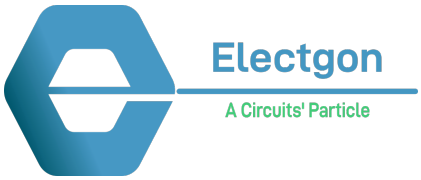


ARM Processors - ARCHITECTURES, CORES AND FAMILIES

ELECTGON
www.electgon.com
ma_ext@gmx.net

09.04.2018

Contents

Abstract

Many processors are now available on the market provided by leaders of chip foundries like Analog Devices, Intel, ARM, power PC and Atmel. Choosing which processor to work with will not be based on difference in structure, otherwise it will be based mainly on best experience and available development kit with reasonable pricing. Due to that ARM processor is commonly used in broad applications in embedded systems domain as well as its development environment is available with good pricing. This document is providing first step to know about ARM processors history and difference between its major architectures and families.

1 Introduction

Advanced RISC Machine (ARM) is a processor architecture that is used in modern CPUs. The company itself was founded in 1990 as spin out of collaboration between Acron and Apple computers that were using this architecture in their computers.

ARM architecture first saw light in 1985 after a project in Berkeley university called VLSI Project came up with the RISC architecture between 1980 and 1984. VLSI design group in Acron started design of ARM1 based on this RISC philosophy. Between 1985 and 1990, many versions of ARM architecture were developed. When the company was founded in 1990 it released ARMv6, now according to ARM website in 2015 latest architecture is ARMv8.

If you are familiar with processor architectures, you would recognize that an architecture means Instruction Set & hardware model. What is meant by hardware model is the different techniques that are used to implement the processor, for example Von-Neumann or Harvard. 8 or 16 or 32 bits as well as register set and operation modes. So we have different techniques for the hardware implementation as well as Instruction set that is developed by time, in addition to dependency of instructions set on data size.. all these factors lead to production of different versions and models of ARM processors. Not only but also the application for which this processor is used may affect in the design of the processor. ARM is organizing this versioning of processor by indicating an ARM architecture which is evolved mainly according to instructions set. With new instruction set, you can make configuration to hardware, this means you have different cores within this architecture. So you can say that some ARM cores are based on specific architecture version or ARM architecture is built on specific core.

To clarify it, ARM has mainly 2 instruction set, they are called ARM, Thumb. Each has its own instructions as shown in table 1. Referring to this table you can notice that Thumb as some additional instructions. Such a change make difference between processor performance.

ARM	
Mnemonic	Instruction
ADC	Add with carry
ADD	Add
AND	And
B	Branch
BIC	Bit clear
BL	Branch with link
BX	Branch and exchange
CDP	Coprocesor Data Processing
CMN	Compare Negative
CMP	Compare
EOR	Exclusive OR
LDC	Load coprocessor from Memory
LDM	Load multiple registers
LDR	Load register from memory
MCR	Move CPU register to coprocessor register
MLA	Multiply Accumulate
MOV	Move register or constant
MRC	Move from coprocessor register to CPU register
MRS	Move PSR status/flags to Register
MSR	Move register to PSR status/flags
MUL	Multiply
MVN	Move negative register
ORR	Or
RSB	Reverse Subtract
RSC	Reverse Subtract with Carry
SBC	Subtract with Carry
STC	Store coprocessor register to memory
STM	Store Multiple
STR	Store register to memory
SUB	Subtract
SWI	Software Interrupt
SWP	Swap register with memory
TEQ	Test bitwise equality
TST	Test bits

Thumb	
Mnemonic	Instruction
ADC	Add with carry
ADD	Add
AND	And
ASR	Arithmetic Shift Right
B	Unconditional branch
Bxx	Conditional branch
BIC	Bit clear
BL	Branch with link
BX	Branch and exchange
CMN	Compare Negative
CMP	Compare
EOR	Exclusive OR
LDMIA	Load multiple
LDR	Load word
LDRB	Load byte
LDRH	Load halfword
LSL	Logical Shift Left
LDSB	Load sign-extended Byte
LDSH	Load sign-extended Halfword
LSR	Logical Shift Right
MOV	Move register or constant
MUL	Multiply
MVN	Move negative register
NEG	Negate
ORR	Or
POP	Pop registers
PUSH	Push registers
POR	Rotate Right
SBC	Subtract with Carry
STMIA	Store Multiple
STR	Store word
STRB	Store byte
STRH	Store halfword
SUB	Subtract
SWI	Software Interrupt
TST	Test bits

Table 1: ARM and Thumb Instruction Set

This table is only as an example of how instruction set may differ from version to another, but Architecture versions is not changing according to change of these instructions only. There are other considerations also like Operation modes 16 or 32 or 64 addressing, …and other technical specification that I don't like to discuss here. A good reference to understand the difference between each architecture can be checked in this link:

http://www.davespace.co.uk/arm/introduction-to-arm/

2 Architecture, Core, Family

ARM and Thumb instruction sets are the oldest. You can find out what are instructions set supported in each architecture in ARM website which is depicted in figure 1

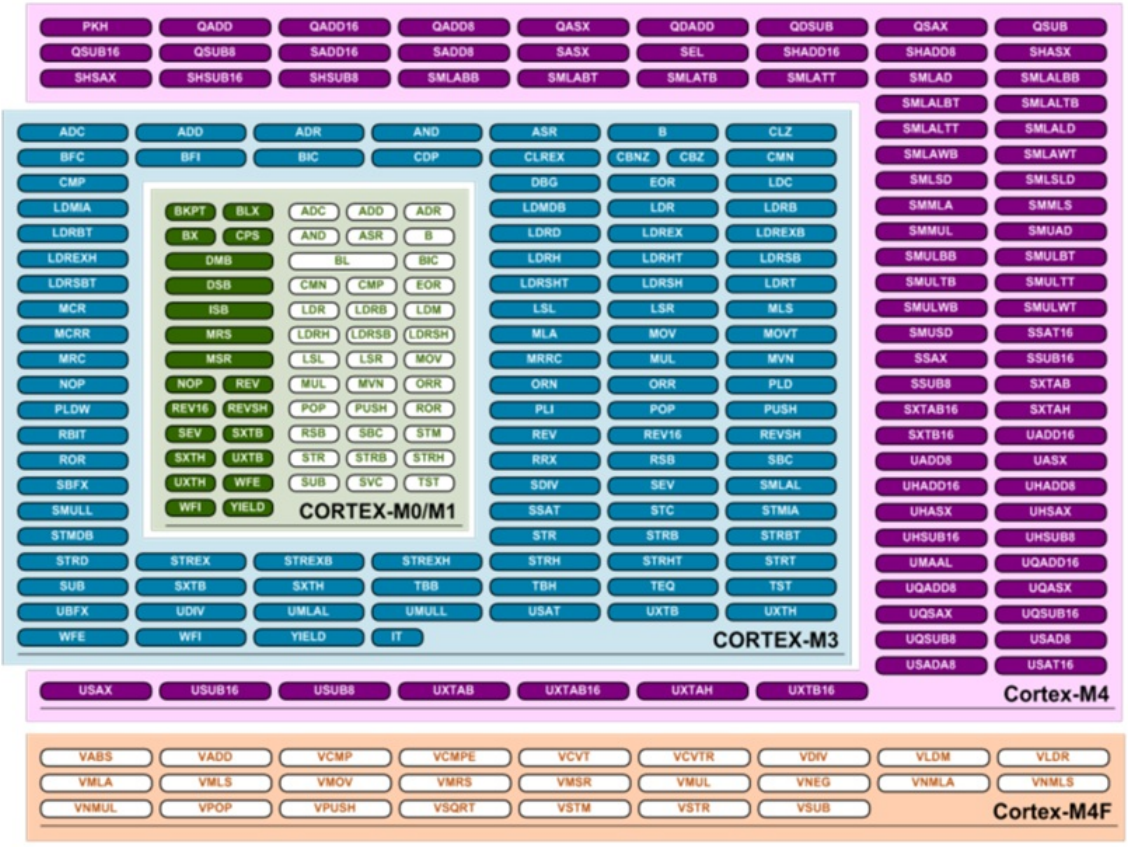


Figure 1: ARM Instructions in Different Architectures

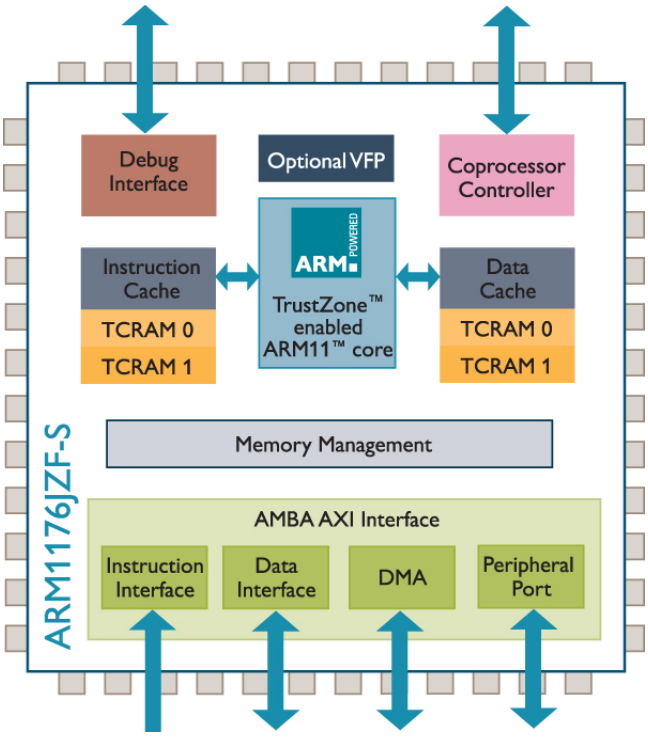
I hope it is clear now why we have different ARM architectures, remaining to mention only that any instructions added are serving specific application, for example there are instructions that are added to support floating point operations, other are added to enhance Java applications operations, others for signal processing,... for example ARMv6 is supporting new multiprocessing instructions LDREX, STREX and more than 60 SIMD instructions. Some cores has been produced based on ARMv6 architecture like what is shown in table 2.

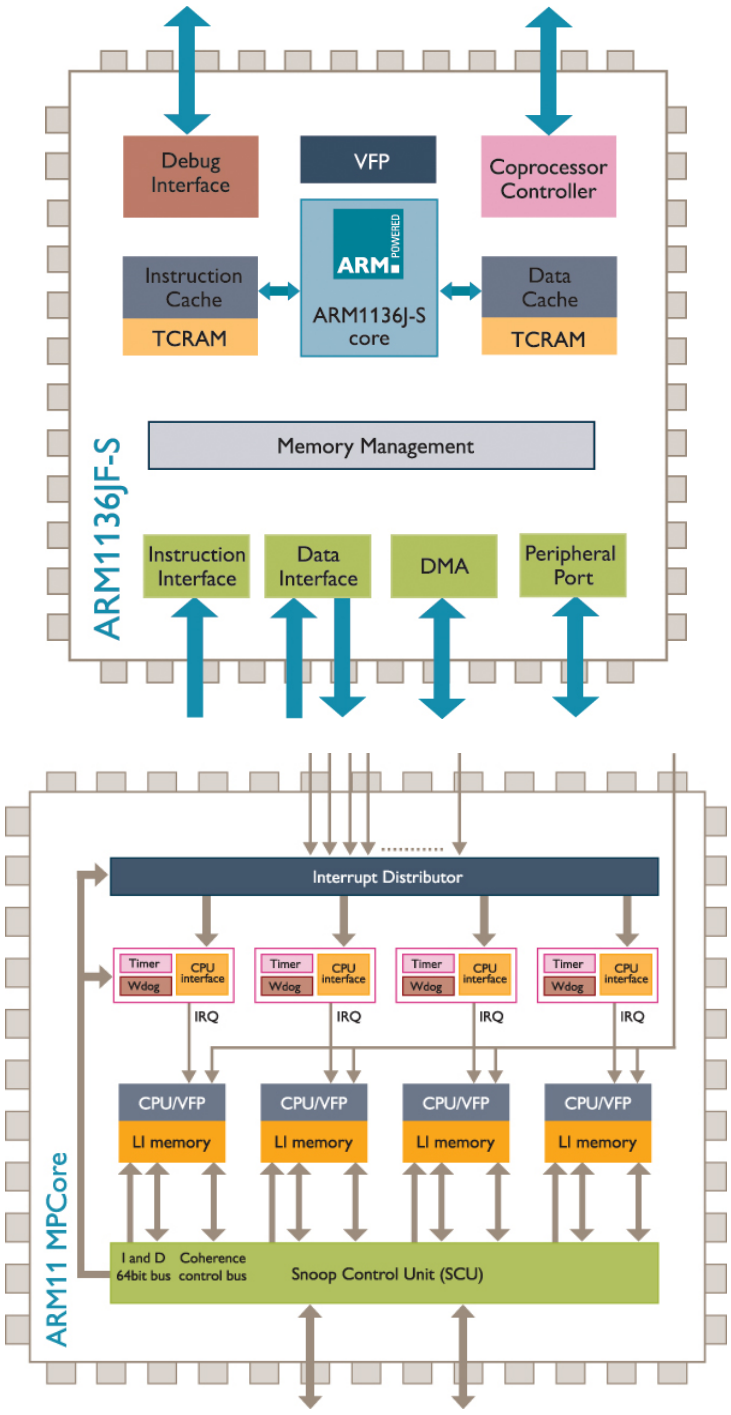
	1136EJ(F)-S	1156T2(F)-S	1176JZ(F)-S	MPCore11
Architecture	Harvard	Harvard	Harvard	Harvard
Cache	4-64K Instr 4-64K Data 8 words/line	0-64K Instr 0-64K Data 8 words/line	4-64K Instr 4-64K Data 8 words/line	16-64K Instr 16-64K Data 8 words/line
Associativity	4-way	4-way	4-way	4-way
TCM	0-64K Instr 0-64K Data	0-256K Instr 0-256K Data	0-64K Instr 0-64K Data	None
Replacement	Random Round Robin	Random Round Robin	Random Round Robin	Random Round Robin
Write Startegy	Write Through Write Back	Write Through Write Back	Write Through Write Back	Write Through Write Back
MMU/MPU	MMU	MPU	MMU	MMU
Hi Vectors	Yes	Yes	Yes	Yes
Streaming	Yes	Yes	N/A	Yes
Standby mode	Yes	Yes	Yes	Yes
Bus	AHB/APB	AXI	AXI	AXI
VFP Support	Yes	Yes	Yes	Yes

Table 2: ARMv6 Cores

So we have different cores 1136, 1156, 1176, ... that are built based on ARMv6 architecture. All these core are grouped in ARM11 family. So we have ARM11 family doesn't mean that it is using ARMv11. It is using ARMv6 but name comes from cores forming this family.

The difference between these cores that is discussed in previous table can be reflected in core layout as shown in the following figures





We have different cores then, these cores are grouped into families. Usually these families are serving specific application. For instance, according to ARM website in 2015, there are currently 4 families of ARM processors CORTEX A, CORTEX R, CORTEX M and Specialist processors. Old families like Classic Family (ARM7, ARM9, ARM11) are going to be not supported.

In order to know your processor belongs to which family, you can check ARM website:
http://www.arm.com/products/processors/index.php

Also figure 2 summarizes all available cores and which architecture it is based on

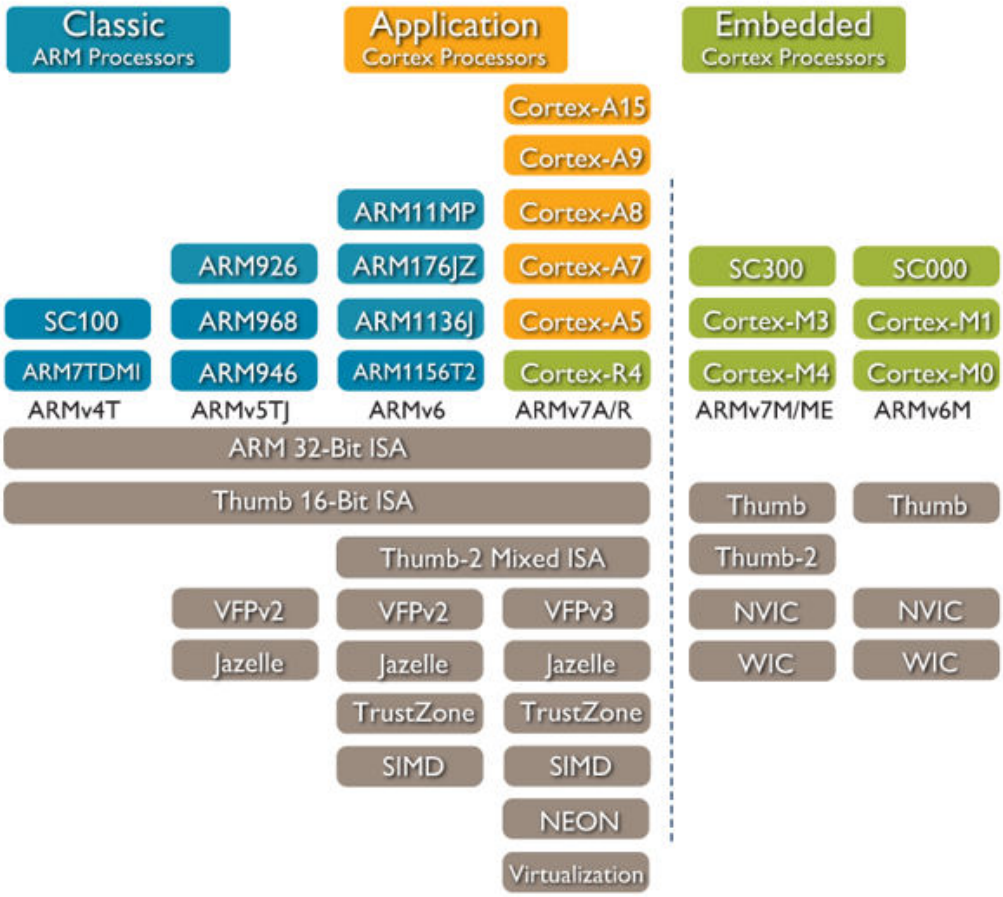


Figure 2: ARM Families

3 ARM Development Boards

ARM processors are used nowadays in 75% of embedded systems applications. Not only but also big chip manufacturers like Broadcom, Texas Instruments, Atmel, Xilinx... are using ARM processors in their chips. Individuals can also work with ARM processors, they can purchase ARM processors as a chip or they can work with an ARM-based development board. These development boards are normally provided with necessary interfaces and peripherals like USB, PS2 ,Ethernet, HDMI... so that it can work as a PC if connected with monitor, keyboard and mouse.

Many many development boards are available now, as an example Versatile and Integrator are development boards support ARM processors. All what we need to operate it as a computer is to build an operating system that can work in this board. This idea already has been implemented, you can find many hardware platform that can already host Linux operating system. It is listed in the following link:

http://elinux.org/Main_Page

Bibliography

[1] http://www.davespace.co.uk/arm/introduction-to-arm/

[2] http://www.arm.com/products/processors/index.php

[3] http://elinux.org/Main_Page