EVOLUTION OF ARM MICROCONTROLLERS

ARM series of microcontrollers – ARM Families

Important ARM features

History of ARM series of Microcontrollers

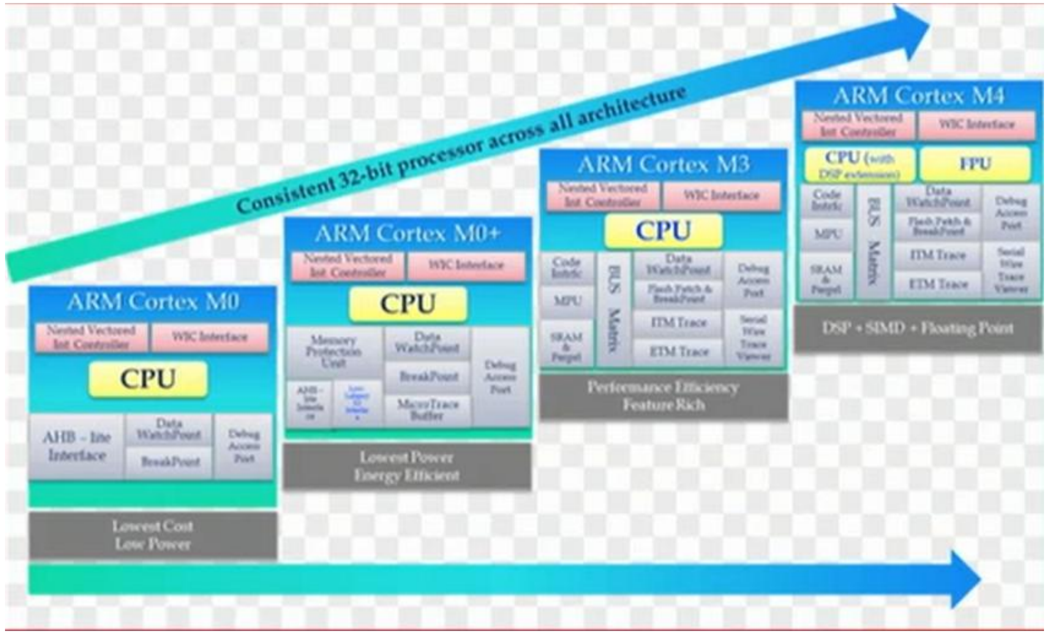
- Architectural ideas developed in 1983 by Acorn Computers
  - To replace the 8-bit 6502 microprocessor in BBC computers
  - The first commercial RISC implementation
- The company founded in 1990
  - **Advanced RISC Machine (ARM)**
  - Initially owned by Acorn, Apple and VLSI

Introduction

- 32-bit RISC architecture
- Developed by ARM Corporation, previously known as *Acorn RISC Machine*
- Licensed to companies that want to manufacture ARM based CPUs or SOC products
- Helps the licensee to develop their own-processors complaint with ARM instruction set architecture

Why do we talk about ARM?

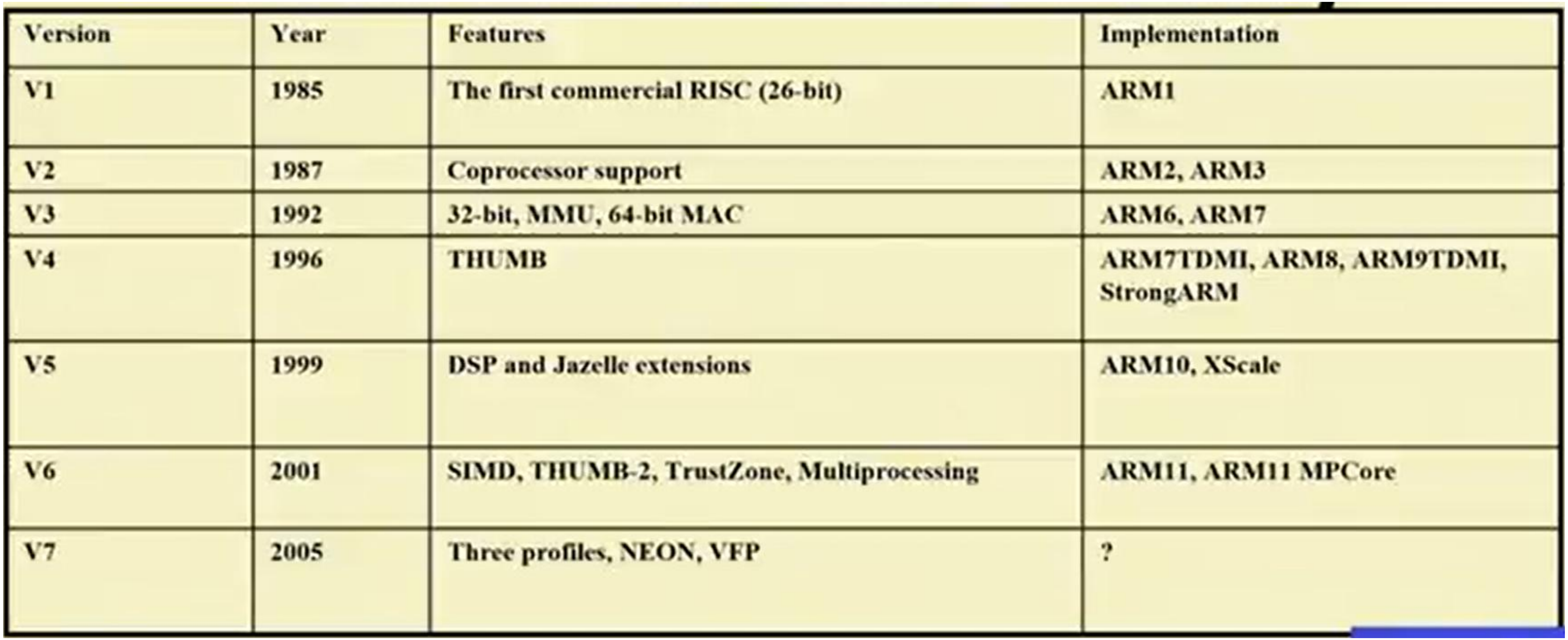
- One of the most widely used processor cores
- Some application examples:
  - ARM: iPod
  - ARM9: BenQ, Sony Ericsson
  - ARM11: Apple iPhone, Nokia N93, N100
  - 90% of 32-bit embedded RISC processors till 2010
- Mainly used in battery-operated devices:
  - Due to low power consumption and reason performance



Popular ARM Architectures

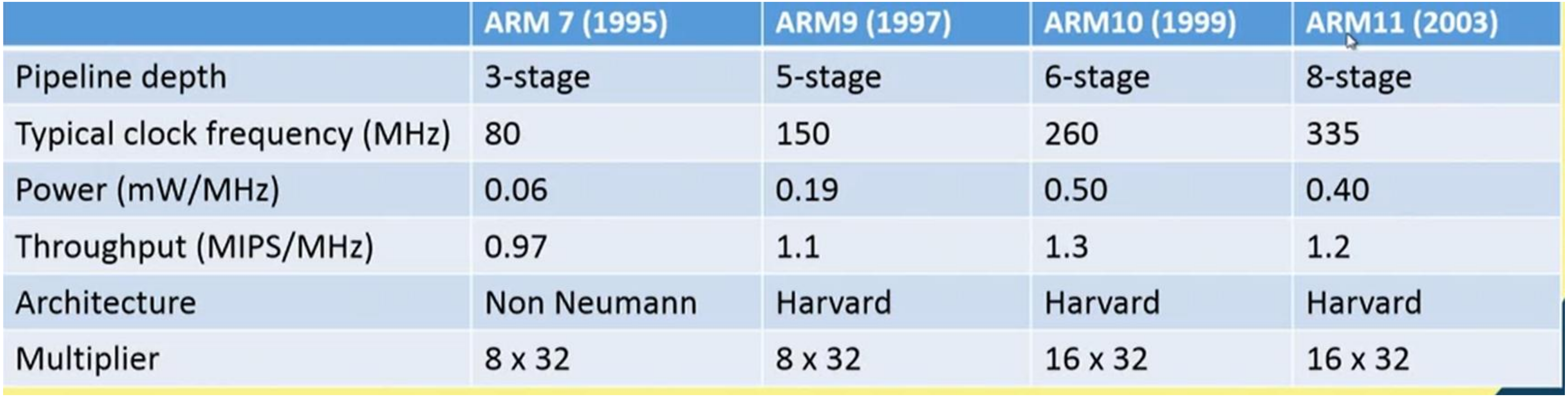
- ARM7
  - 3 pipeline stages (fetch/decode/execute)
  - High code density/low power consumption
  - Most widely used for low-end systems
- ARM9
  - Compatible with ARM7
  - 5 stages (fetch/decode/execute/memory/write)
  - Separate instruction and data cache
- ARM10
  - 6-stages (fetch/issue/decode/execute/memory/write)

ARM Architecture History



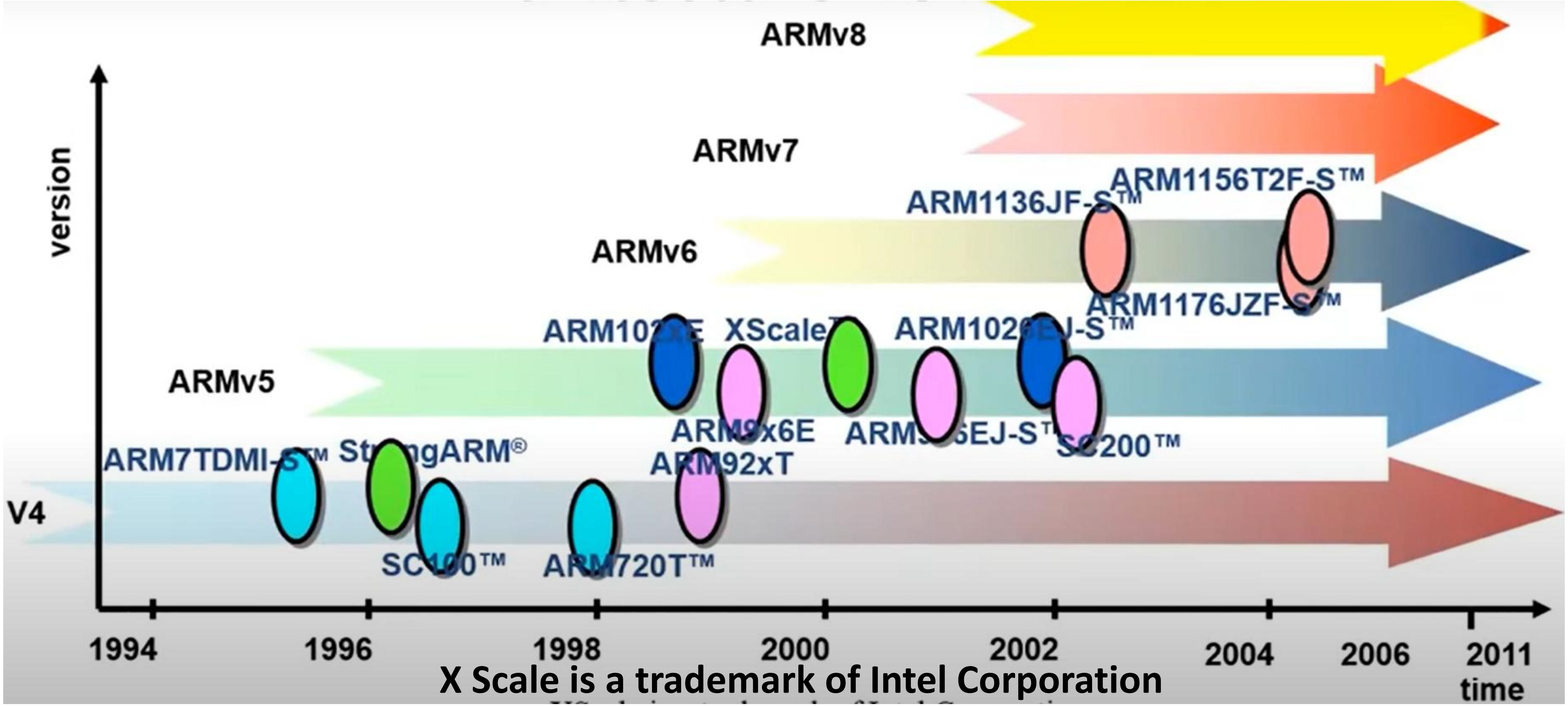
Version	Year	Features	Implementation
V1	1985	The first commercial RISC (26-bit)	ARM1
V2	1987	Coprocessor support	ARM2, ARM3
V3	1992	32-bit, MMU, 64-bit MAC	ARM6, ARM7
V4	1996	THUMB	ARM7TDMI, ARM8, ARM9TDMI, StrongARM
V5	1999	DSP and Jazelle extensions	ARM10, XScale
V6	2001	SIMD, THUMB-2, TrustZone, Multiprocessing	ARM11, ARM11 MPCore
V7	2005	Three profiles, NEON, VFP	?

ARM Family Comparison



	ARM 7 (1995)	ARM9 (1997)	ARM10 (1999)	ARM11 (2003)
Pipeline depth	3-stage	5-stage	6-stage	8-stage
Typical clock frequency (MHz)	80	150	260	335
Power (mW/MHz)	0.06	0.19	0.50	0.40
Throughput (MIPS/MHz)	0.97	1.1	1.3	1.2
Architecture	Non Neumann	Harvard	Harvard	Harvard
Multiplier	8 x 32	8 x 32	16 x 32	16 x 32

Architecture Revisions



X Scale is a trademark of Intel Corporation

ARM processor families

- ARMv1
  - First version of ARM processor
    - 26-bit addressing, no multiply/coprocessor
- ARMv2
  - ARM2, First commercial chip
  - Included 32-bit result multiply instructions/coprocessor support
- ARMv2a
  - ARM3 chip with on-chip cache
  - Added load and store
  - Cache mangement

ARM processor families

- ARMV3

  ARM6, 32 bit addressing, virtual

  Memory support
- ARMV4-ARM7TDMI, Strong ARM

  Includes MMU

  Unified 8K cache

  Five stage pipeline
- T: Thumb, 16-bit instruction set
- D: on-chip Debug support, enabling the processor to halt in response the processor to halt in response to a debug request

ARM processor families

- ARM v7

  ARM11

  Thumb 2 (mixed 16 and 32 bit)

  ARM cortex implementation

- ARM v8A

  32-bit and 64-bit

  A32, T32 and A64 instruction sets

  Virtual memory system

  Supporting rich operating systems

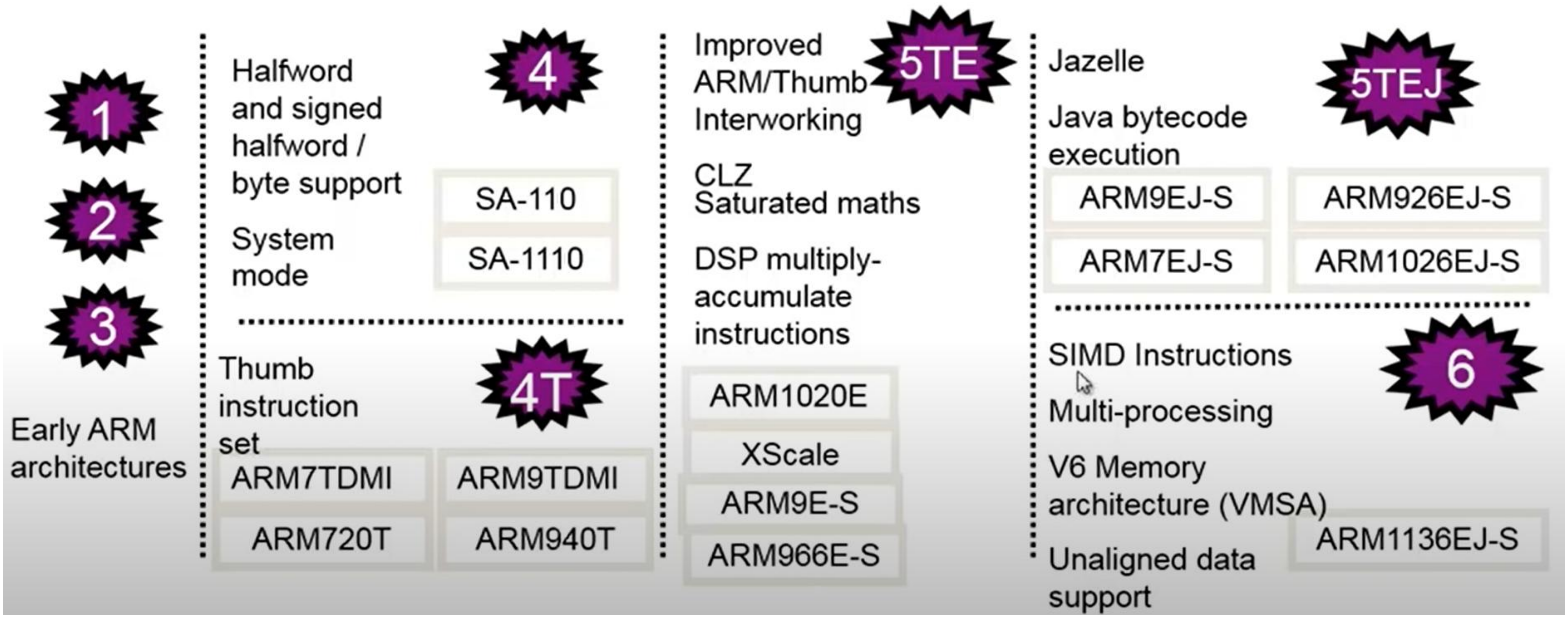
- ARM v8R, v8M

  Instructions used in single bit operation

  T32/Thumb® instruction set only

  Protected memory systems

Development of the ARM Architecture



1

2

3

Early ARM architectures

4

Halfword and signed halfword / byte support

System mode

SA-110

SA-1110

4T

Thumb instruction set

ARM7TDMI

ARM9TDMI

ARM720T

ARM940T

5TE

Improved ARM/Thumb Interworking

CLZ

Saturated maths

DSP multiply-accumulate instructions

ARM1020E

XScale

ARM9E-S

ARM966E-S

5TEJ

Jazelle

Java bytecode execution

ARM9EJ-S

ARM926EJ-S

ARM7EJ-S

ARM1026EJ-S

6

SIMD Instructions

Multi-processing

V6 Memory architecture (VMSA)

Unaligned data support

ARM1136EJ-S

ARM is based on RISC Architecture

- RISC supports simple but powerful instructions that execute in a single cycle at high frequency
- MAJOR DESIGN FEATURES:
  - Instructions: reduced set/single cycle/fixed length
  - Pipeline: decode in one stage / no need for microcode
  - Registers: large of general-purpose registers (GPR’s)
  - Load/store instructions work on registers on register’s only: load/store instructions to transfer data from/to memory
- Now-a-days CISC machines also implement RISC concepts

ARM features

- ARM architecture is different from pure RISC:
    - Variable cycle execution for certain instructions (multiple register load/store for higher code density)
    - In-line barrel shifter results in more complex instructions (improves performance and code density)
    - Thumb 16-bit instruction set (results in improvement in code density by about 30%)
    - Conditional execution (reduces branch and improves performance)
    - Enhanced instructions (some DSP instructions are present)

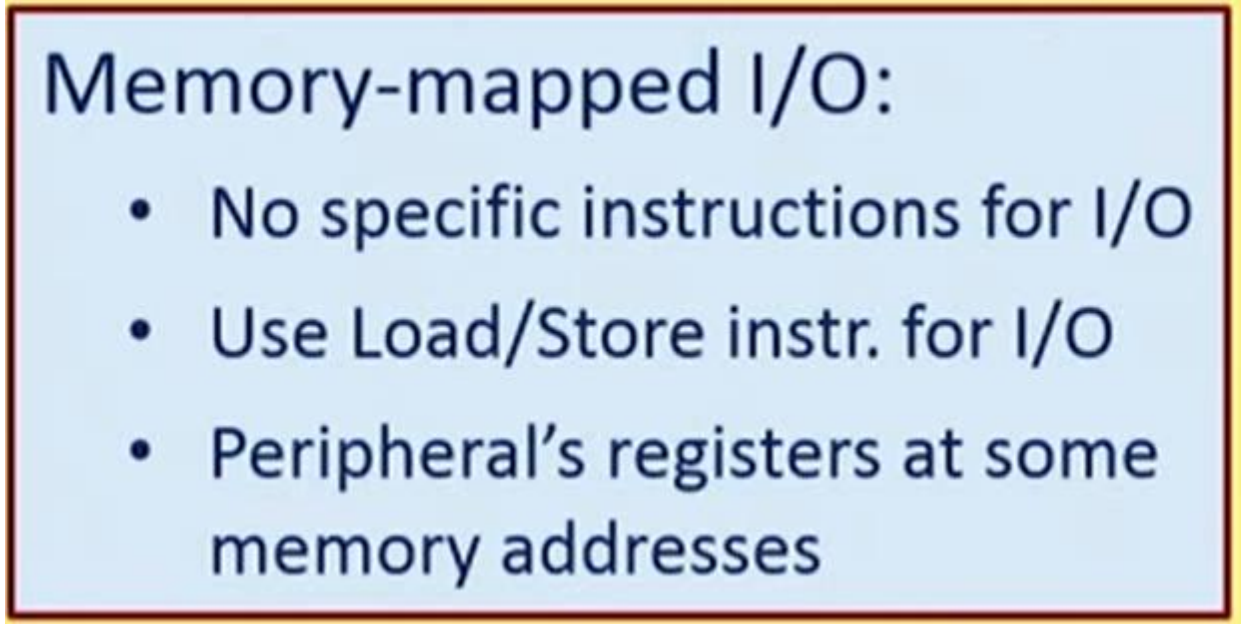
Features

- ARM cores are very simple, requires relatively lesser number of transistors, leaving enough space on die to realize other functionalities on the silicon
- Instruction set architecture and the pipeline design aimed at minimizing energy consumption
- Also capable of running 16-bit THUMB instruction set-greater code density and enhanced power saving
- Higher performance
- Highly modular architecture-the only mandatory part is the integer pipeline, all other components are optional
- Built-in JTAG debug port and on-chip embedded in-circuit emulator (ICE) that allows programs to be downloaded and fully debugged in-system

ARM Nomenclature

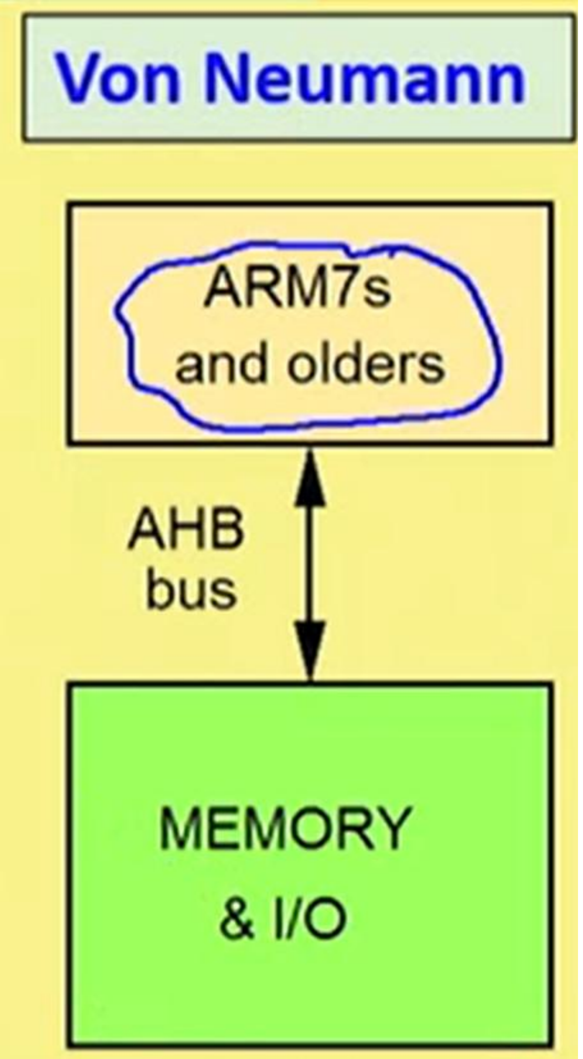
ARM {x}{y}{z}{T}{D}{M}{I}{E}{J}{J}{F}{-S}

X:	family
Y:	Memory management/protection unit
Z:	cache
T:	Thumb 16 bit decoder
D:	JTAG debug
M:	fast multiplier
I:	embedded ICE macrocell
E:	enhanced instruction
J:	Jazelle
F:	vector floating point
S:	synthesizible version

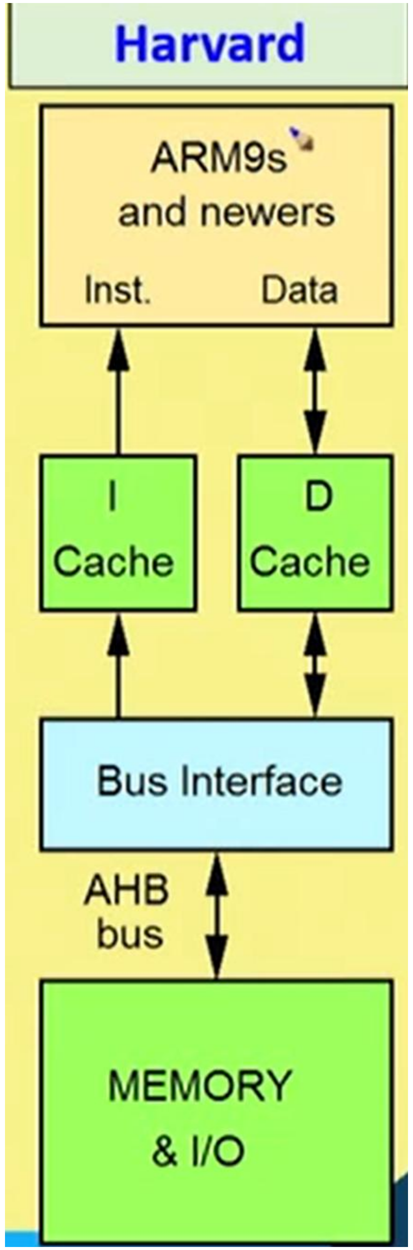


Memory-mapped I/O:

- No specific instructions for I/O
- Use Load/Store instr. for I/O
- Peripheral's registers at some memory addresses



Von Neumann
ARM7s and olders
AHB bus
MEMORY & I/O



Harvard
ARM9s and newers
Inst.
Data
I Cache
D Cache
Bus Interface
AHB bus
MEMORY & I/O