02

ARM PROCESSOR FUNDAMENTALS

한동대학교

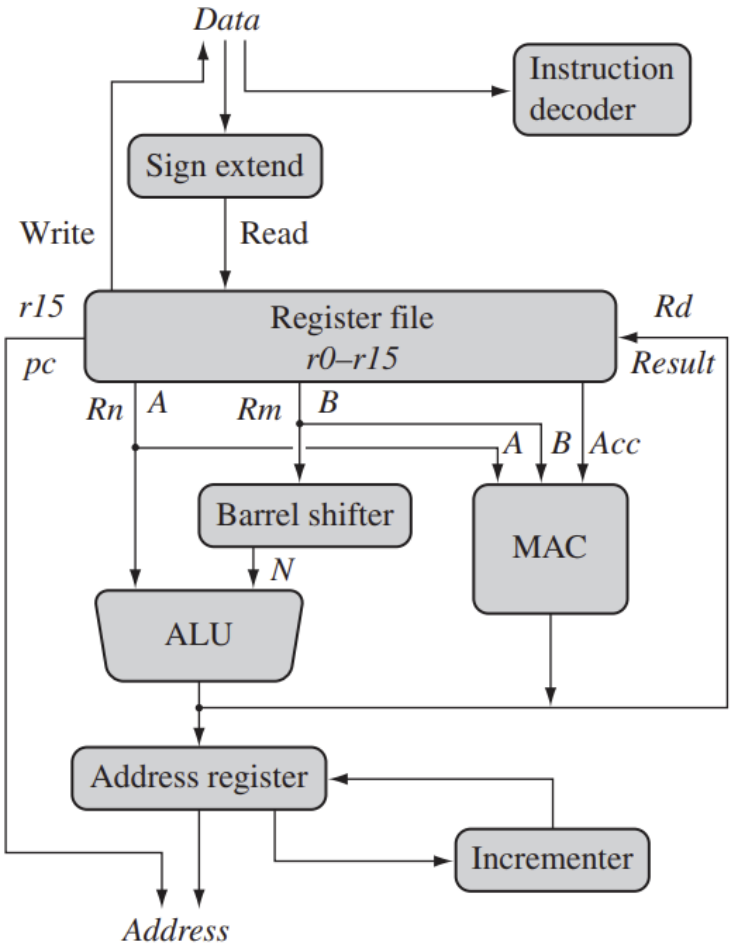
마이크로프로
세서응용

Agenda

- **Registers and Current Program Status Register**
- **Pipeline**
- Exception and Interrupt
- Core Extensions
- Architecture Revisions and ARM Family

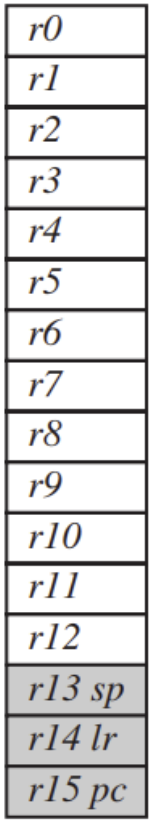
ARM core Dataflow model

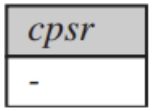
- Von neuman architecture: data item and instruction share the same bus
- RISC(Load-store architecture) has only two instruction types transferring data in and out of processor: load, store
- Data items are placed in register file – a storage bank
- Source registers: Rn and Rm
- Destination register : Rd
- Rm can be preprocessored by Barrel shifter before it enters ALU



ARM registers

- 32-bit sized 16 General purpose Registers available in user mode
  - *rx* (*x* : 0 ~ 15): stores data item or address
- Special purpose registers
  - r13:stack pointer (sp) stores the head of the stack
  - r14: link register (lr) where core puts the return address
  - r15: program counter (pc) contains the address of the next instruction

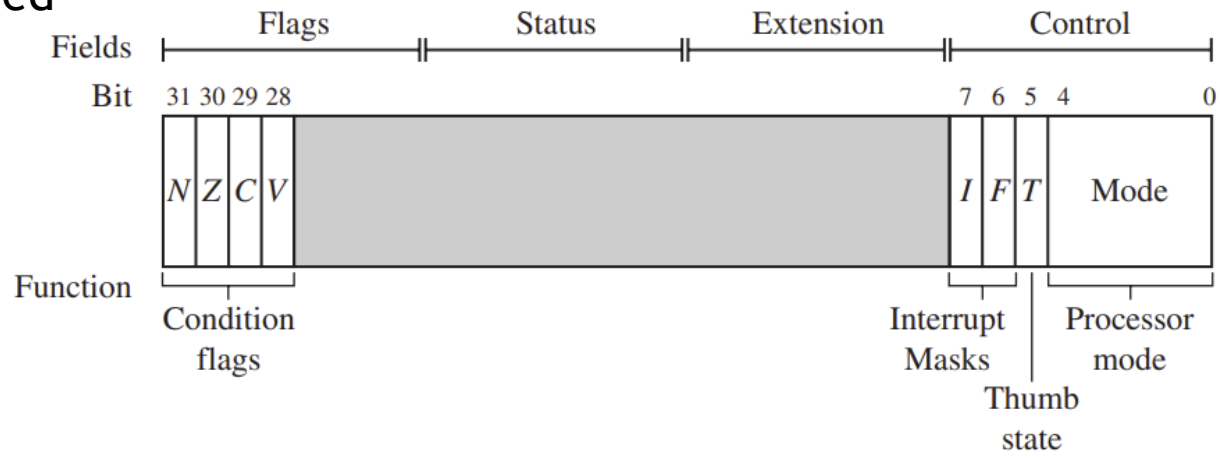




cpsr: Current Programming Status Register

- CPSR : used to monitor and control internal operations
- 32-bit register
- 4 fields (each field: 8 bits length)
  - Flags: conditions (NZCV, J flag can be included for a Jazelle-enabled processor)
  - Control: processor mode, state, interrupt mask bits
  - Status
  - Extension

reserved

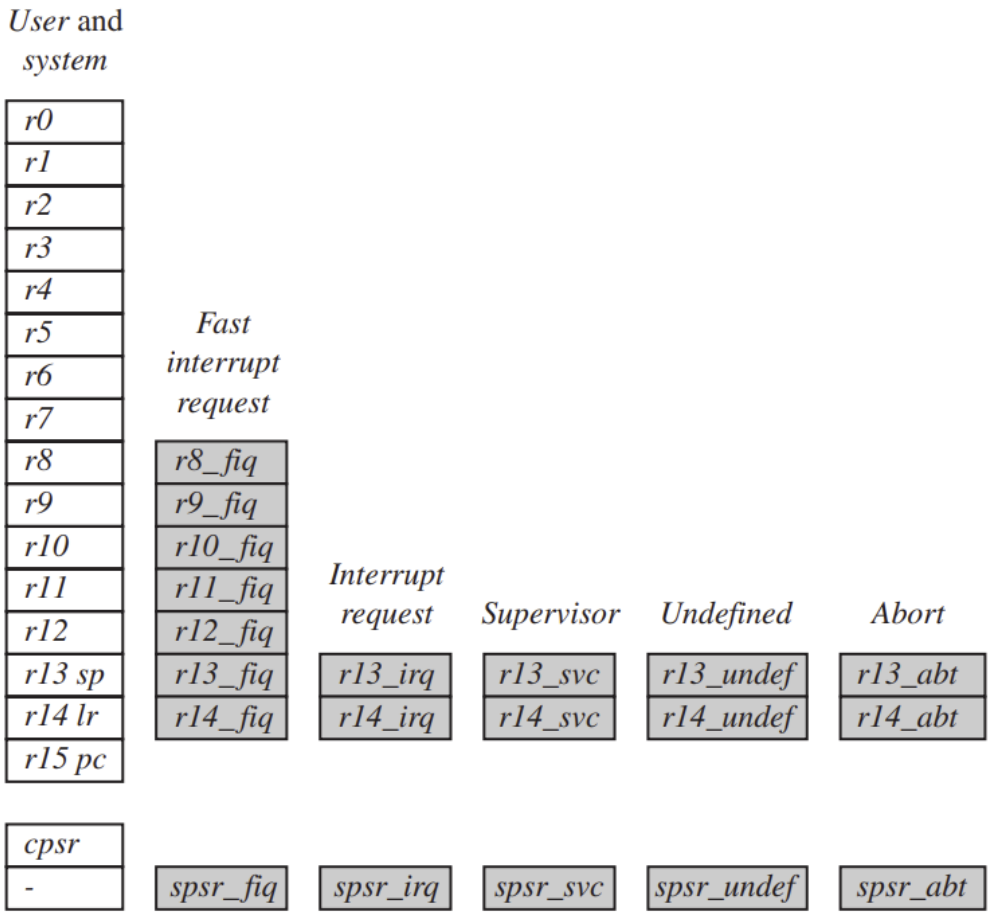


Processor mode

- Processor mode determines the active registers and access rights to cpsr
  - Privileged mode: full read-write access to cpsr
  - Non-privileged mode: only allow read access to control field in cpsr, but read-write access to condition flags
- 7 processor modes : 6 privileged + 1 nonprivileged
  1. Abort (failure to access memory)
  2. Fast interrupt request (interrupt level high)
  3. Interrupt request (interrupt level low)
  4. Supervisor: after reset and OS kernel operation mode
  5. System: special user mode for full read-write to cpsr
  6. Undefined: encounters undefined instruction
  7. User : runs application program

Banked registers

- Of All 37 registers 20 shaded (banked) registers are hidden
- All processor mode except system mode has a set of associated banked registers that are subset of main 16 registers
- A baked register maps one-to-one onto a user mode register



Processor mode change

- processor mode changes by writing mode in cpsr in privileged mode or by hardware to respond interrupt or exception
- Interrupts and Exceptions causing the mode chage
  - Interrupt request, fast interrupt request, software interrupt
  - data abort, prefetch abort, undefined instruction
- CPSR is copied to SPSR (Saved PSR) when interrupt or exception occurs.(used for restoration of CPSR)

Mode	Abbreviation	Privileged	Mode[4:0]
Abort	abt	yes	10111
Fast interrupt request	fiq	yes	10001
Interrupt request	irq	yes	10010
Supervisor	svc	yes	10011
System	sys	yes	11111
Undefined	und	yes	11011
User	usr	no	10000

State of Core

- State determines which instruction set is being executed
  - ARM state: when both T and J = 0, Only ARM instruction set is active
  - Thumb state: when T bit = 1, Only Thumb instruction set is active
  - Jazelle state : when J bit = 1, Only Jazelle instruction set is active
- State change: by special branch instruction

[ARM and Thumb instruction set features]

	ARM (*cpsr* $T = 0$)	Thumb (*cpsr* $T = 1$)
Instruction size	32-bit	16-bit
Core instructions	58	30
Conditional execution[a]	most	only branch instructions
Data processing instructions	access to barrel shifter and ALU	separate barrel shifter and ALU instructions
Program status register	read-write in privileged mode	no direct access
Register usage	15 general-purpose registers +*pc*	8 general-purpose registers +7 high registers +*pc*

Interrupt Mask

- IM used to stop Specific interrupt requests
- Two interrupt request levels
  - Interrupt request (IQ) → if Flag I of cpsr
  - Fast interrupt request (FIQ) → Flag F of cpsr
- If IQ or FIQ bit is set (1), the corresponding level interrupt is disabled
  - If FIQ is set (= 1), the fast interrupt is disabled

Condition Flags

Flag	Flag name	Set when
Q	Saturation	the result causes an overflow and/or saturation
V	oVerflow	the result causes a signed overflow
C	Carry	the result causes an unsigned carry
Z	Zero	the result is zero, frequently used to indicate equality
N	Negative	bit 31 of the result is a binary 1

- Notation of cpsr: when bit is binary 1 => capital letter, otherwise use lower case letter
  - Example: cpsr = nzCvqjiFt_SVC: Carry set, IRQ are enabled FIQ are disabled, in ARM state, Supervisor mode

Conditional Execution

- Most ARM instructions can be executed on the value of condition flags (conditional execution)

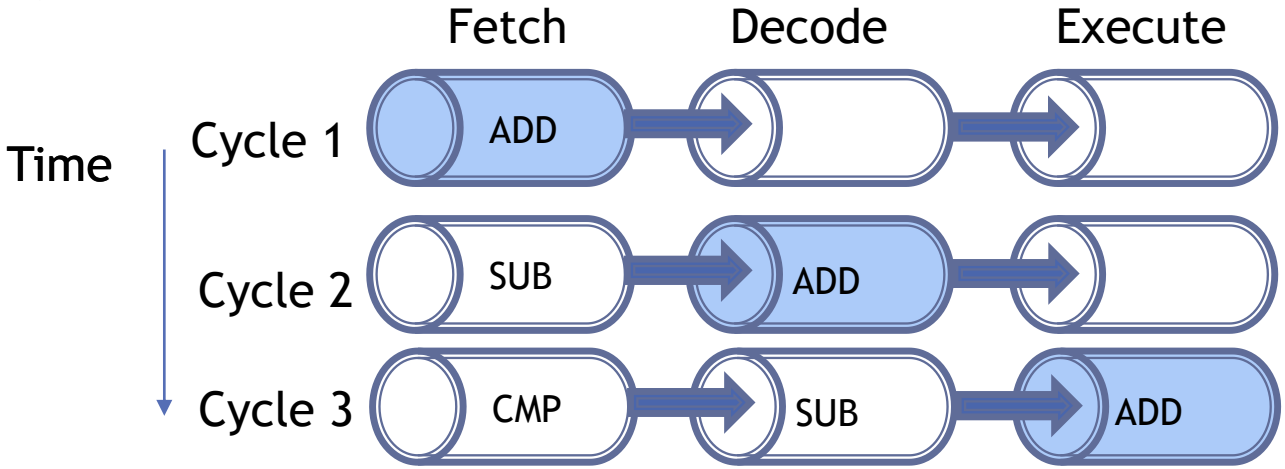
Mnemonic	Name	Condition flags
EQ	equal	*Z*
NE	not equal	*z*
CS HS	carry set/unsigned higher or same	*C*
CC LO	carry clear/unsigned lower	*c*
MI	minus/negative	*N*
PL	plus/positive or zero	*n*
VS	overflow	*V*
VC	no overflow	*v*
HI	unsigned higher	*zC*
LS	unsigned lower or same	*Z* or *c*
GE	signed greater than or equal	*NV* or *nv*
LT	signed less than	*Nv* or *nV*
GT	signed greater than	*NzV* or *nzv*
LE	signed less than or equal	*Z* or *Nv* or *nV*
AL	always (unconditional)	ignored

Agenda


Registers and Current Program Status Register
Pipeline
Exception and Interrupt
Core Extensions
Architecture Revisions and ARM Family

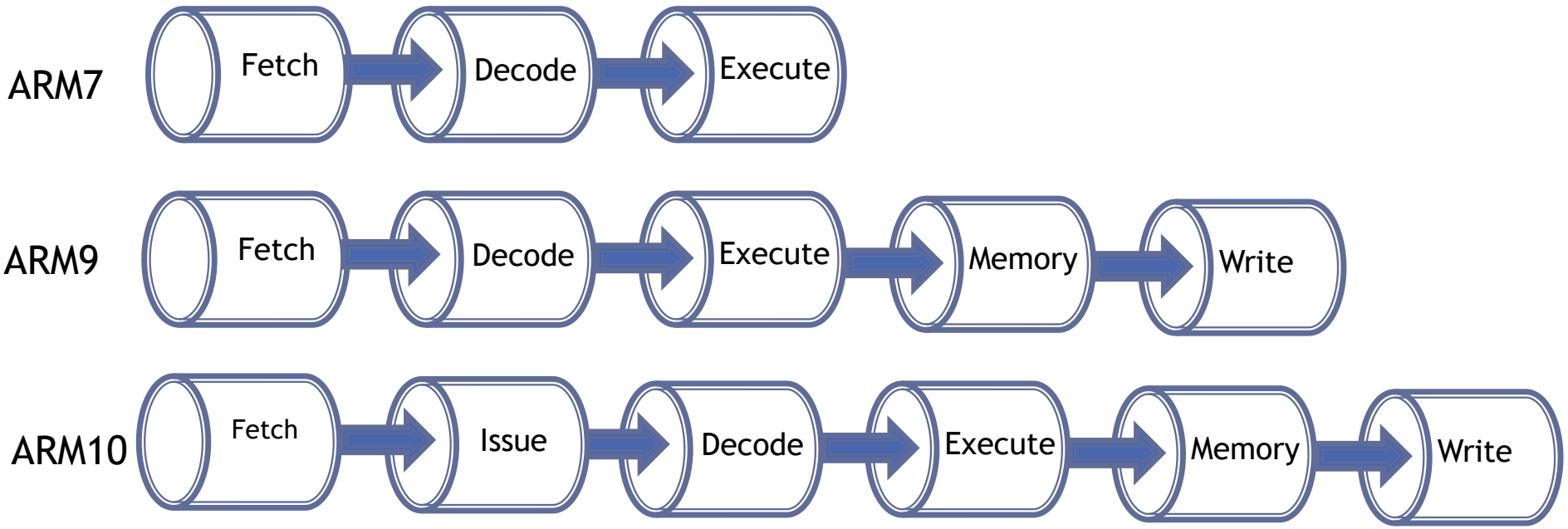
ARM 7 pipeline

- In RISC processor, Pipeline speed up execution by fetching next instruction while other instructions are decoded and executed
- 3 stage pipeline
  - Fetch : load an instruction from memory
  - Decode: identifies the instruction to be executed
  - Execute: processes the instruction and writes the results back to a register

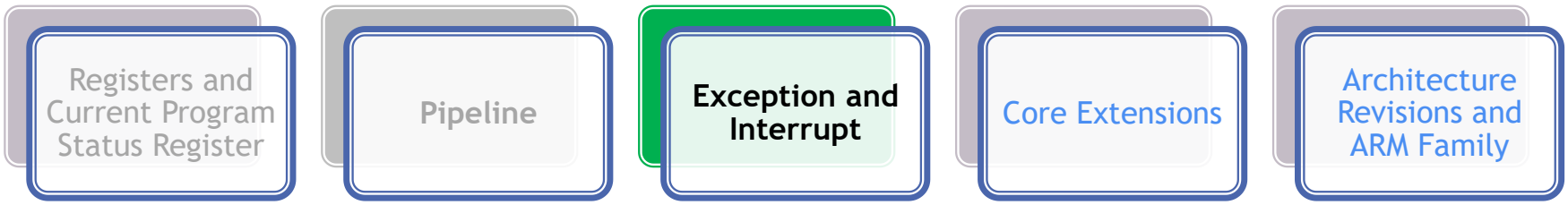


Pipeline for each ARM family

- Different ARM family uses different pipeline

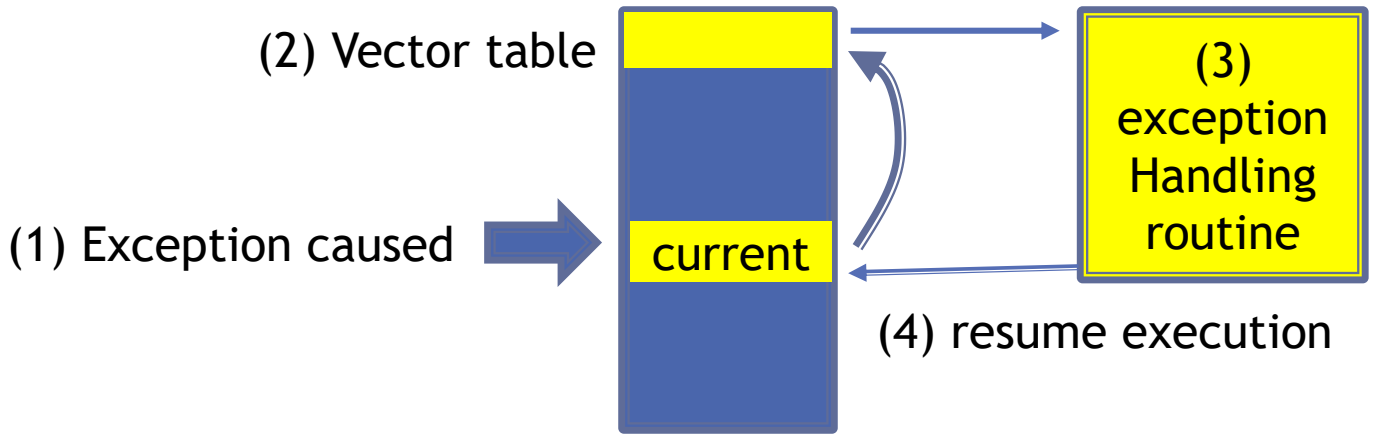


Agenda


Registers and Current Program Status Register
Pipeline
Exception and Interrupt
Core Extensions
Architecture Revisions and ARM Family

Exception and Vector Table

- When an exception or interrupt occurs,
  - the processor suspends normal execution and
  - starts loading instructions from the **exception vector table**
- Each vector table entry contains a form of **branch instruction** pointing to the start of a handler of the interrupt or exception
- address for the vector table: 0x00000000 or 0xffff0000



Exception Vector Table

- Vector Table has first (branch) instruction to routine for each exception
  - Reset vector : power is applied (initialization routine)
  - Undefined instruction vector: unknown instruction is read
  - Software interrupt vector: when execute a SWI instruction to invoke OS routine
  - Prefetch abort vector: attempt to fetch an instruction without the correct access permission
  - Data abort vector: attempt to access data memory without the access permission
  - Interrupt request vector: raised if IRQ
  - Fast interrupt request vector: raised if FIQ

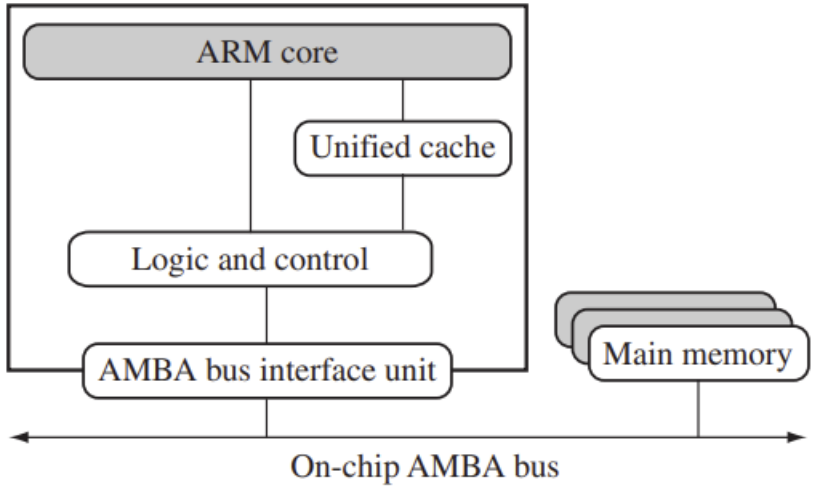
Exception/interrupt	Shorthand	Address	High address
Reset	RESET	0x00000000	0xffff0000
Undefined instruction	UNDEF	0x00000004	0xffff0004
Software interrupt	SWI	0x00000008	0xffff0008
Prefetch abort	PABT	0x0000000c	0xffff000c
Data abort	DABT	0x00000010	0xffff0010
Reserved	—	0x00000014	0xffff0014
Interrupt request	IRQ	0x00000018	0xffff0018
Fast interrupt request	FIQ	0x0000001c	0xffff001c

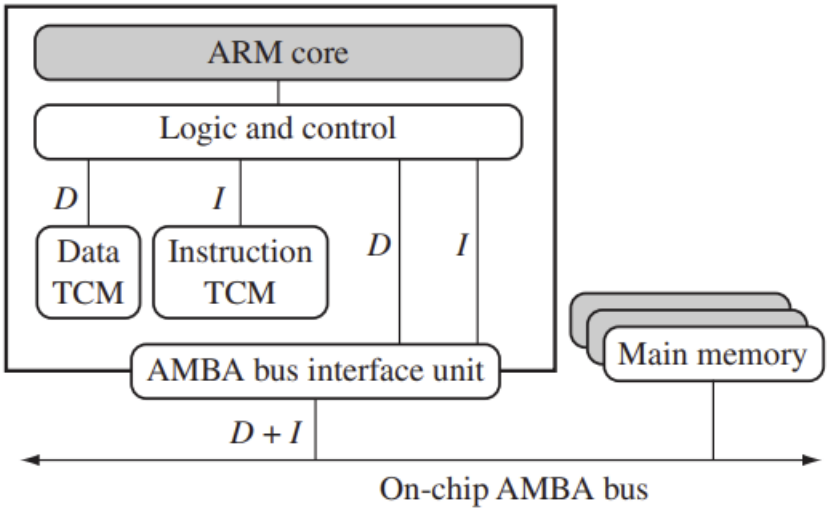
Agenda


Registers and Current Program Status Register
Pipeline
Exception and Interrupt
Core Extensions
Architecture Revisions and ARM Family

Cache and Tightly Coupled Memory

- Architecture of Cache Attached to the Von Neumann-style core: combines data and instruction into a unified cache
- Harvard Architecture with TCM and separate cache for Data and Instruction
  - TCM (Tightly Coupled Memory) is a fast SRAM located close to the core to gurantee the deterministic fetch clock cycles





Memory Management

- Memory management hardaware protects system from inappropriate hardware access.
- 3 types of memory management hardware by ARM cores
  - No extension : Nonprotected memory
  - Memory protection unit(MPU) provides limited memory protection for systems requiring memory protection but with simple memory map
  - Memory management unit (MMU) provides full protection uses a set of translation table for virtual-to-physical address map as well as access permission conrol

Agenda


Registers and Current Program Status Register
Pipeline
Exception and Interrupt
Core Extensions
Architecture Revisions and ARM Family

Coprocessors

- Coprocessor extends the processing feature of a core
- Specialized group of New instructions

Agenda


Registers and Current Program Status Register
Pipeline
Exception and Interrupt
Core Extensions
Architecture Revisions and ARM Family

Architecture Revision and family

- An ARM ISA (Instruction Set Architecture) has many processor implementations
  - ISA evolved to keep up with the demands of embedded market
  - Source code Backward compatibility
- With each family a number of variations of memory management, cache, TCM
- ARM nomenclature

  ARM{x}{y}{z}{T}{D}{M}{I}{E}{J}{F}{-S}
  - X : family (7,8,9,10,11 – ARM8 family was soon superseded)
  - Y: memory management/ protection unit
  - Z: cache
  - T: Thumb 16-bit decoder
  - D: JTAG debug
  - M: fast multiplier
  - I : Embedded ICEmacroCell
  - E: Enhanced instructions (assume TDMI)
  - J: Jzaelle
  - F: vector floating-point unit
  - S: synthesizable version

ARM Family (1)

Revision	Example core	ISA enhancement
ARMv1	ARM1	in 1985, First ARM processor, 26-bit addressing
ARMv2	ARM2	in 1987, 32-bit multiplier, 32-bit coprocessor
ARMv2a	ARM3	On-chip cache, Atomic swap instruction
ARMv3	ARM6, ARM7DI	in 1990, 32-bit addressing, Separate cpsr and spsr, MMU support, New modes – undefined instruction
ARMv3M	ARM7M	Signed and unsigned long multiply
ARMv4	strongARM	in 1995, Load-store instructions for signed and unsigned halfword/bytes, New mode –system
ARMv4T	ARM7TDMI, ARM9T	in 1994, introduced **T**humb instruction set, embedded **D**ebugger, single cycle **M**ultiplier, **I**CE(In-Circuit-Emulation) for hardware emulation in SW
ARMv5TE	ARM9E, ARM10E	in 1997, Extra instructions added for changing between ARM and Thumb, Enhanced multiply instructions, Enhanced DSP instructions
ARMv5TEJ	ARM7EJ	Java accelerator
ARMv6	ARM11	in 2002, Introduced SIMD instructions for better multimedia processing, improved memory handling through Physical Address Extension

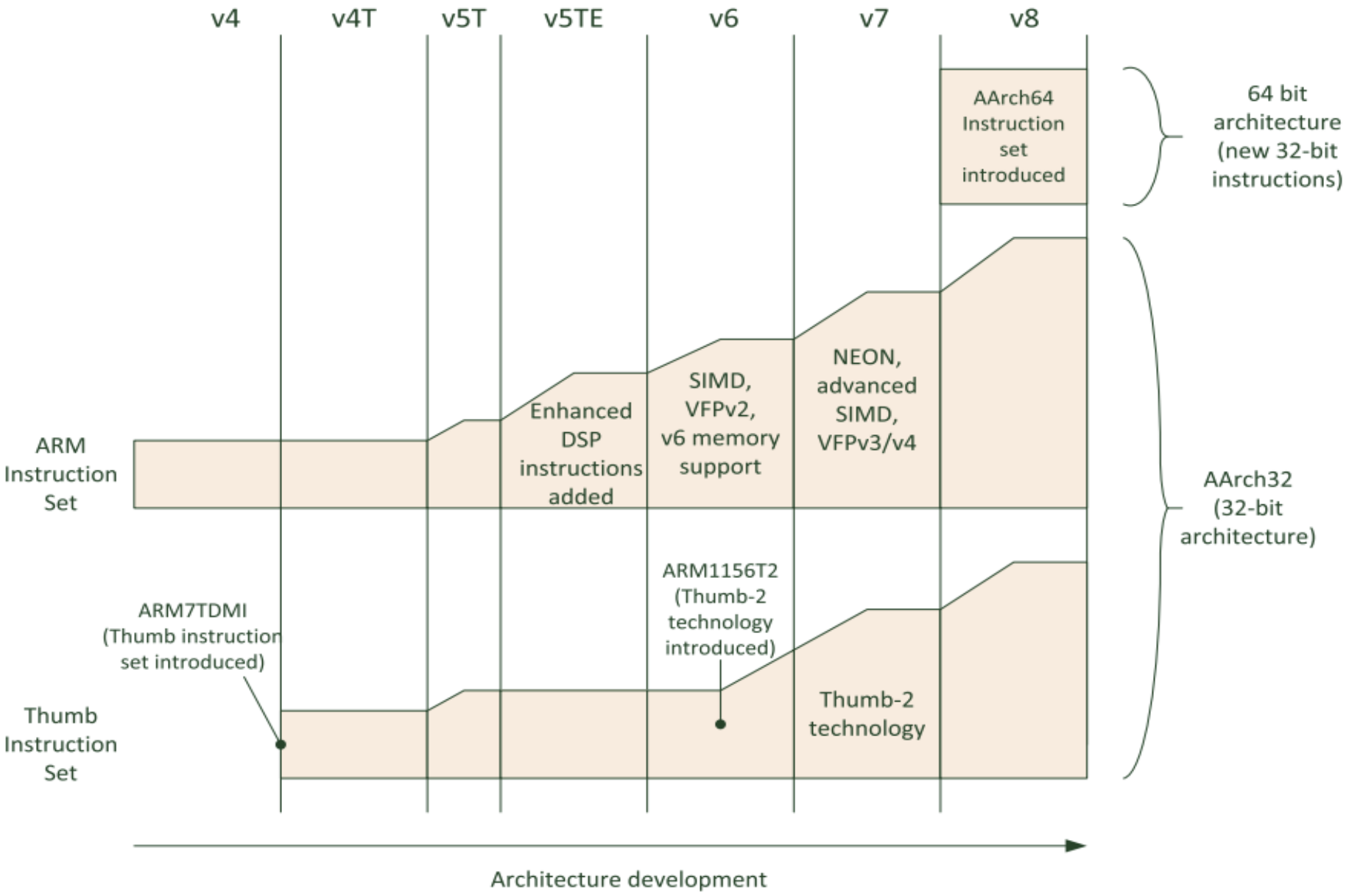
ARM Family (2)

Revision	Example core	ISA enhancement
ARMv6-M	Cortex-M0, M0+	ultra low-power and cost-effective processor for low-cost devices
ARMv7	Cortex-A5, A7, A8, A9, A12, A15, A17 Cortex-R4, R5, R7, R8 Cortex-M3, M4, M7	in 2005, Three instruction profile Cortex-A,-R,-M, Enhanced Instruction Set: introduction of Thumb-2 to support a mix of 32-bit and 16-bits instruction set, big.LITTLE architecture pairs big (Cortex-A15) and little (Cortex-A7) achieving improvement in energy efficiency and performance, Hardware Virtualization support enabling more efficient use of HW resource and better isolation and security between different OS running on the same HW, Neon Advanced 128bit SIMD and VFPv3 (Vector Floating Point version 3), Physical Address Extensions (LPAE) allowing 1TB (40bits address) space, Security Extension (TrustZone)
ARMv8	Cortex-A32, A35, A53, A55,A57, A72, A73, A75, A76, A77,A78 Cortex-R52 Cortex-M23, M33	in 2011, ARM announced 64-bit support (AArch64) while maintaining compatibility with AArch32, Enhanced Memory Model especially for multi-core and multi-threaded, Advanced SIMD extensions capable for both AArch64 and AArch32 states, Improved Cryptography Features accelerating encryption/decryption, Enhanced Virtualization for multiple OS and isolation on the same HW
ARMv9	Cortex-A710, A-510, Cortex-X2 (significant performance for mobile and labtop)	introduced in March 2021 CCA(Confidential Compute Architecture) SVE2(Scalable Vector Extension 2) AI and ML boosting through dedicated AI instruction

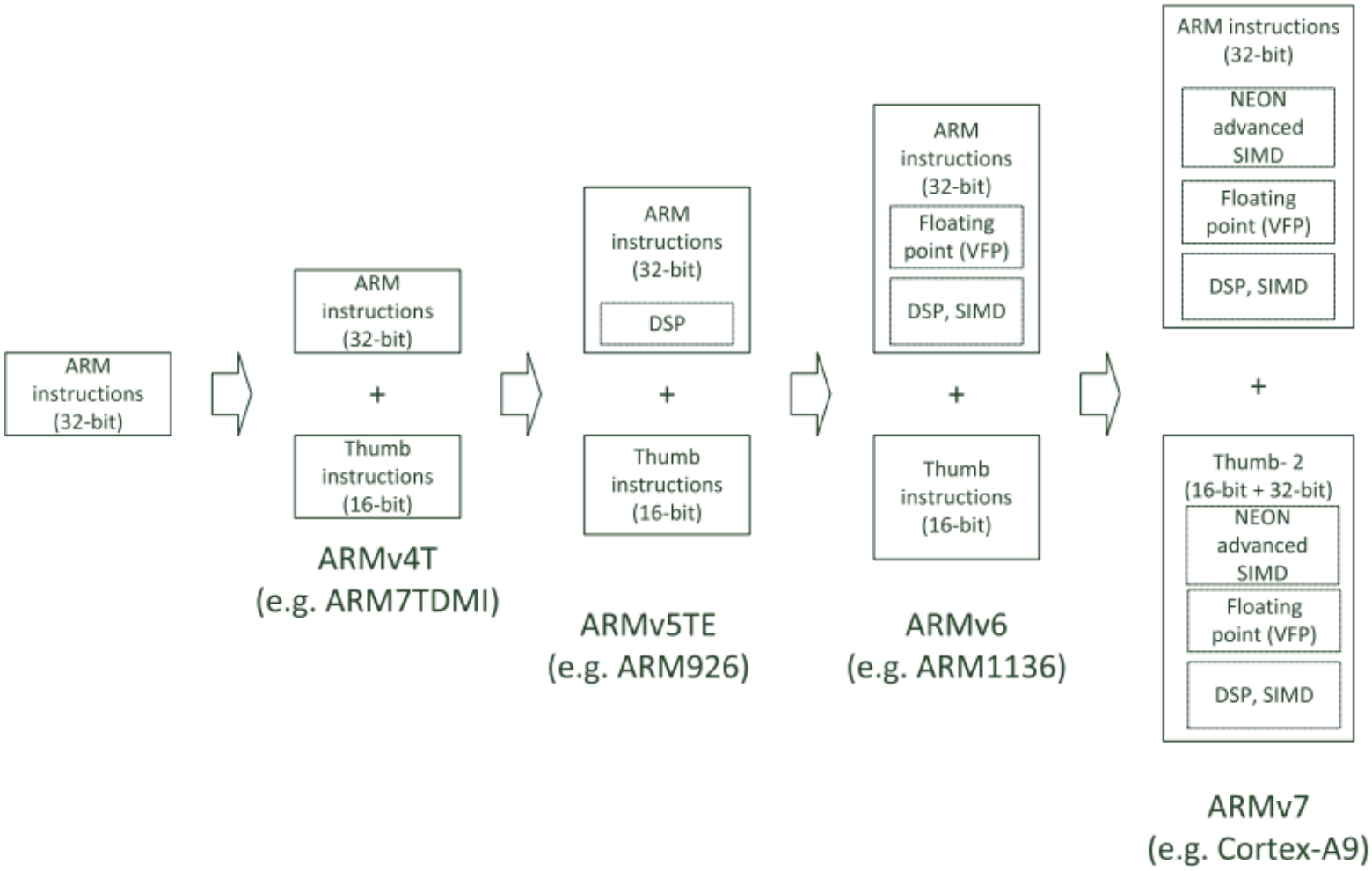
ARM ISA Evlolution: Thumb & Thumb-2

- Thumb
  - ARM7TDMI processor (in 1995) supports software supports to switch between 32-bit ARM state and 16-bit Thumb state
  - For smaller code size (30% reduction)
  - a subset of ARM instruction in 16-bit format
  - Restriction: register choices, available addressing mode, reduced range of immediate value for data and address
- Thumb-2
  - ARM1156T-2 processor (in 2003) supports method to combine 16-bit and 32-bit instruction sets in one operation state
  - both 16-bit compact code size and 32-bit high performance
  - most of the operations previously possible in ARM instruction. But, different instruction encoding to ARM

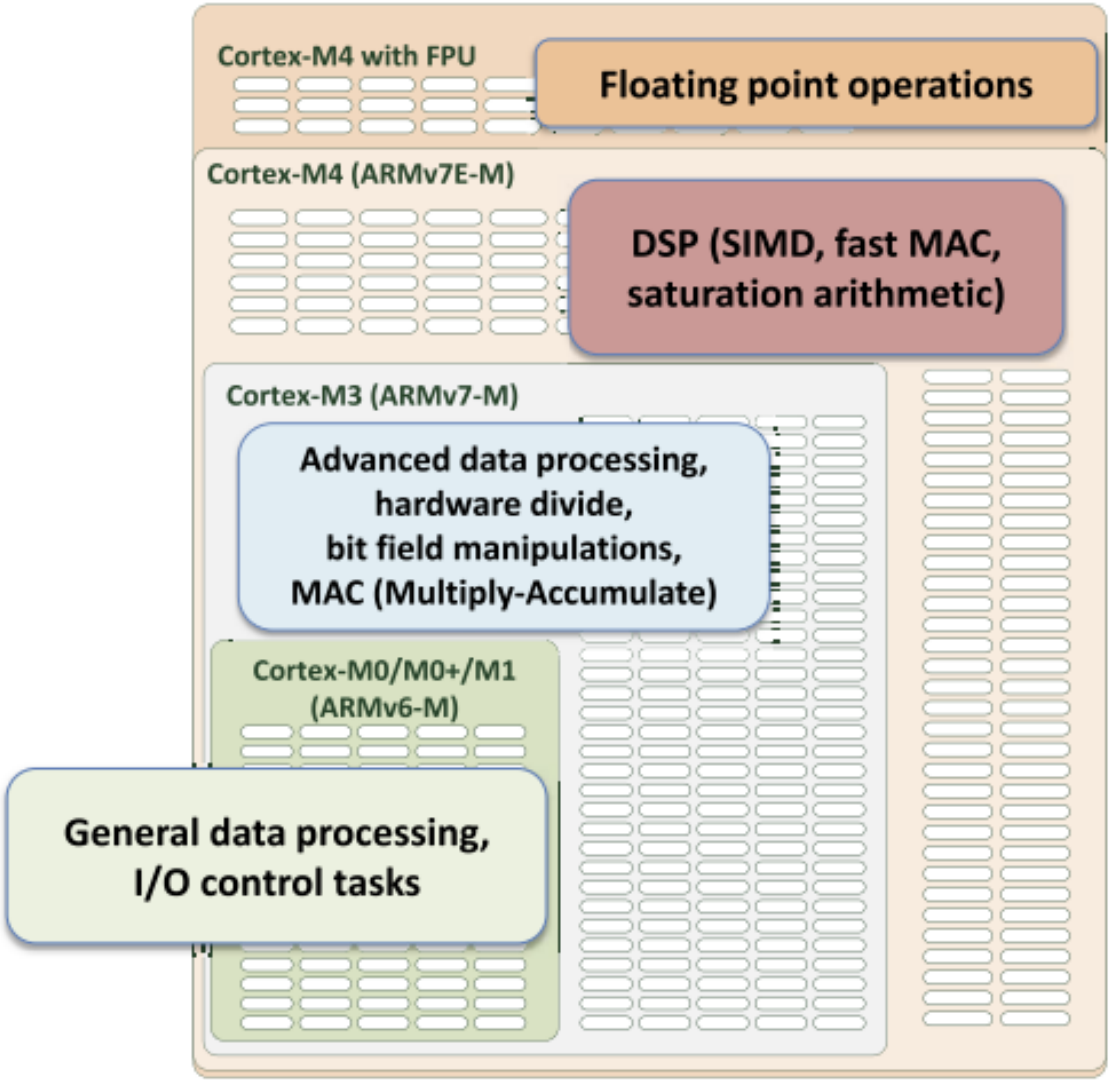
Architecture Development



Evolution of the ARM ISA



ISA comparison ARM Cortex-M processor



ARM family attribution comparison

	ARM7	ARM9	ARM10	ARM11
Pipeline depth	three-stage	five-stage	six-stage	eight-stage
Typical MHz	80	150	260	335
mW/MHz[a]	0.06 mW/MHz	0.19 mW/MHz (+ cache)	0.5 mW/MHz (+ cache)	0.4 mW/MHz (+ cache)
MIPS[b]/MHz	0.97	1.1	1.3	1.2
Architecture	Von Neumann	Harvard	Harvard	Harvard
Multiplier	8 × 32	8 × 32	16 × 32	16 × 32

[a] Watts/MHz on the same 0.13 micron process.

[b] MIPS are Dhrystone VAX MIPS.

ARM7 Family

- ARM7TDMI : the first of a new range of processors introduced in 1995 by ARM.
  - A very good performance-to-power ratio
  - The first core that includes Thumb instruction set, fast multiplier, embeddedICE debug technology
  - This core has been licensed by many semiconductor companies.
- ARM7TDMI-S :same operating characteristics as ARM7TDMI but synthesizable
- ARM720T : most flexible member of ARM7 (includes MMU and 8K cache ➔ Linux and MS embedded operating systems)
- ARM7EJ-S : 5-stage pipeline and execute ARMv5TEJ instruction. Only one ARM version providing both Java acceleration and enhanced instructions but without memory protection

ARM9 Family

- 5-stage pipeline, Havard architecture (separate D+I cache)
- v4T architecture
  - ARM920T: first ARM9 family
  - ARM940T: smaller D+I cache and MPU
- ARM9E-S : synthesizable version with E extension
  - v4TE architecture : ARM946E-S and ARM966E-S
  - ARM946E-S : TCM and MPU. Size of TCM and caches are configurable for embedded application for deterministic real-time response.
- ARM926EJ-S : synthesizable core in announced in 2000,
  - designed for portable Java enabled devices such as 3G phone
  - MMU, configurable TCM, and D+I caches

ARM10 , ARM11 Family

- ARM10 Family
  - For performance it extends ARM9 pipeline to six stages
  - Optional vector floating-point (VFP) unit
  - ARM1020E :32K D+I cache, dual 64-bit bus interface for increased performance
- ARM11 Family
  - significant performance enhancements and power-efficiency
  - in 2003, ARM1136J-S was the first ARMv6 implementation
  - 8 pipeline stages with separate load-store and arithmetic pipelines
  - SIMD extension for media processing, especially to increase video processing performance

ARMv7

- Thumb-2 instruction set architecture for the mix of 16-bits and 32-bits instructions
- Three processor Profiles targeting different domain
  - Cortex-A profile : Application
    - ✓For High performance applications such as smartphone and tablets
  - Cortex-R profile : Real Time
    - ✓For applications where reliable and quick, deterministic response time is critical. Embedded systems for Automotive, industrial, telecommunication applications
  - Cortex-M profile : Microcontroller
    - ✓For applications requiring cost-effective solution and energy efficiency and modest computing capabilities
    - ✓IoT device, sensor modules, low-power embedded systems

ARMv8

- ARMv8-A represents a fundamental change to ARM architecture :64bot architecture :
  - 64-bit architecture : AArch64
  - big.LITTLE configuration (cortex-A57, Cortex-A53 cores)
    - ✓Apple A7 used in iPhone 5S
    - ✓Samsung Exynos5433 used in Galaxy Note 4