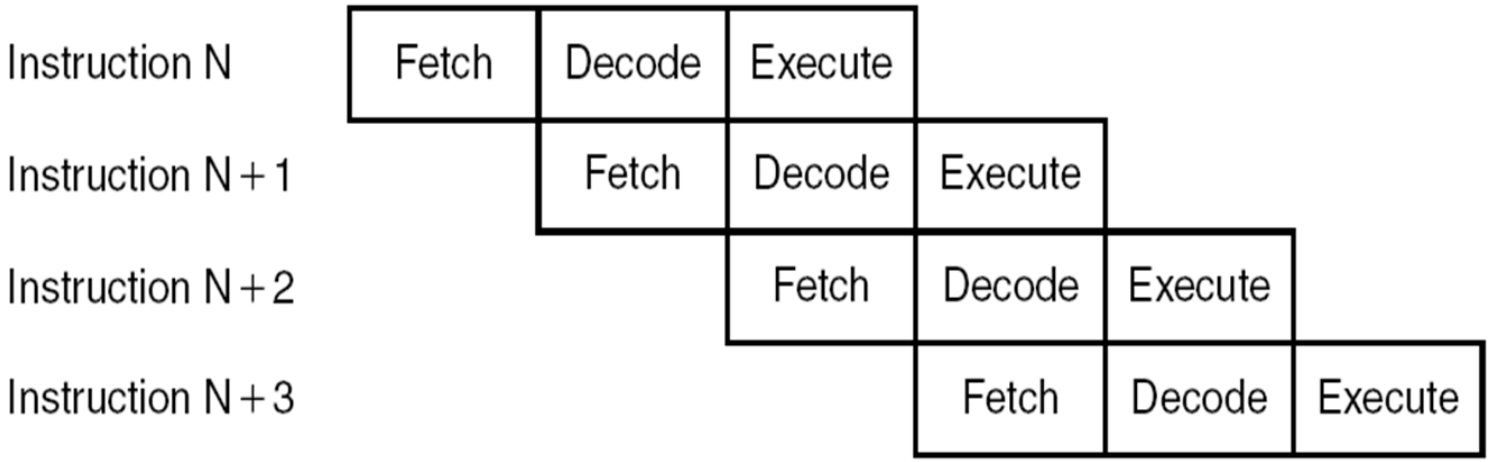
- **General-purpose register (범용 레지스터) : R0~R12**
  - Low register : R0~R7의 8개 레지스터
    - 모든 Thum2 명령어는 접근이 가능
  - High register : R8~R12의 5개 레지스터
    - 16비트 명령어는 접근이 불가능
- **SP (Stack Pointer) : R13**
  - MSP (Main Stack Ponter) : 디폴트로 지정
    - 운영체제의 커널과 예외(exception) 핸들러가 사용
  - PSP (Process Stack Pointer)
    - 사용자 프로그램(application code)이 사용
- **LR (Link Register) : R14**
  - 서브루틴이 호출 될 때 복귀 주소(return address)를 저장

ARM Cortex-M3 Core

- **PC (Program Counter) : R15**
  - 현재 수행중인 프로그램의 주소를 저장하는 레지스터이다.
- **xPSR (Program Status Register)**
  - 수행중인 프로그램의 상태를 저장하는 레지스터이다.
- **Special registers**
  - PRIMASK : NMI(non-maskable interrupt)와 Hard fault를 제외한 모든 인터럽트를 비활성화시키는 레지스터
  - FAULTMASK : NMI를 제외한 모든 인터럽트를 비활성화 시키는 레지스터
  - BASEPRI : 특정 우선 순위 또는 그 이하의 우선 순위를 가지는 인터럽트를 비활성화 시키는 레지스터
  - CONTROL : 스택 포인터 및 접근 모드의 선택을 위한 레지스터

ARM Cortex-M3 Core

- **3-Stage Pipeline**
  - 클럭 속도를 높이지 않고도 프로세서의 동작 속도를 높일 수 있는 여러 가지 방법 중 하나
  - ① t 시간 : 1번 명령어가 인출
  - ② (t+1) 시간 : 1번 명령어는 해독, 2번 명령어가 인출
  - ③ (t+2) 시간 : 1번 명령어는 실행, 2번 명령어는 해독, 3번 명령어가 인출



ARM Cortex-M3 예외

- 예외(**Exceptions**)
  - 프로세서가 동작 도중 예상하지 못한 상황, 예외적인 상황, 또는 외부 인터럽트가 발생하는 상황
  - 최대 255개까지 가능(1~255)
  - System Exception과 External Interrupt로 분류
- **System Exception**
  - Cortex-M3 프로세서 내에서 발생
  - Reset, NMI, Hard fault 등
  - 최대 15개 (1~15번) 까지 가능
- **External Interrupt**
  - 주변 장치(GPIO, Timer등)에서 발생
  - 최대 240개(16~255번)까지 가능
  - MCU 제조사 별로 구성이 다르다.

ARM Cortex-M3 예외

- **System Exceptions**

번호	종 류	우선 순위	설 명
1	Reset	-3(가장 높음)	리셋
2	NMI	-2	Non-maskable Interrupt (발생 차단 불가 인터럽트) 발생
3	Hard Fault	-1	Fault 핸들러가 별도로 지정되지 않은 오류의 발생
4	MemManage Fault	설정 가능	메모리 관리 오류 : MPU 설정 위반, 허가되지 않은 메모리에 접근
5	Bus Fault	설정 가능	버스 에러 발생
6	Usage Fault	설정 가능	프로그램 에러에 의한 예외 발생
7-10	Reserved	-	예비
11	SVCall	설정 가능	시스템 서비스 호출 (System Service Call)
12	Debug Monitor	설정 가능	디버그 모니터(break point, watch point, 외부 디버그 요청) 발생
13	Reserved	-	예비
14	PendSV	설정 가능	시스템 장치의 보류 가능한 호출 발생
15	SYSTICK	설정 가능	시스템 타이머 (System Tick timer)

ARM Cortex-M3 예외

- **GPIO,** 타이머 등과 같은 주변장치에 의해서 발생되는 인터럽트
  - 16번부터 255번까지 240개를 가질 수 있다.
  - MCU 제조사별, MCU 모델별로 각각 다르게 구성된다.

Exception Number	Exception Type	Priority
16	External Interrupt #0	Programmable
17	External Interrupt #1	Programmable
…	…	…
255	External Interrupt #239	Programmable

ARM Cortex-M3 예외

- **Exception**은 중첩되거나 연속으로 발생할 수 있다.
  - 각각의 Exception에 우선 순위 설정이 필요
- 고정된 우선순위를 가지는 **Exception : 변경 불가**
  - Reset : -3, NMI : -2, Hard Fault : -1
- 나머지 **Exception**은 사용자가 우선순위 설정이 가능
  - 우선 순위 설정 : 3bit 에서 8bit 사용가능
  - 8단계(1~8) ~ 256단계(1~256)

ARM Cortex-M3 예외

- **Cortex-M3 프로세서의 우선 순위 설정**
  - 우선 순위 설정에 3 ~ 8 bit (8~256 단계) 할당 가능
  - 하지만 우선 순위 단계가 많아질수록 하드웨어의 게이트 수가 늘어나고 전력 소모가 커짐
  - 따라서 대부분의 Cotex-M3 기반의 MCU는 8비트보다 작게 사용
- **Interrupt Handler**
  - 어떤 Interrupt 발생했을 때 이에 대응하는 Interrupt Service Routine 으로 연결해주는 역할을 하는 명령.
  - Interrupt Service Routine 의 주소가 저장Interrupt Service Routine (ISR)
- **Interrupt Service Routine(ISR)**
  - 인터럽트 발생시 이에 대응하는 프로세서의 동작 (즉, 인터럽트의 처리 동작) 을 정의한 함수

ARM Cortex-M3 예외

- **중첩 인터럽트(Nested Interrupt)**
  - 현재 실행중인 인터럽트보다 우선 순위가 더 높은 인터럽트가 발생할 경우, 현재의 동작을 멈추고 높은 순위의 인터럽트를 먼저 처리하는 기능
  - 이때 먼저 실행 중이던 인터럽트는 우선 순위가 높은 인터럽트가 끝나면 다시 실행
- **NVIC는 미리 정해진 우선 순위에 따라 자동적으로 중첩 인터럽트 동작을 수행함**
  - 이 경우 레지스터의 stacking 및 unstacking은 자동으로 처리됨