

ARM 기초

유튜브 주소 : <https://youtu.be/3PiW4HeLSG8>

ARM 개요

ARM 레지스터

Cortex-M3

ARM 개요

- ARM : Advanced RISC Machine
 - 임베디드 기기에 많이 사용되는 RISC 프로세서
 - RISC : Reduced Instruction Set Computer(축소 명령어 집합 컴퓨터)
 - CPU 명령어의 개수를 줄여 명령어 해석시간을 줄임으로써 개별 명령어의 실행속도를 빠르게 한 컴퓨터
 - 사물인터넷, 안드로이드, 랩탑, 인터넷 모뎀, 닌텐도 등에 사용되는 CPU 체계
 - 일반적인 데스크탑 PC - amd, intel cpu 명령체계로 이루어짐(CISC)
 - e.g. x86(32비트), x64(64비트), AMD64(64비트)
 - ARM용 기기 - RISC 명령체계로 이루어짐

ARM 개요

- RISC 특징
- 고정 길이의 명령어(16 or 32 bit)를 사용하여 더욱 빠르게 해석할 수 있다
- 간단하고 적은 종류의 명령어와 주소 지정 모드를 사용한다.
- 명령어가 하드웨어적이므로 호환성이 낮다.
- 모든 연산은 하나의 클럭으로 실행되므로 파이프라인을 기다리게 하지 않는다.
- 많은 수의 범용 레지스터(GPR)를 사용하여 메모리 접근을 줄인다.
- CISC, RISC 비교

| | CISC | RISC |
|----------|------|------|
| 명령어의 수 | 많다 | 적다 |
| 레지스터 | 적다 | 많다 |
| 처리속도 | 느리다 | 빠르다 |
| 설계(내부구조) | 복잡하다 | 간단하다 |
| 전력소모 | 많다 | 적다 |

ARM 개요

- ARM 프로세서 종류 - 크게 3가지로 분류

- ARM Cortex-A (Application)

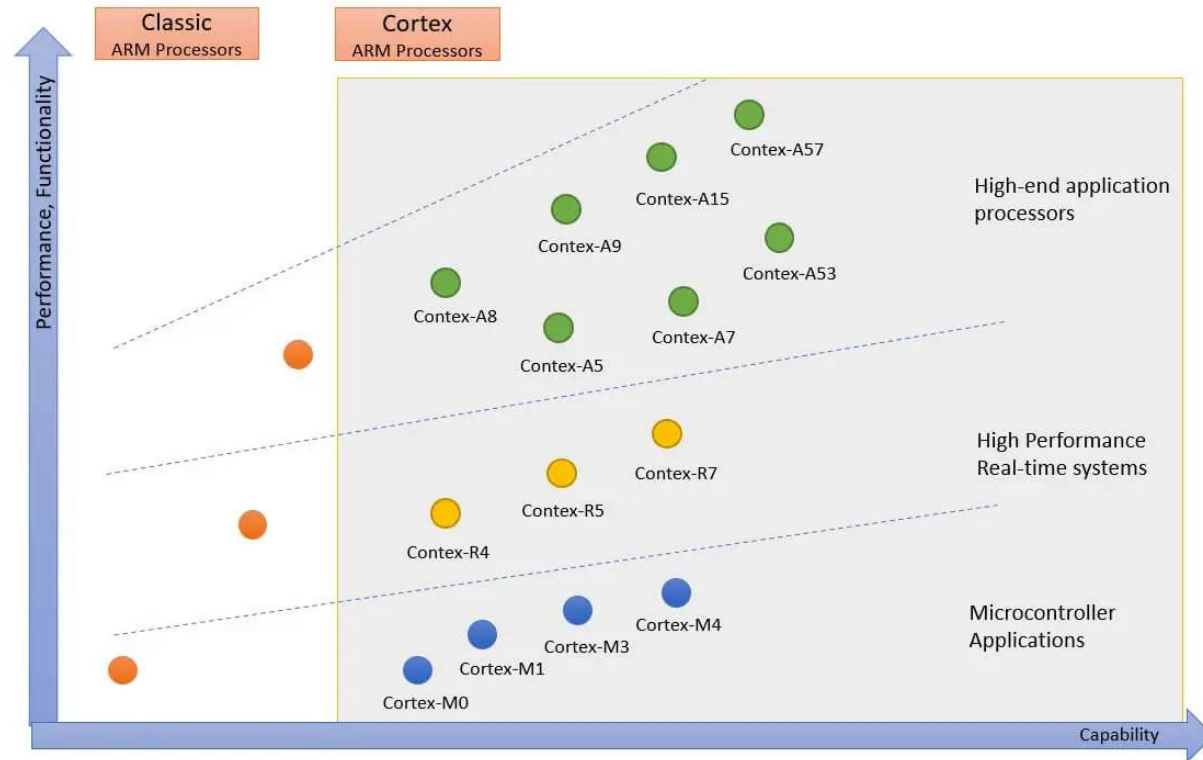
- 복잡한 OS 및 사용자 응용 프로그램에 사용하는 응용 프로그램 프로세서 계열

- ARM Cortex-R (Real-Time)

- 실시간 시스템을 위한 임베디드 프로세서 계열

- ARM Cortex-M (Microcontroller)

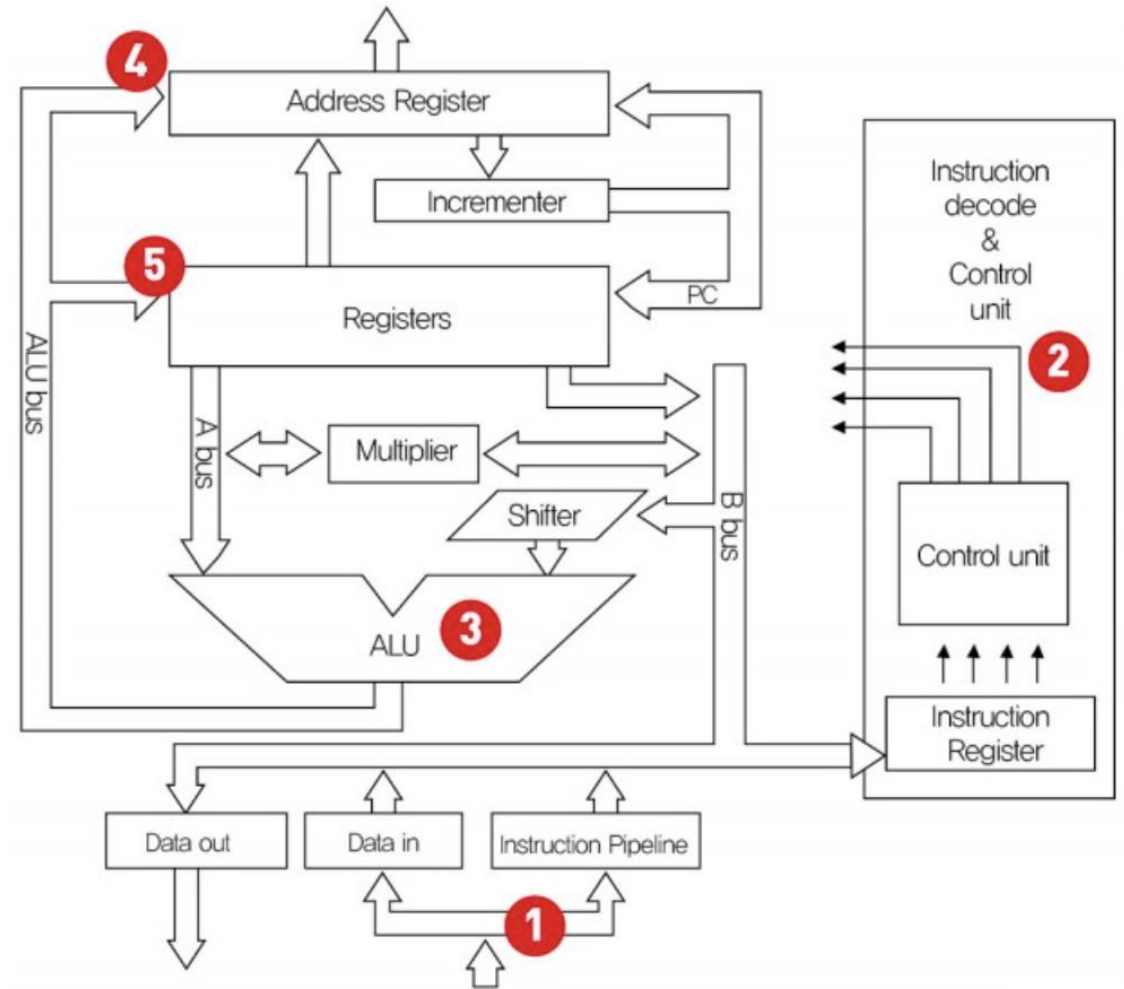
- 저가형 응용프로그램에 최적화된 임베디드 프로세서 계열



ARM 개요

- ARM 구조(ARM7TDMI Processor Block Diagram)

- 1) 메모리에서 명령어 fetch 후 pipeline에 집어 넣음
- 2) pipeline의 명령어를 control unit에서 해독
- 3) 해독된 명령어를 ALU에서 실행
- 4) 메모리 접근 명령어일 경우 메모리 접근할 주소 계산
- 5) 결과를 다시 레지스터에 write



ARM 레지스터

- ARM 동작 모드 : 어떤 권한을 지니고 어떤 동작을 수행하는지 나타냄

| 모드 | 권한 | 설명 |
|------------------|---------------|--|
| User | Un-Privileged | 일반적인 사용자 프로그램이 실행되는 모드 |
| FIQ | Privileged | 빠른 인터럽트 처리 모드 User 모드와 SVC 모드보다 높은 우선순위로 인터럽트 처리 |
| IRQ | | 일반적인 인터럽트 처리 모드 |
| Supervisor (SVC) | | OS의 커널모드에서 사용되는 모드 시스템 리소스(I/O, memory, register) 관리 권한을 지님 |
| Abort | | 인터럽트나 예외 처리 중 메모리와 관련된 오류 발생 시 해당 오류를 해결하거나 처리 하기 위한 모드 |
| Undefined | | 일반적으로 정의되지 않은 명령어를 처리하기 위한 모드 |
| System | | User 모드와 동일한 용도이나 프로세서가 모든 코드에 대해 접근 권한을 지니기 때문에 보안성이 낮음 |

ARM 레지스터

- 전통적인 ARM(ARM7, ARM9) 레지스터 구조
- 37개의 32 bit 레지스터 구성
- 동작모드 별 사용가능한 레지스터가 다름
- R0 ~ R15 : 범용 레지스터
- R13, R14, R15 : 특별한 용도로 사용
- R13 : Stack Pointer
- R14 : Link Register
- R15 : Program Counter
- CPSR : 현재 status를 저장하는 레지스터
- Banked register : 각 모드 별 따로 존재(20개)

| System & User | FIQ | Supervisor | Abort | IRQ | Undefined |
|---------------|----------|------------|----------|----------|-----------|
| R0 | R0 | R0 | R0 | R0 | R0 |
| R1 | R1 | R1 | R1 | R1 | R1 |
| R2 | R2 | R2 | R2 | R2 | R2 |
| R3 | R3 | R3 | R3 | R3 | R3 |
| R4 | R4 | R4 | R4 | R4 | R4 |
| R5 | R5 | R5 | R5 | R5 | R5 |
| R6 | R6 | R6 | R6 | R6 | R6 |
| R7 | R7 | R7 | R7 | R7 | R7 |
| R8 | R8_fiq | R8 | R8 | R8 | R8 |
| R9 | R9_fiq | R9 | R9 | R9 | R9 |
| R10 | R10_fiq | R10 | R10 | R10 | R10 |
| R11 | R11_fiq | R11 | R11 | R11 | R11 |
| R12 | R12_fiq | R12 | R12 | R12 | R12 |
| R13 | R13_fiq | R13_svc | R13_abt | R13_irq | R13_und |
| R14 | R14_fiq | R14_svc | R14_abt | R14_irq | R14_und |
| R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) | R15 (PC) |

ARM State Program Status Registers

| | | | | | |
|------|----------|----------|----------|----------|----------|
| CPSR | CPSR | CPSR | CPSR | CPSR | CPSR |
| | SPSR_fiq | SPSR_svc | SPSR_abt | SPSR_irq | SPSR_und |

▤ = banked register

ARM 레지스터

• CPSR 레지스터 구조

• N : 음수 플래그

• Z : 제로 플래그

• C : 캐리 플래그

• V : 오버플로우 플래그

• Q : 포화 플래그

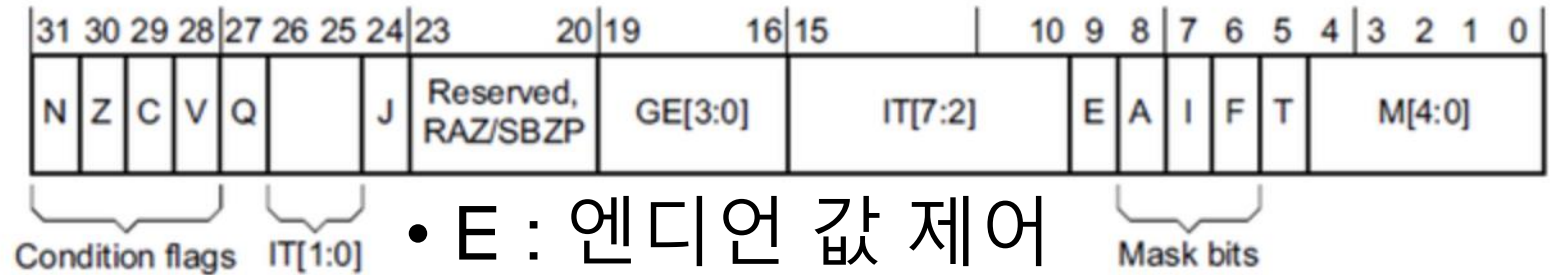
• J : 자바 실행 가능 여부 표시

• GE : general-purpose register Extend 필드

• M4~M0 : 현재 프로세서 모드 표시

• Res : 확장을 위해 예약된 비트

• IT : If-Then 문의 조건 코드를 저장하는 데 사용



• E : 엔디언 값 제어

• A : Asynchronous abort 비활성

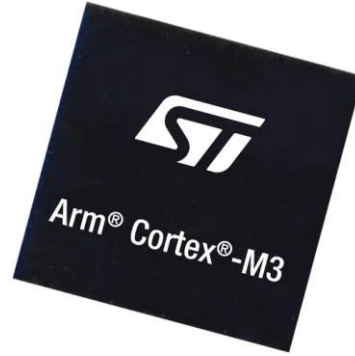
• I : IRQ 비활성

• F : FIQ 비활성

• T : Thumb or ARM 상태 표시

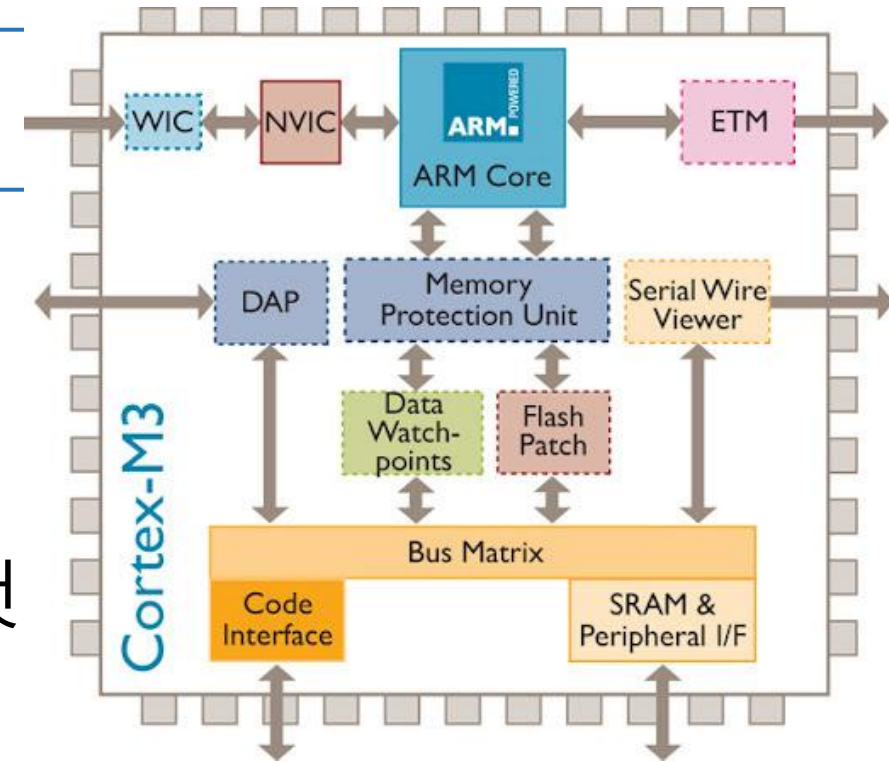
Cortex-M3

- 32 bit ARM 프로세서
- ARMv7-M 아키텍처 기반
- Harvard architecture
 - 프로그램 코드와 데이터를 별도의 메모리에 저장하는 구조
- Thumb-2 명령어만 사용 가능
 - Thumb-2 : ARM과 기존의 Thumb을 섞어서 쓸 수 있는 명령 체계
- Pipeline Branch Speculation
 - Fetch - Decode - Execute + 분기 예측(Branch Speculation)을 수행



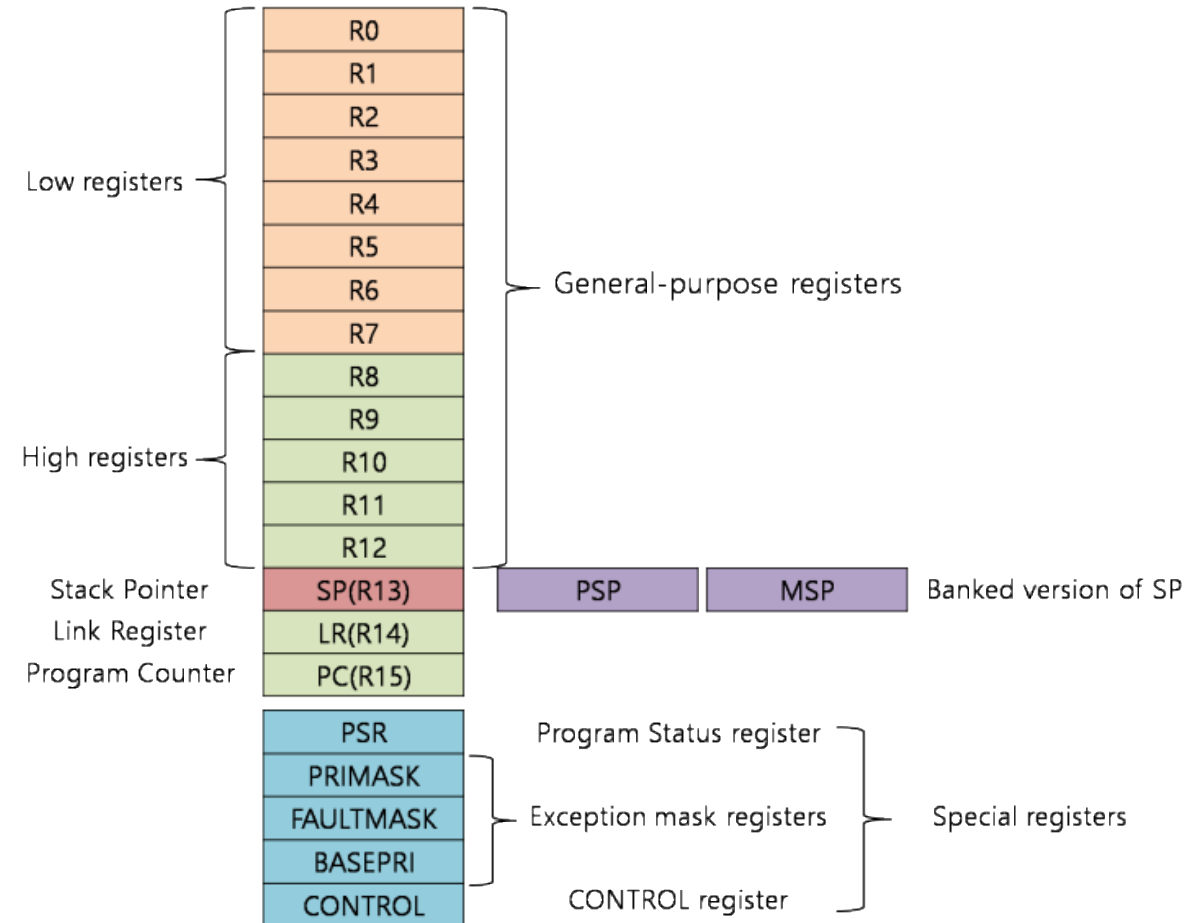
Cortex-M3 아키텍처

- ETM (Embedded Trace Macrocell)
 - Trace 정보를 수집하는 디버깅 방식
- FPB (Flash Patch and Breakpoint)
 - Flash Patch - 바이너리 코드를 수정하는 것
- DWT (Data Watchpoint and Trace)
 - 프로세서가 접근할때 선택적 데이터 트레이스를 가능하게 해주는 컴포넌트
 - 프로파일링 트레이스 : 다양한 동작에서 CPU가 사용하는 클럭 수 포착
 - 이벤트 트레이스 : 처리한 예외의 이력 및 기간 제공
- DAP (Debug Access Port)
 - 외부의 JTAG 디버거와의 인터페이스를 담당하는 컴포넌트



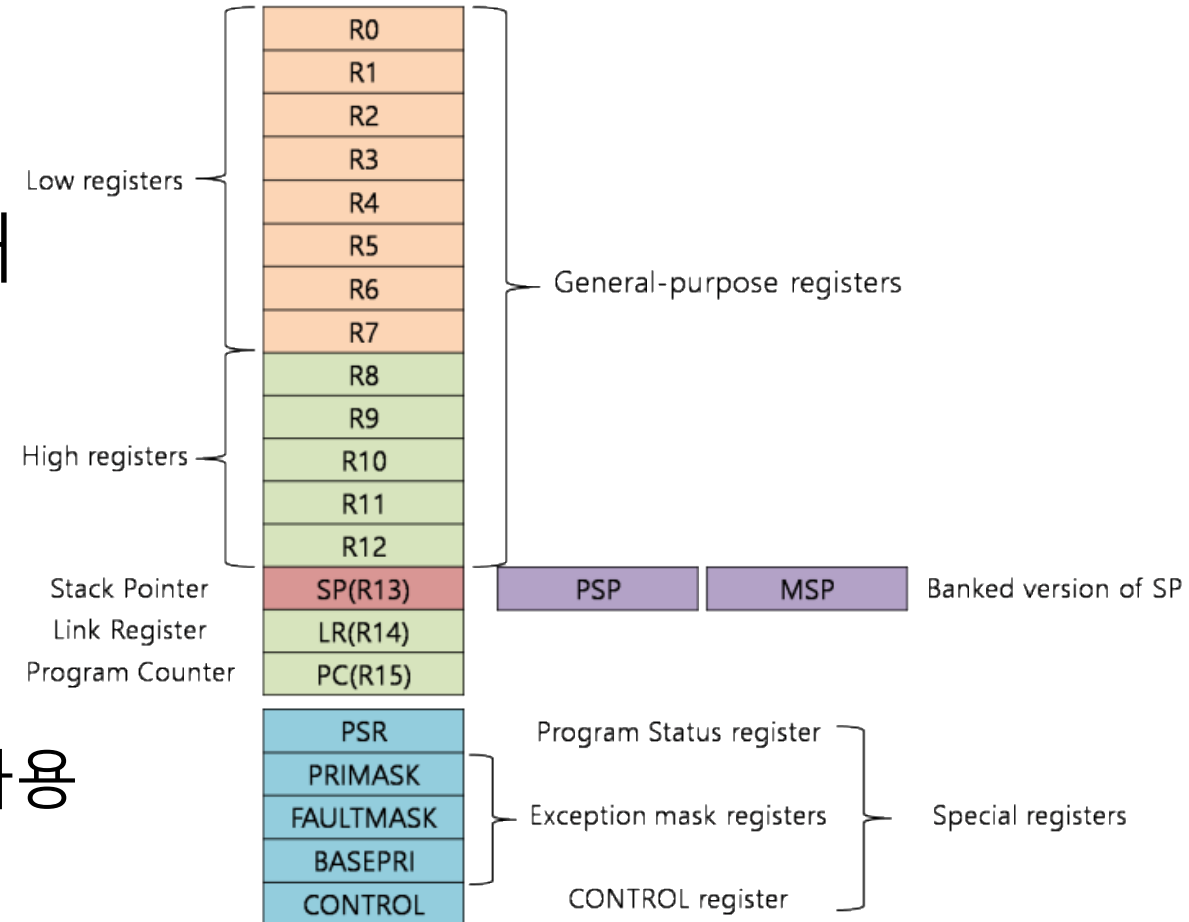
Cortex-M3 레지스터

- R0~R12 : 범용 레지스터
 - R0~R7 : 모든 명령어에서 접근 가능
 - R8~R12 : 32bit 명령어만 접근 가능
- R13 : Stack Pointer
- R14 : Link Register
- R15 : Program Counter
- PSR : Program Status Register
 - APSR, IPSR, EPSR로 구성됨
 - IEPSR(EPSR & IPSR) : read only
 - IAPSR(APSR & IPSR) : read-write
 - EAPSR(APSR & EPSR) : read-write



Cortex-M3 레지스터

- APSR : 프로그램 상태 레지스터
- IPSR : 인터럽트 상태 레지스터
- EPSR : 확장 프로그램 상태 레지스터
- PRIMASK : 프로그램 예외 마스크
- FAULTMASK : 예외 마스크
- BASEPRI : 베이스 우선 순위
- CONTROL : 컨트롤 레지스터
 - 스레드 모드에서 MSP와 PSP 선택에 사용
- MSP : 메인 스택 포인터
- PSP : 프로세스 스택 포인터



Q & A