

ARM's processor lines

Dezső Sima

November 2018

(v3.2)

© Dezső Sima 2018

ARM's processor lines

- 1. Evolution of ARM
- 2. Evolution of the ARM ISA
- 3. Overview of ARM's processor families
- 4. Evolution of the Cortex-A series
ARMv7/v8.0 ISA-based models
- 5. Cortex-A models based on the ARMv8.0 ISA
- 6. Cortex-A models based on the ARMv8.2 ISA
- 7. Overview of ARM's Mali graphics series
- 8. References

Note

In the Lecture (Fall 2018) **only the following Sections will be discussed:**

Section 1: Evolution of ARM

From Section 2: Evolution of the ARM ISA

2.1: Overview

From Section 2.2: ISA extensions introduced to enhance compute capabilities

2.2.1: Overview

Section 3.4: Processors implementing the ARM v7 - ARM v8 ISA

Section 4: Evolution of the Cortex-A series models that are based on the ARMv7/v8.0 ISA

Section 6: Cortex-A models based on the ARMv8.2 ISA

1. Evolution of ARM

1. Evolution of ARM

- **ARM** (**ARM Holdings plc**) is a British multinational semiconductor company with its head office in Cambridge, acquired by Softbank (Japan) in 2016.
- The company
 - **designs** low power
 - **ARM processors** for the **embedded, mobile and server market**,
 - **mobile GPUs** (termed as **Mali GPUs**) as well as
 - **design tools** (development studios etc.),
 - and **licences**
 - their **IP (Intellectual Property)** including their **ISA** but **does not fabricate semiconductors**.

Example: ARM's IP offer relating the Cortex-A73 processor [89]

Graphic IP

- ARM Mali™-G71
- Mali-DP550
(Display processor)
- Mali-V550
(Video Processor)

Other IP

- ARM CoreLink™ CCI-550
(Cache Coherent Interconnect)
- CoreLink GIC-500
(Interrupt Controller)
- CoreLink MMU-500
(System Memory Management Unit)
- CoreLink TZC-400
(ARM TrustZone® Controller)
- CoreLink DMC-500/DMC-520
(Dynamic Memory Controller)
- ARM CoreSight™ SoC-400
(Debug and Trace)
- ARM POP™ (Physical IP)

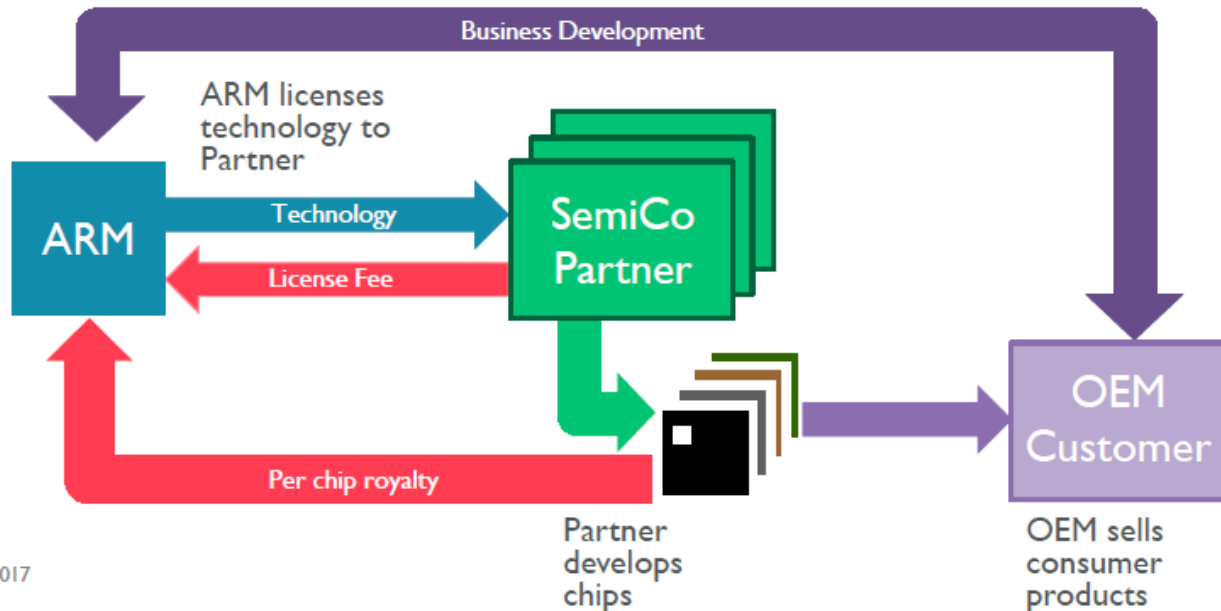
Tools

- ARM DS-5 Development Studio
- Fixed Virtual Platforms
- ARM Versatile™ Express
- ARM Compiler 6
- ARM Fast Models

ARM's business model [117]

ARM Business Model

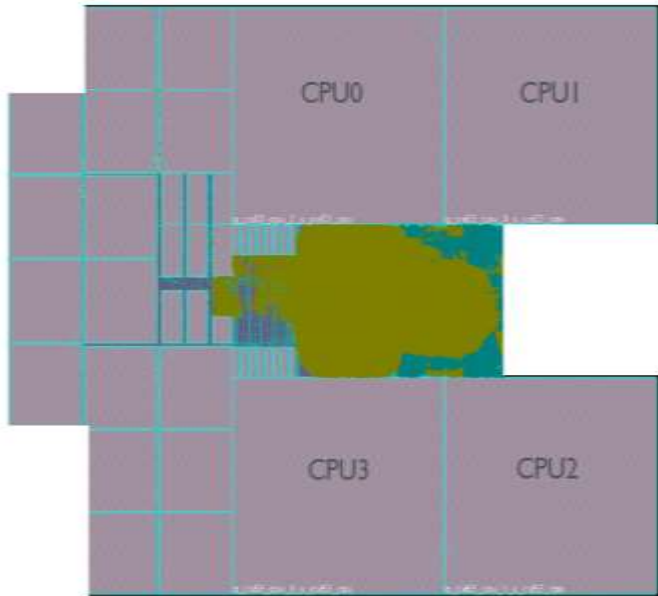
- ARM develops technology that is licensed to semiconductor companies
- ARM receives an upfront license fee and a royalty on every chip that contains its technology



1. Evolution of ARM (4)

Example for the range of configurability of a processor [90]

Configuration options of the Cortex-A35 ranging from mobile to deeply embedded



Cortex-A35

Quad core configuration

32K L1 caches, NEON, Crypto, 1MB L2 cache



< 0.4 mm²
28nm

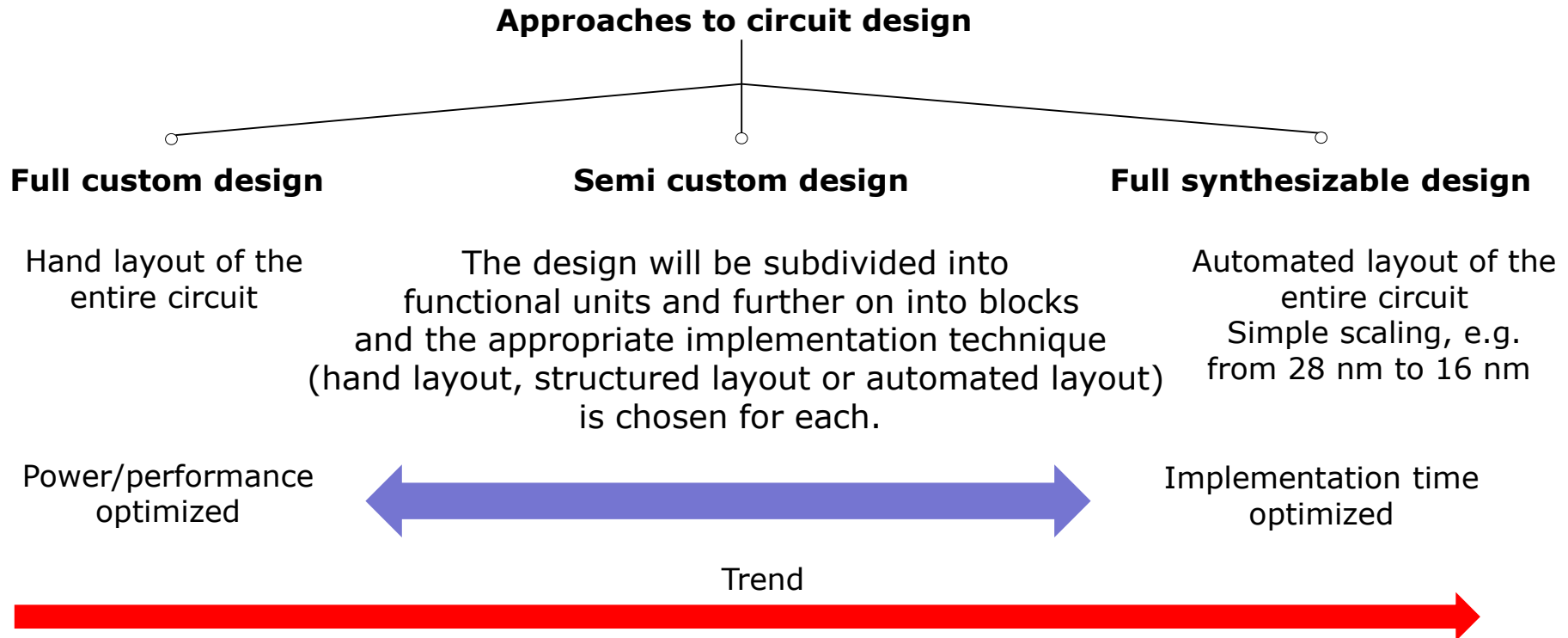
Cortex-A35

Smallest configuration

Single core, 8K L1 caches, no L2

1. Evolution of ARM (5)

Evolution of the approaches used to circuit design -1 [1]



Evolution of the approaches used to circuit design -2 [1]

- ARM's processor designs before the Cortex-A9 were partly hand layouts and partly automated layouts.
- ARM's first fully synthesizable design was the Cortex-A9 processor (announced in 2007).
- Recently, automated design tools are typically used for processor design.
- Over time more and more advanced standard cell libraries were developed that have a large variety of design options, e.g. cell types or drive strengths, which lessens the need for custom design.

1. Evolution of ARM (7)

Design approaches used by Qualcomm to develop recent processors [116]

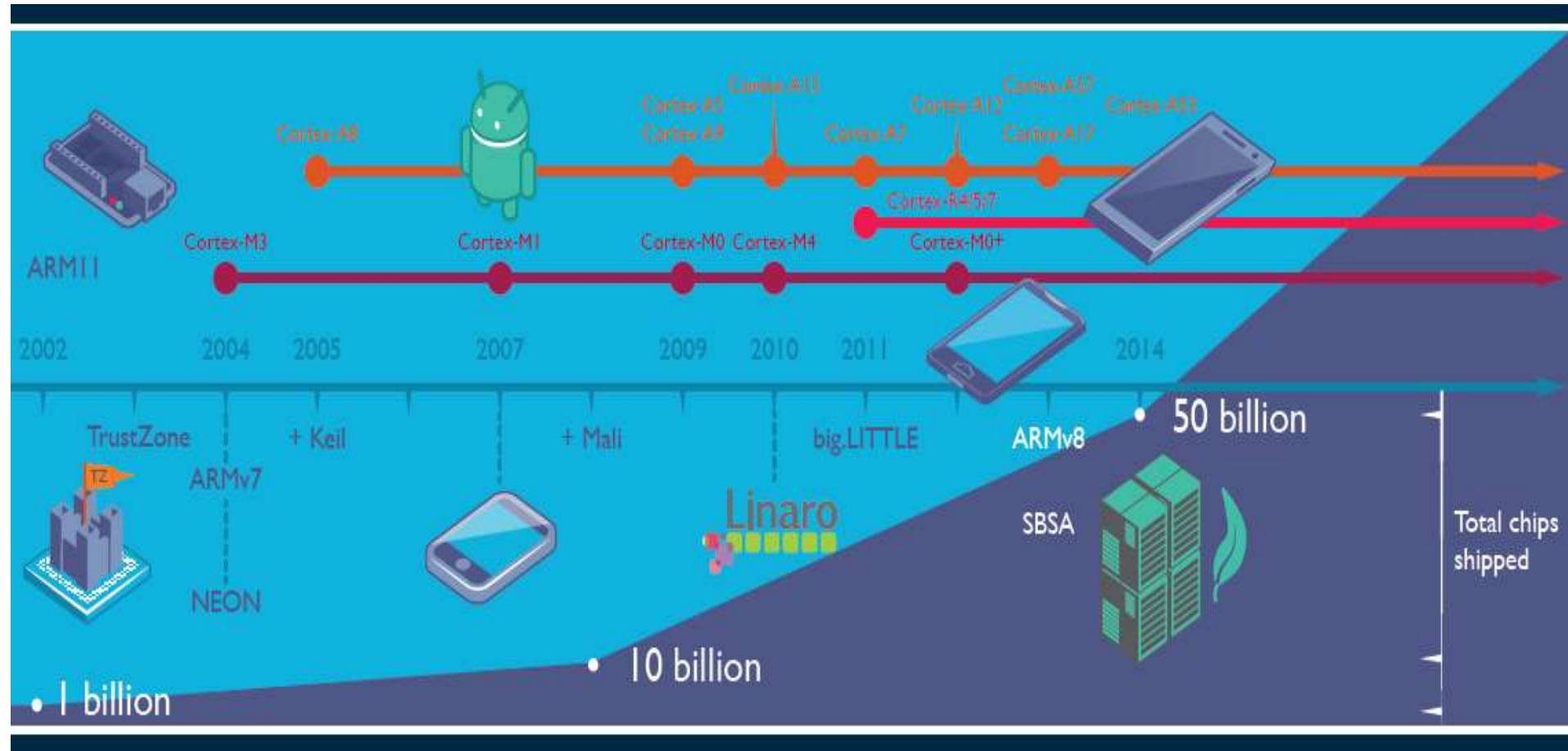
810	820	835
Stock ARM (Full synthesizable design)	Custom	Semi-Custom : Built on ARM Cortex® Technology
Rapid 64-bit deployment	High performance with improved power	Higher performance with extreme power efficiency
Limited System Integration	Tight system integration	Tight system integration
4xA53 2.0+ 4xA57 1.55 20 nm (2015)	2xKryo 2.15+ 2xKryo 1.59 14 nm (2015)	4xKryo 280 2.45+ 4xKryo 280 1.9 10 nm (2016)

Dominance of ARM designs in the embedded and mobile market (including smartphones and tablets)

- **ARM designs dominate recently the embedded and the mobile market** (including smartphones and tablets).
- As of 2014 more than 50 billion ARM based processors have been produced in total, up from 10 billion in 2008 [59], [19], as indicated in the next Figure.

1. Evolution of ARM (9)

Total number of ARM based chips shipped [19]



Keil: Software development tool for embedded processors

Linaro: Nonprofit company, established by ARM, Freescale, IBM, Samsung, ST-Ericsson and TI to support open source software developers using Linux on SoCs.

SBSA: Server Base System Architecture, a standardized server platform for 64-bit ARM processors.

1. Evolution of ARM (10)

Historical remarks [60], [61] -1

- ARM's parent company is **Acorn Computers** (UK).
- Acorn Computers started their *Acorn RISC Machine* project in October 1983 (two years after the introduction of the IBM PC) to develop an own powerful processor for a line of business computers.
- The acronym **ARM** was coined originally at this time (1983) from the designation *Acorn RISC Machine*.
- In 1990 the company **Advanced RISC Machines Ltd. (ARM Ltd.)** was founded as a joint venture of Acorn Computers, Apple Computers and VLSI Technology.
- Accordingly, also the interpretation of ARM was changed to "Advanced RISC Machines".

Historical remarks -2



Figure: The headquarters of ARM Ltd. about 1990 [62]

Historical remarks -3

Finally, in 1998 the company [went to the stock exchange](#) and its name was changed to [ARM Holdings plc](#), to its current designation.

Historical remarks -4



Figure: ARM's recent headquarter in Cambridge (UK) [78]

Acquisition of ARM by Softbank (Japan)

- Announced in 07/2016
- Completed in 09/2016
- Price: 31 bUSD (ARM's revenues in 2015 \approx 1.5 bUSD)

2. Evolution of the ARM ISA

- 2.1 Overview
- 2.2 ISA extensions introduced to enhance compute capabilities
- 2.3 ISA extensions introduced to reduce the code size
- 2.4 ISA extensions introduced to enhance security

(Only Section 2.1 and 2.2.1 will be discussed)

2.1 Overview

2.1 Overview

- There are **eight ARM ISA versions**, designated as **ARMv1 to ARMv8**. These are described in the related Architecture Reference Manuals.
- The **earliest versions** (ARMv1 and ARMv2) provided an **address range of only 26 bits**, the **first ISA version with 32 bit address range** was the ARMv3.
- Accordingly, we consider the ISA version **ARMv3** as **ARM's basic ISA** and discuss its evolution subsequently.

2.1 Overview (2)

Example: Half page of the ARMv8 Architecture Reference Manual []

(1/2 from 6354)

C7.2.332 UQSHL (register)

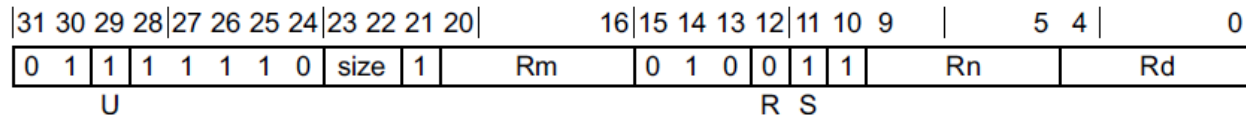
Unsigned saturating Shift Left (register). This instruction takes each element in the vector of the first source SIMD&FP register, shifts the element by a value from the least significant byte of the corresponding element of the second source SIMD&FP register, places the results in a vector, and writes the vector to the destination SIMD&FP register.

If the shift value is positive, the operation is a left shift. Otherwise, it is a right shift. The results are truncated. For rounded results, see [UQRSHL](#).

If overflow occurs with any of the results, those results are saturated. If saturation occurs, the cumulative saturation bit [FPSR.QC](#) is set.

Depending on the settings in the [CPACR_EL1](#), [CPTR_EL2](#), and [CPTR_EL3](#) registers, and the current Security state and Exception level, an attempt to execute the instruction might be trapped.

Scalar



Scalar variant

UQSHL <V><d>, <V><n>, <V><m>

Decode for this encoding

```
integer d = UInt(Rd);
integer n = UInt(Rn);
integer m = UInt(Rm);
integer esize = 8 << UInt(size);
```

2.1 Overview (3)

Key features of ARM's basic ISA

- It is a **32-bit RISC ISA** capable to process basically **32-bit scalar FX or logical data**.
- The ISA has **16 32-bit registers**, called the **core registers**.
- **13** out of them are used as **general purpose registers (GPRs)**, the remaining three are **dedicated registers**, as shown below.

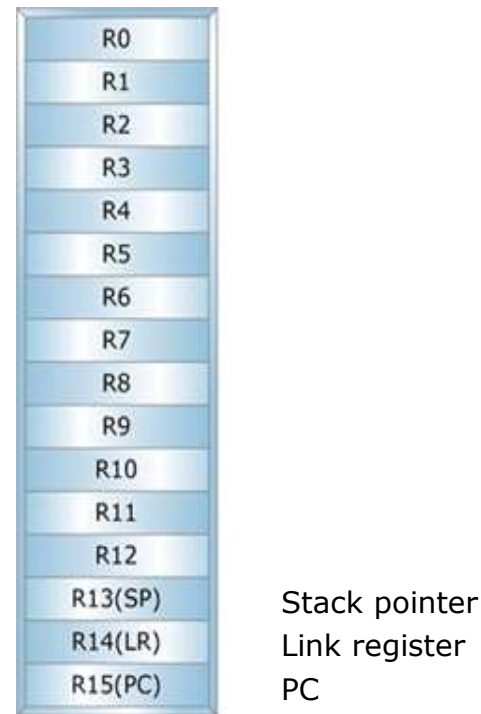
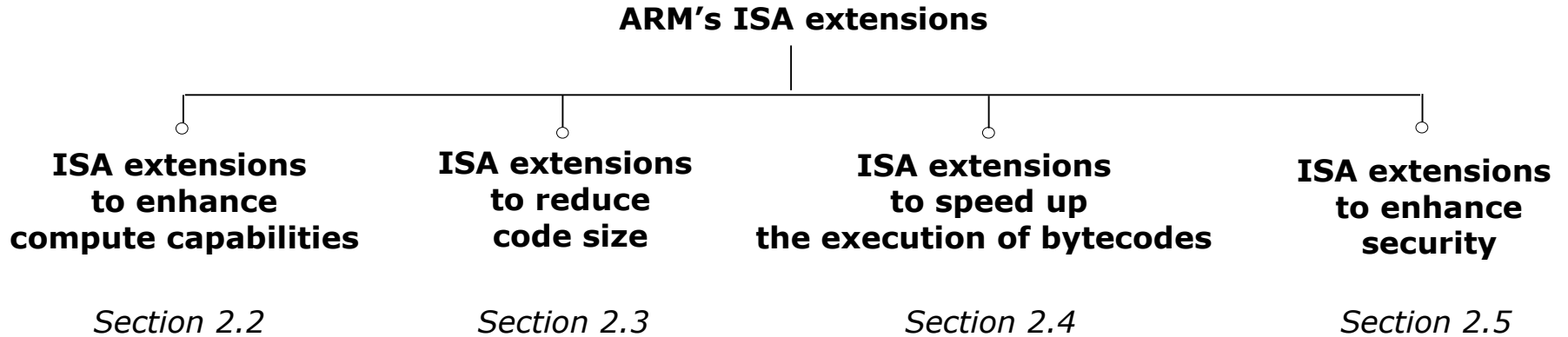


Figure: The core registers of the ARM ISA (in the ISA versions ARMv3-ARMv7) [63]

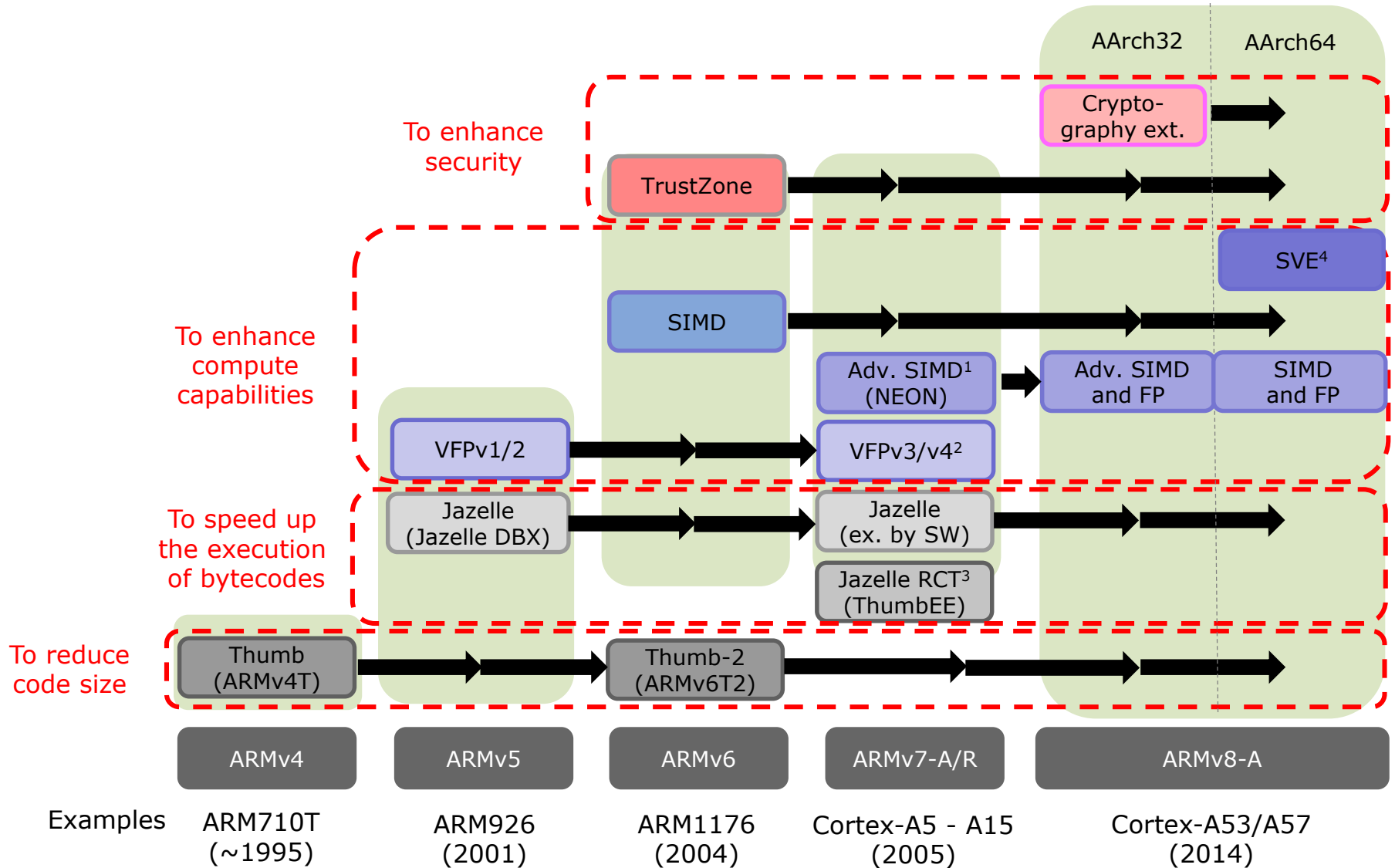
2.1 Overview (4)

Main extensions introduced in ARM's basic ISA (simplified) -1



2.1 Overview (5)

Main extensions introduced in ARM's basic ISA (simplified) -2 (Based on [64])



Remarks: See on the next slide.

Remarks

¹The Advanced SIMD architecture extension is commonly referred to as the NEON technology.

The VFP subset became deprecated in the ARMv8 ISA.

³Jazelle-RTC (ThumbEE) became deprecated in ARMv7 Issue C in 10/2011.

⁴The SVE (Scalable Vector Extension) subset became introduced in the ARMv8 ISA only in 2016.

2.2 ISA extensions introduced to enhance compute capabilities

- 2.2.1 Overview
- 2.2.2 The GPR register set based SIMD extension
- 2.2.3 Secondary register set based VFP and NEON extensions
- 2.2.4 The SVE register set based SVE extension

Only Section 2.2.1 will be discussed.

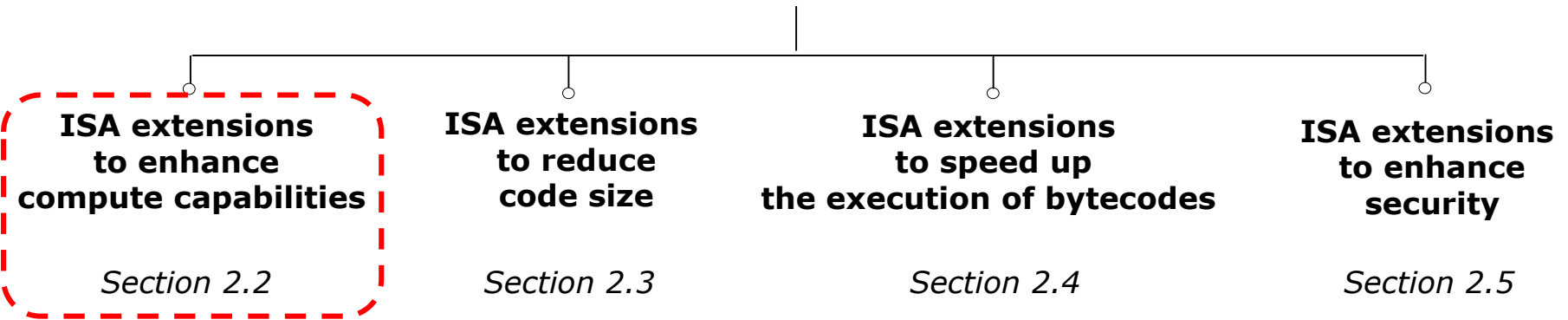
2.2.1 Overview

2.2.1 Overview (1)

2.2 ISA extensions introduced to enhance compute capabilities

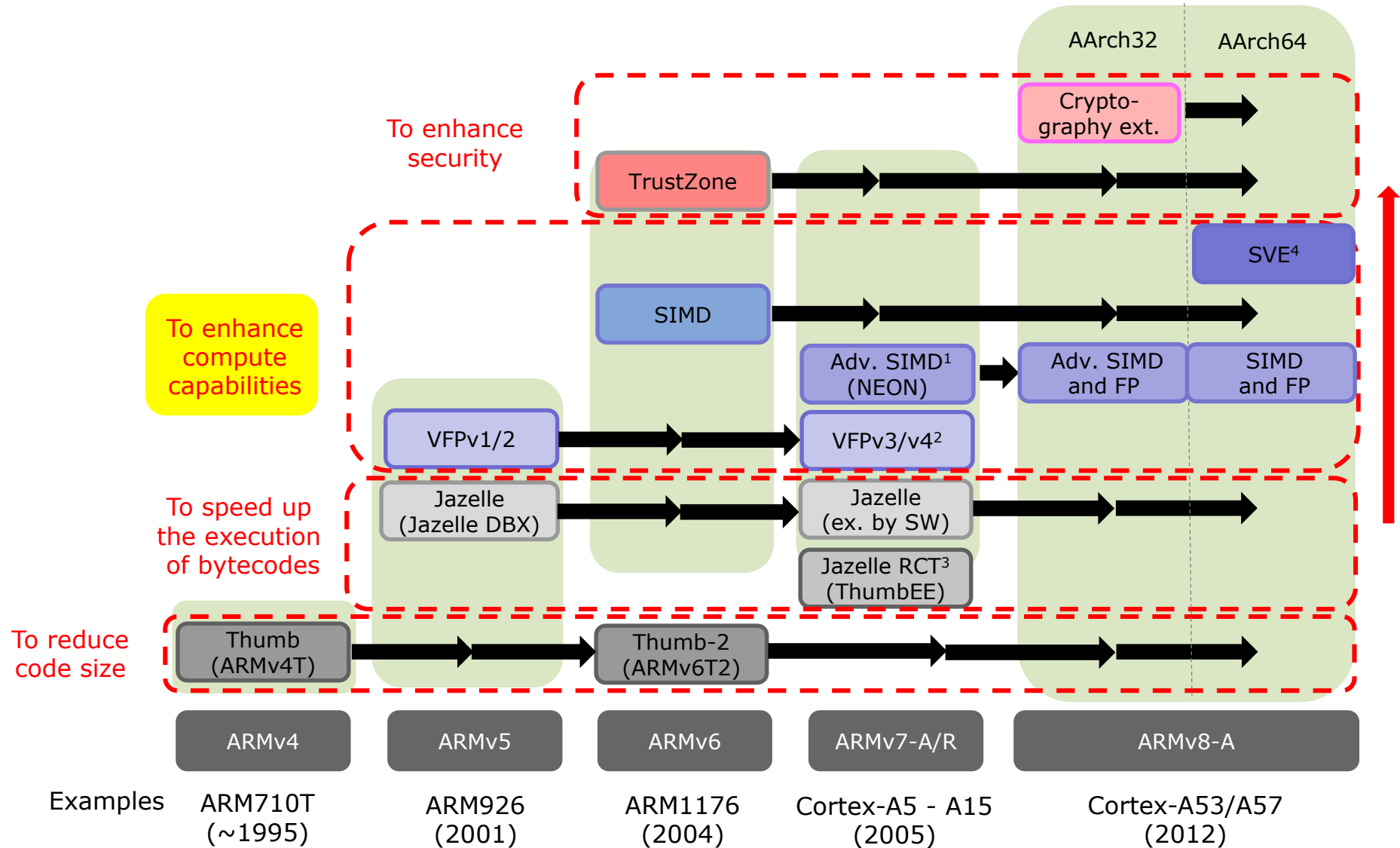
2.2.1 Overview

ARM's ISA extensions



2.2.1 Overview (2)

Overview of the ISA extensions introduced to enhance compute capabilities



Examples

ARM710T
(~1995)

ARM926
(2001)

ARM1176
(2004)

Cortex-A5 - A15
(2005)

Cortex-A53/A57
(2012)

Remarks: See on the slide 2.1 Overview (5).

ISA extensions introduced to enhance compute capabilities

ARM's ISA extensions to enhance compute capabilities

```
graph TD; Title[ARM's ISA extensions to enhance compute capabilities] --- C1[The GPR register set based SIMD extension]; Title --- C2[A secondary register set added with FP and SIMD extensions]; Title --- C3[An SVE register set added with SIMD extensions]; C1 --- D1[It supports FX-SIMD operations]; C1 --- A1[ARMv6]; C1 --- S1[Section 2.2.2]; C2 --- D2[It supports FX/FP scalar and SIMD operations]; C2 --- A2[ARMv5]; C2 --- S2[Section 2.2.3]; C3 --- D3[It supports FX/FP SIMD operations on long vector data]; C3 --- A3[ARMv8.2]; C3 --- S3[Section 2.2.4];
```

The GPR register set based SIMD extension

It supports
FX-SIMD operations

ARMv6

Section 2.2.2

A secondary register set added with FP and SIMD extensions

It supports
FX/FP scalar and SIMD operations

ARMv5

Section 2.2.3

An SVE register set added with SIMD extensions

It supports
FX/FP SIMD operations
on long vector data

ARMv8.2

Section 2.2.4

ISA extensions introduced to enhance compute capabilities

ARM's ISA extensions to enhance compute capabilities

**The GPR register set based
SIMD extension**

It supports
FX-SIMD operations

ARMv6

Section 2.2.2

**A secondary register set added
with FP and SIMD extensions**

It supports
FX/FP scalar and SIMD operations

ARMv5

Section 2.2.3

**An SVE register set added
with SIMD extensions**

It supports
FX/FP SIMD operations
on long vector data

ARMv8.2

Section 2.2.4

2.2.1 Overview (5)

ISA extensions introduced to enhance compute capabilities (until 2012)

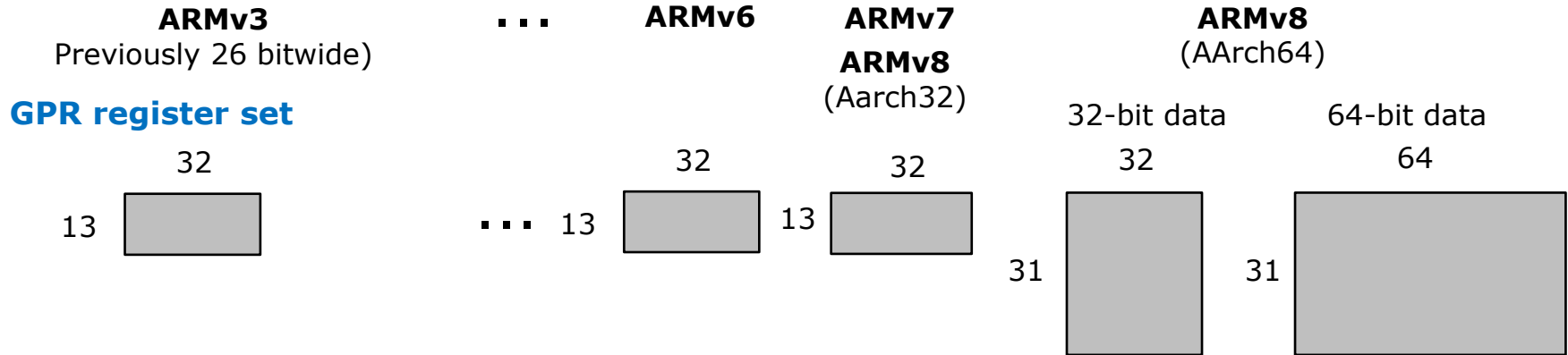
ARM ISA		Name of the extensions	Basic arch.		GPR-based FX-SIMD data type extension	Secondary register set based extensions			
Name			GPRs	Data type		Available reg. set	Scalar data types	Vector data types	
ARMv1	1985		n.a.						
ARMv2	1989		n.a.						
ARMv3	1991								
ARMv4	1996								
ARMv5	Year	VFP(v1)¹	13x32	FX32		FP register set 32x32/16x64	FP 32/64	FP 32/64 Serially ex.	
		VFP2¹							
ARMv6		SIMD				32-bit wide FX8/16			
ARMv7	2005	VFPv3²				Adv. SIMD and FP register set 32x64 or 16X128	+FP 16 ⁴	FP 16 ⁴ /32/64 Serially executed	
		VFPv4^{2,3,4}		+ FMA					
		Adv. SIMD (NEON)^{4,5}		32x64 or16x128	64/128-bit wide FX 8/16/32/64 FP 16 ⁵ /32/64 +FMA ⁵				
ARMv8	2012	AArch32	31x64	FX32 /64	32-bit wide FX8/16	32x64 or 16x128	FP 16 ⁴ /32/64	As for ARMv7 Adv. SIMD	
		AArch64			32/64-bit wide FX8/16/32	SIMD and FP register set 32x128	FX 8/.../64 FP 16 ⁴ /32/64	As for ARMv7 Adv. SIMD	

Remarks

- ¹VFPv2 vs. VFP(v1): VFPv2 adds some enhancements and modifications to VFPv1
- ²VFPv3/v4 and advanced SIMD register space:
Certain processors implement only 16 64-bit wide registers with the option to use this register space as 32 32-wide registers.
- ³VFP4 is implemented on certain ARMv7 processors.
It adds FMA (Fused Multiply Accumulate) instructions to the VFPv3 instruction set.
- ⁴FP16 supports only data conversion between FP16 and FP32/FP64.
- ⁵In the Advanced SIMD (NEON) extension FP16 is supported only if VFP3/VFP4 is implemented respectively, FMA if VFP4 is implemented.

2.2.1 Overview (7)

The GPR register set based SIMD extension introduced in the ARMv6 ISA



Scalar data

FX32

Basic ISA

FX32

FX64

**SIMD
extension**

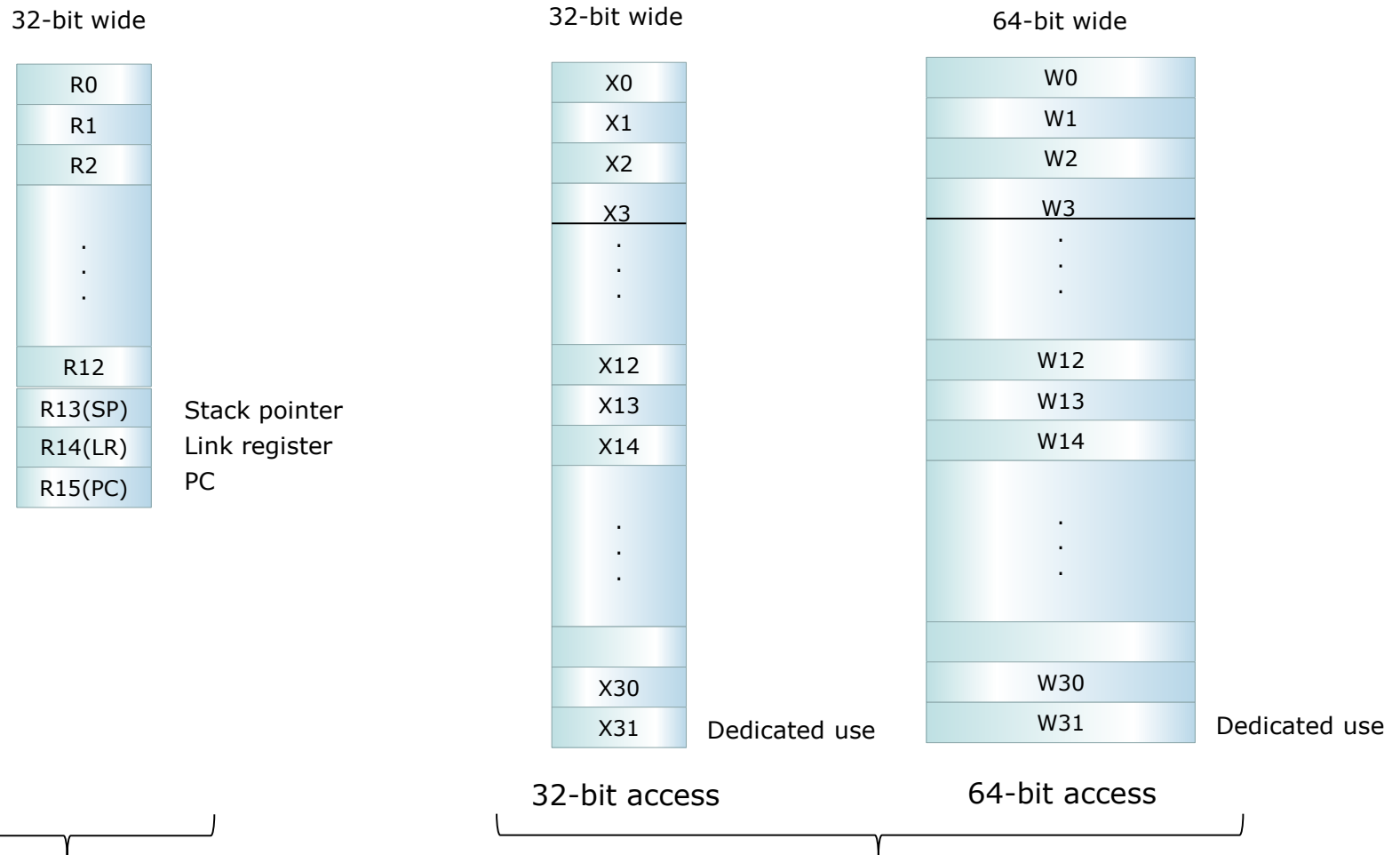
SIMD data

32-bit wide FX-SIMD
FX8/16 operations

2.2.1 Overview (8)

Extension of the GPR register set available in the ARMv8 ISA [63], [66]

The **ARMv8 ISA** expanded the number of GPRs as seen in the Figure below.



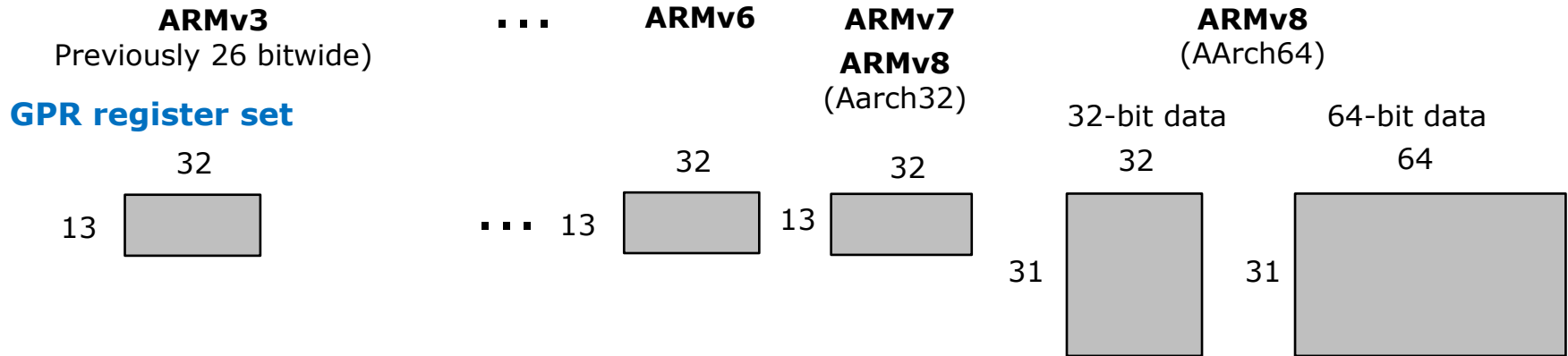
GPRs in the **ARMv3-ARMv7** and the **ARMv8 AArch32** execution mode

GPRs in the **ARMv8 AArch64** execution mode



2.2.1 Overview (8b)

The GPR register set based SIMD extension introduced in the ARMv8 ISA



Scalar data

FX32

Basic ISA

FX32

FX64

SIMD extension

SIMD data

32-bit wide FX-SIMD
FX8/16 operations

SIMD data

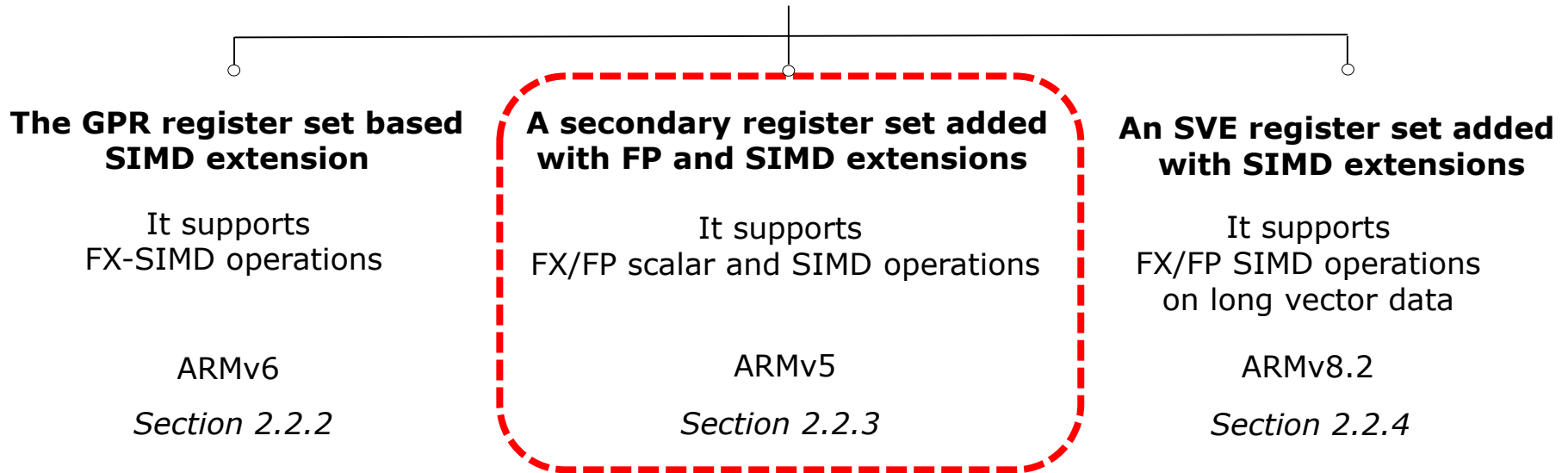
32-bit wide FX-SIMD
FX8/16 operations

SIMD data

64-bit wide FX-SIMD
FX8/16/32 operations

Secondary register set based FP and SIMD extensions -1

ARM's ISA extensions to enhance compute capabilities

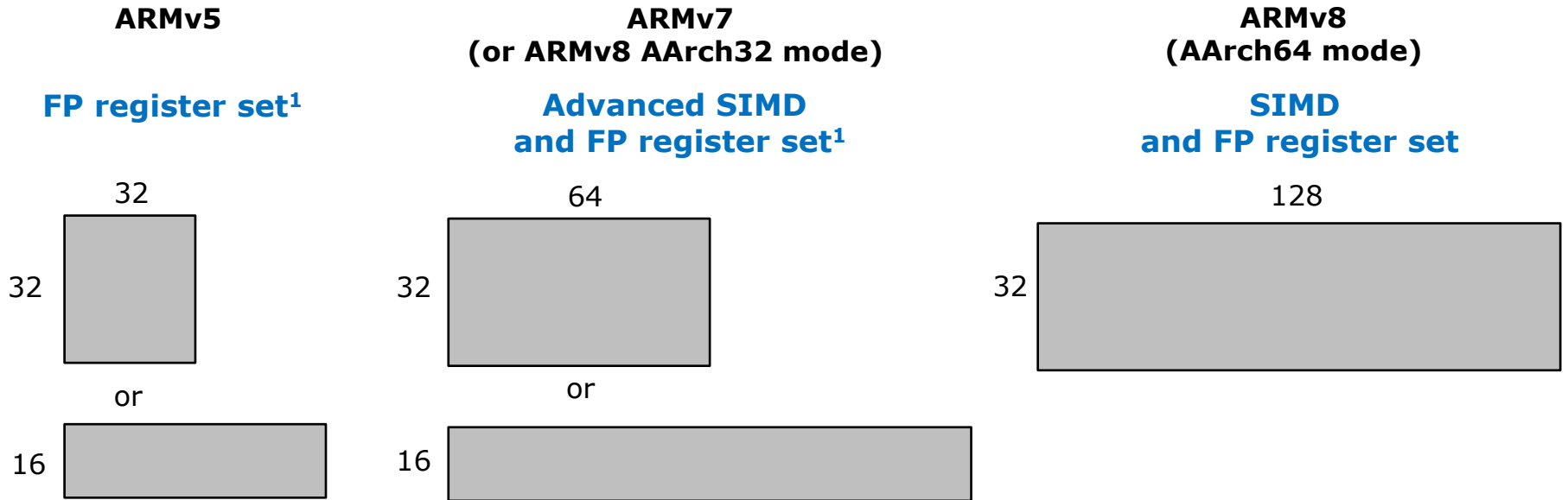


2.2.1 Overview (10)

Secondary register set based FP and SIMD extensions -2

ARM ISA		Name of the extensions	Basic arch.		GPR-based FX-SIMD data type extension	Secondary register set based extensions			
Name			GPRs	Data type		Available reg. set	Scalar data types	Vector data types	
ARMv1	1985		n.a.	FX32					
ARMv2	1989		n.a.						
ARMv3	1991								
ARMv4	1996								
ARMv5	Year	VFP(v1)¹	13x32		FX32	32-bit wide FX8/16	FP register set 32x32/16x64	FP 32/64	FP 32/64 Serially ex.
		VFP2¹							
ARMv6		SIMD							
ARMv7	2005	VFPv3²					Adv. SIMD and FP register set 32x64 or 16x128	+FP 16 ⁴	FP 16 ⁴ /32/64 Serially executed
		VFPv4^{2,3,4}							
		Adv. SIMD (NEON)^{4,5}					32x64 or 16x128		
ARMv8	2012	AArch32	31x64	FX32/64	32-bit wide FX8/16	32x64 or 16x128	FP 16 ⁴ /32/64	As for ARMv7 Adv. SIMD	
		AArch64			64-bit wide FX8/16/32	SIMD and FP register set 32x128	FX 8/.../64 FP 16 ⁴ /32/64	As for ARMv7 Adv. SIMD	

Evolution of the secondary register set

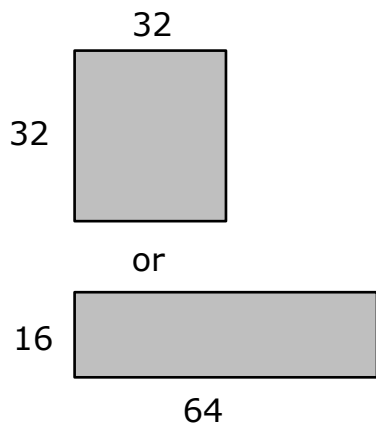


2.2.1 Overview (11b)

Secondary register set based FP and SIMD extensions -3

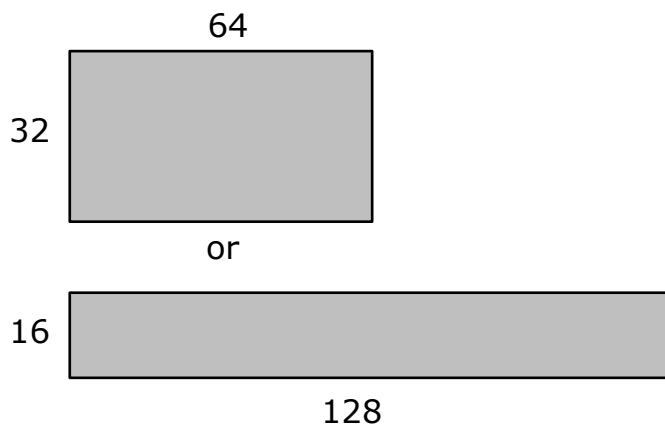
ARMv5

FP register set¹



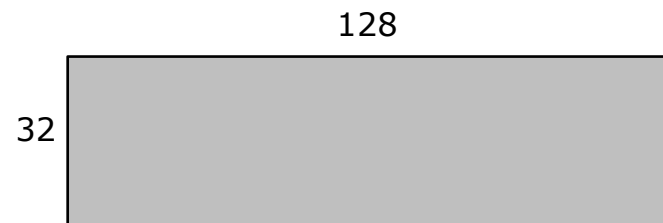
ARMv7 (or ARMv8 AArch32 mode)

Advanced SIMD and FP register set¹



ARMv8 (AArch64 mode)

SIMD and FP register set



VFPv1/v2 extension

Scalar data

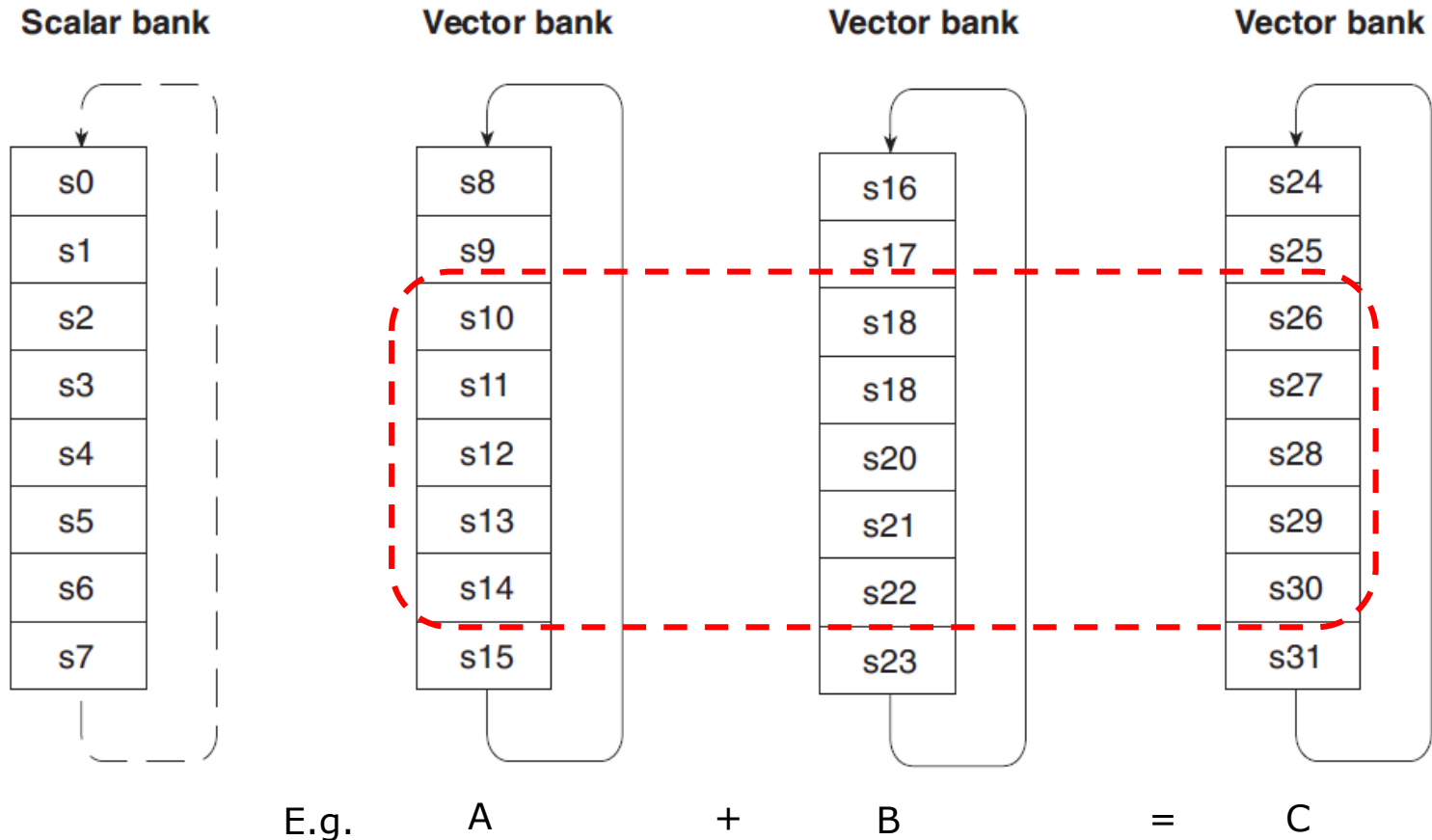
FP32/64

Serially processed FP vector data

Up to 8xFP32 or 4xFP64
operations serially executed

2.2.1 Overview (12)

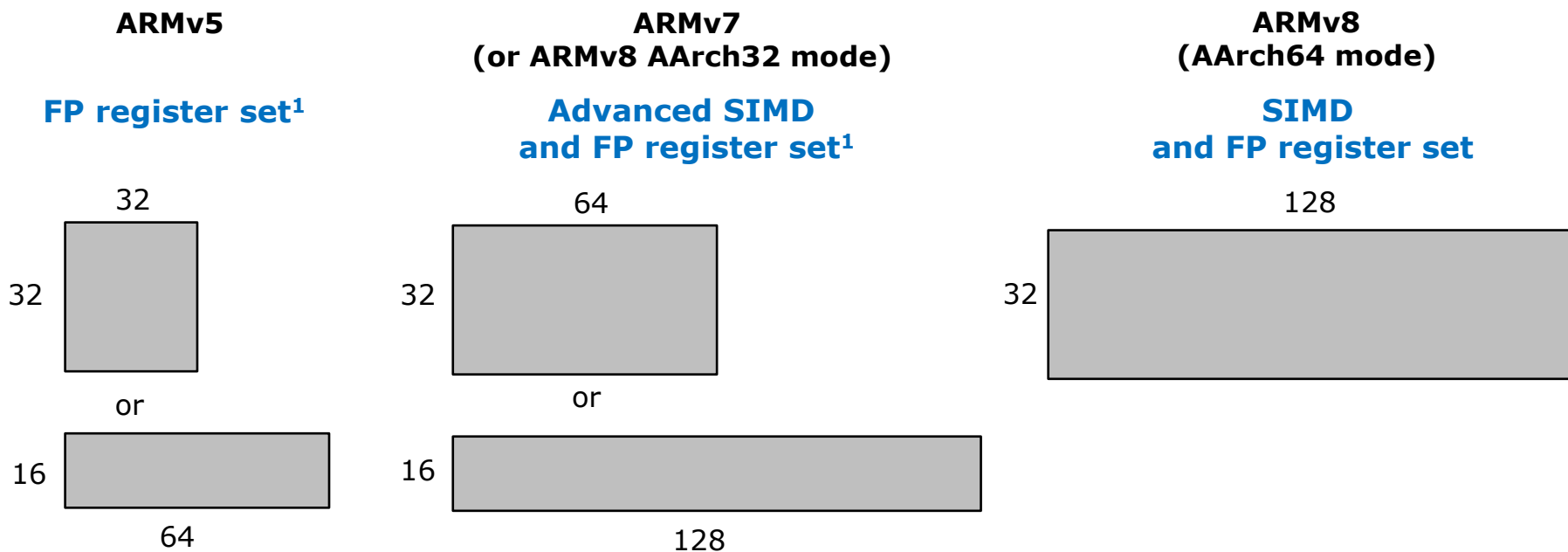
Example: Serially processed 5-element vector operation [65]



- In the example the **input operands** are taken from the registers s10...s14 and s18...s22, and the **result** is written into the registers s26...s30.
- The **execution is sequential** (like a hardware implemented subroutine).

2.2.1 Overview (11b)

Secondary register set based FP and SIMD extensions -3



VFPv1/v2 extension

Scalar data

FP32/64

Serially processed FP vector data

Up to 8xFP32 or 4xFP64
operations serially executed

VFPv3/v4 extension²

Scalar data

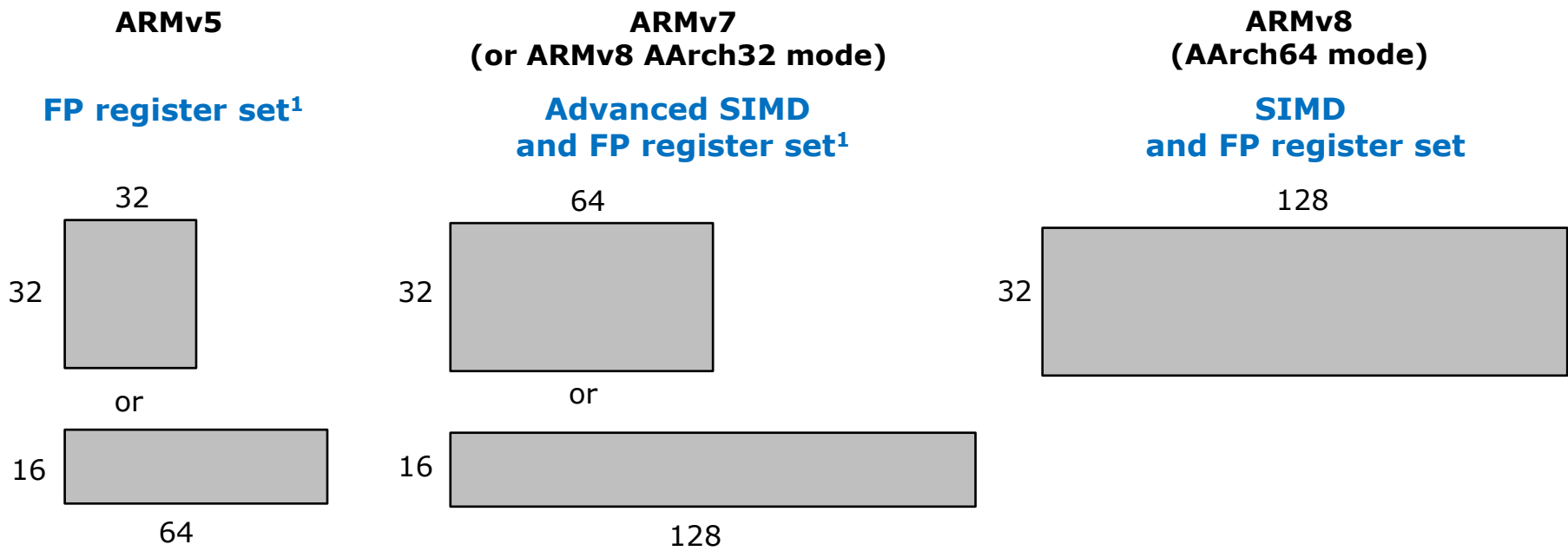
FP16¹/32/64

Serially processed FP vector data

Up to 8xFP32 or 4xFP64
operations serially executed

2.2.1 Overview (11b)

Secondary register set based FP and SIMD extensions -3



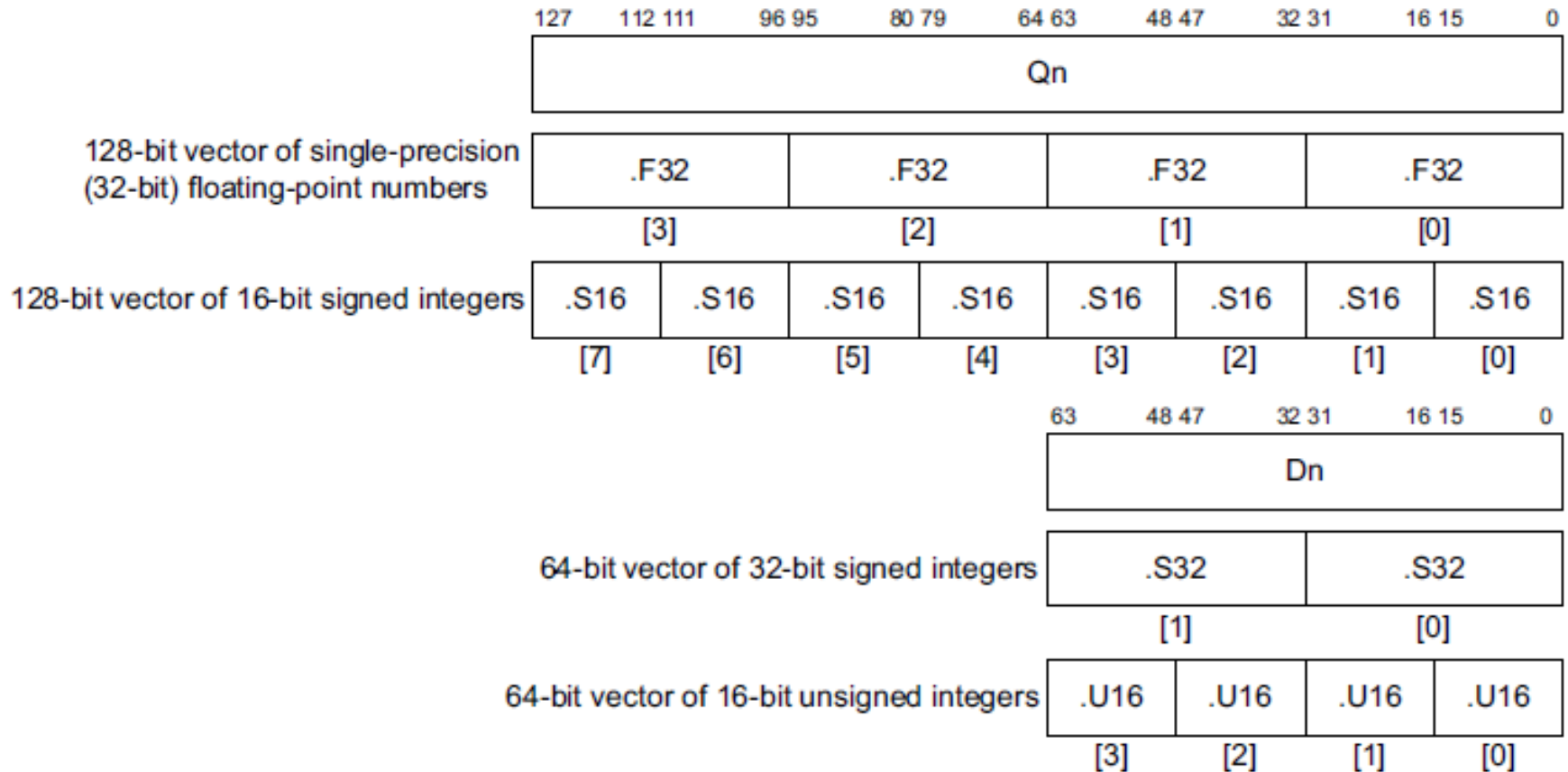
VFPv1/v2 extension
Scalar data
FP32/64
Serially processed FP vector data
Up to 8xFP32 or 4xFP64 operations serially executed

VFPv3/v4 extension²
Scalar data
FP16¹/32/64
Serially processed FP vector data
Up to 8xFP32 or 4xFP64 operations serially executed

Advanced SIMD (NEON) extension
SIMD data
64/128-bit wide SIMD (vector) data
FX8/16/32/64
FP16³/32/64 operations

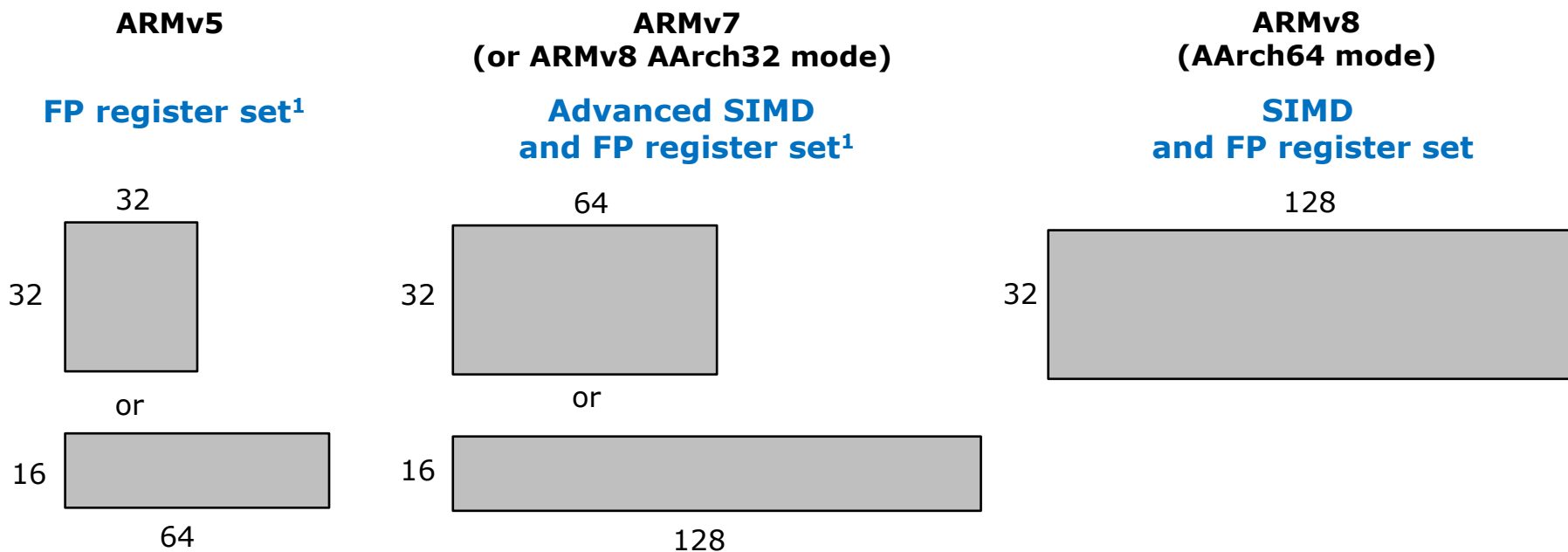
2.2.1 Overview (13)

Example vector data formats of the Advanced SIMD (NEON) extension [86]



2.2.1 Overview (11b)

Secondary register set based FP and SIMD extensions -3



VFPv1/v2 extension

Scalar data

FP32/64

Serially processed FP vector data

Up to 8xFP32 or 4xFP64 operations serially executed

VFPv3/v4 extension²

Scalar data

FP16¹/32/64

Serially processed FP vector data

Up to 8xFP32 or 4xFP64 operations serially executed

SIMD and FP extension

Scalar data

FP16¹/32/64

SIMD (vector) data

64/128-bit wide SIMD data
FX8/16/32/64
FP16³/32/64 operations

Advanced SIMD (NEON) extension

SIMD data

64/128-bit wide SIMD (vector) data
FX8/16/32/64
FP16³/32/64 operations

¹FP16 supports only data conversion between FP16 and FP32/FP64.

The SVE register set based SIMD extension -1

ARM's ISA extensions to enhance compute capabilities

**The GPR register set based
SIMD extension**

It supports
FX-SIMD operations

ARMv6

Section 2.2.2

**A secondary register set added
with FP and SIMD extensions**

It supports
FX/FP scalar and SIMD operations

ARMv5

Section 2.2.3

**An SVE register set added
with SIMD extensions**

It supports
FX/FP SIMD operations
on long vector data

ARMv8.2

Section 2.2.4

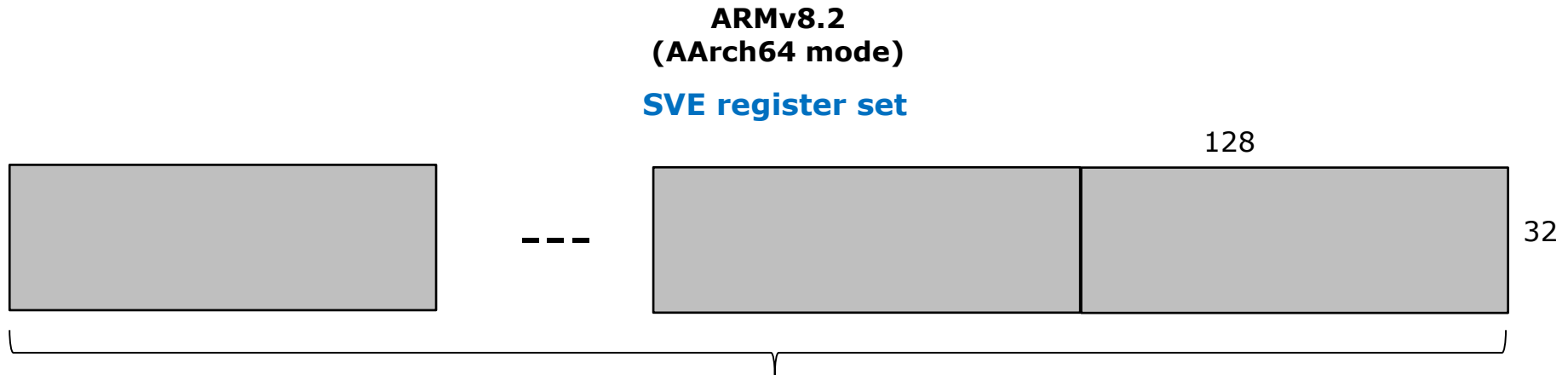
2.2.1 Overview (15)

The SVE register set based SIMD extension -2

ARM ISA		Name of the extensions	The SVE register set based SIMD extension		
Name	Year		Available register set	Scalar data types	Vector data types
ARMv8.2	2016	SVE	SVE register set 32 registers each (1...16)x128-bit wide	---	FX 8/16/32/64/128 FP 16/32/64 (1...16)x128-bit wide FMA available

2.2.1 Overview (16)

The SVE register set based SIMD extension -3



SVE extension

SIMD (vector) data

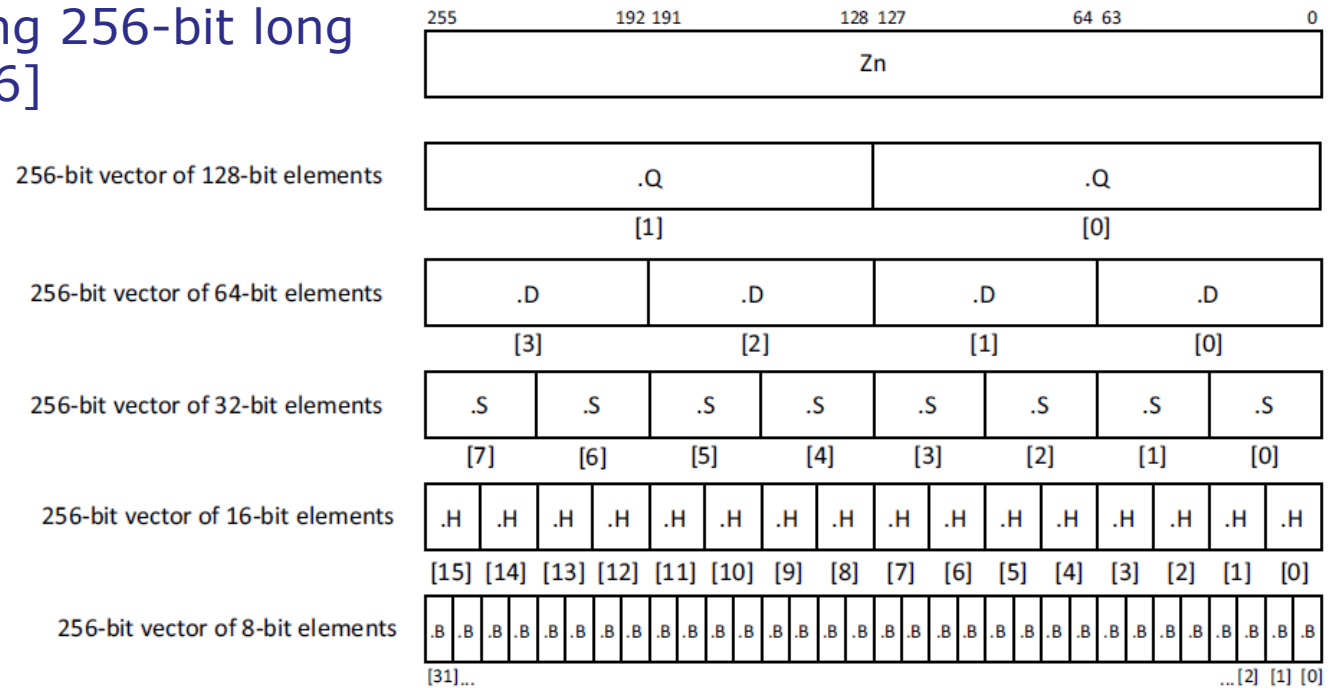
Up to 16 x128-bit wide SIMD data

FX8/16/32/64

FP16³/32/64 operations

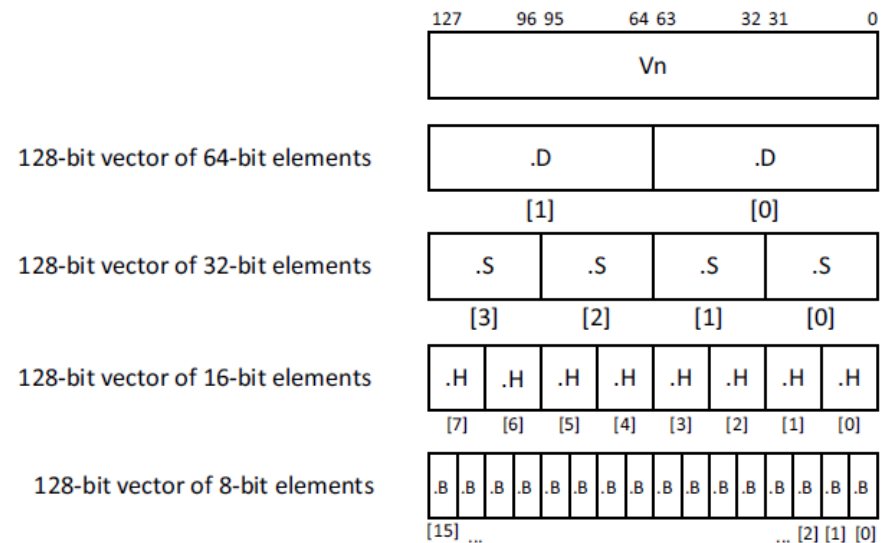
2.2.1 Overview (17)

Data types assuming 256-bit long SVE registers [116]



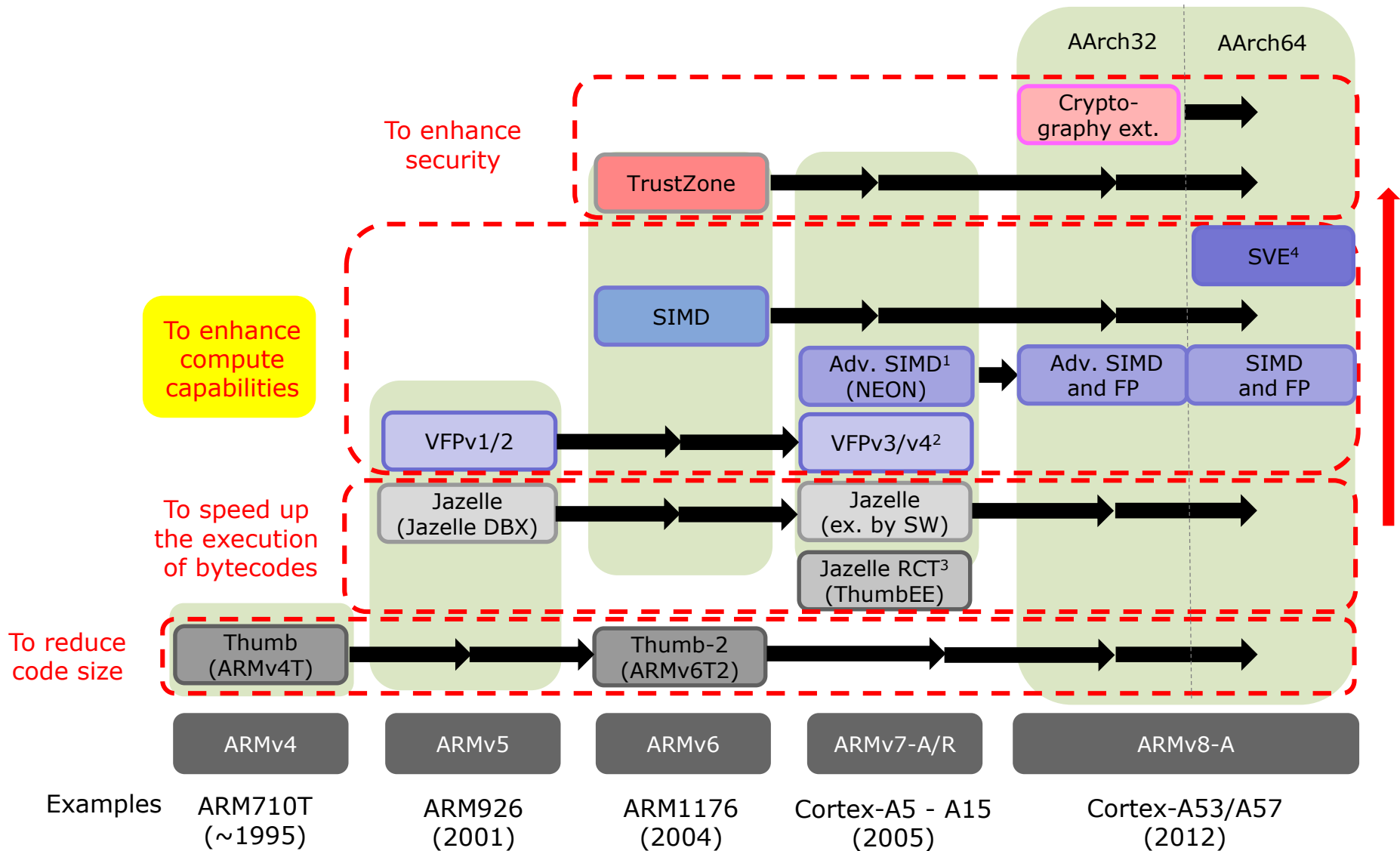
Possible data types

- 8-bit: FX
- 16-bit: FX/FP
- 32-bit: FX/FP
- 64-bit: FX/FP
- 128-bit: FX



2.2.1 Overview (18)

Overview of the ISA extensions introduced to enhance compute capabilities



Remarks: See on the slide 2.1 Overview (5).

2.2.2 The GPR register set based SIMD extension

Neither this nor all subsequent 2.2.x Sections will be discussed.

2.2.2 The GPR register set based SIMD extension

ARM's ISA extensions to enhance compute capabilities

```
graph TD; Root[ARM's ISA extensions to enhance compute capabilities] --- B1[The GPR register set based SIMD extension]; Root --- B2[A secondary register set added with FP and SIMD extensions]; Root --- B3[An SVE register set added with SIMD extensions];
```

The GPR register set based SIMD extension

It supports
FX-SIMD operations

ARMv6

Section 2.2.2

A secondary register set added with FP and SIMD extensions

It supports
FX/FP scalar and SIMD operations

ARMv5

Section 2.2.3

An SVE register set added with SIMD extensions

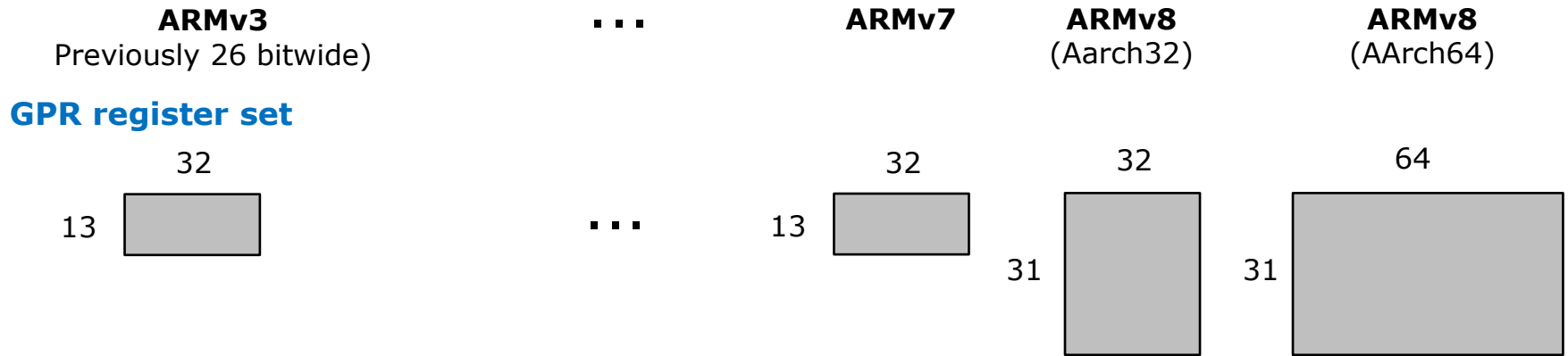
It supports
FX/FP SIMD operations
on long vector data

ARMv8.2

Section 2.2.4

2.2.2 The GPR register set based SIMD extension (2)

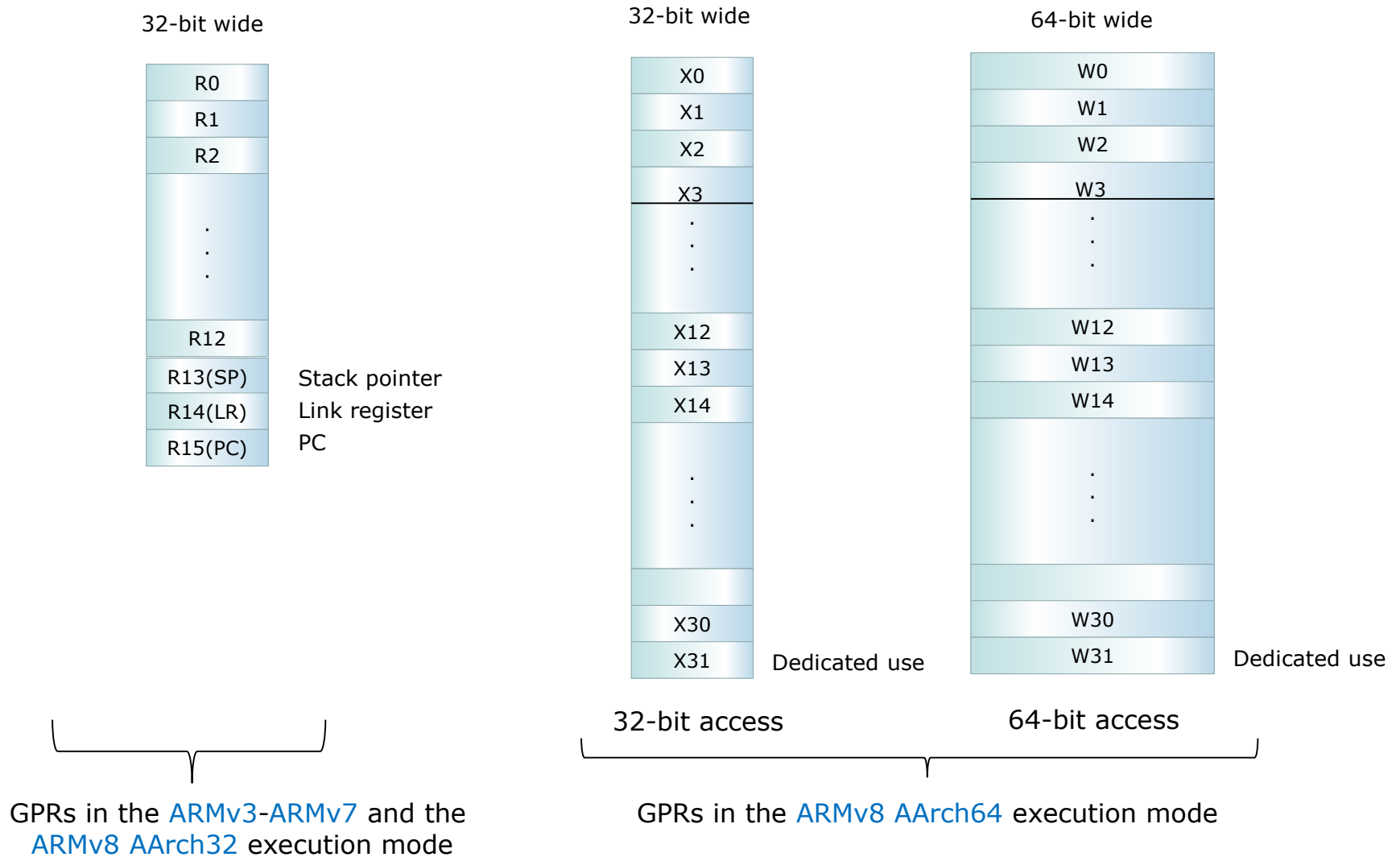
Evolution of the GPR register set in the ARM ISA -1



2.2.2 The GPR register set based SIMD extension (3)

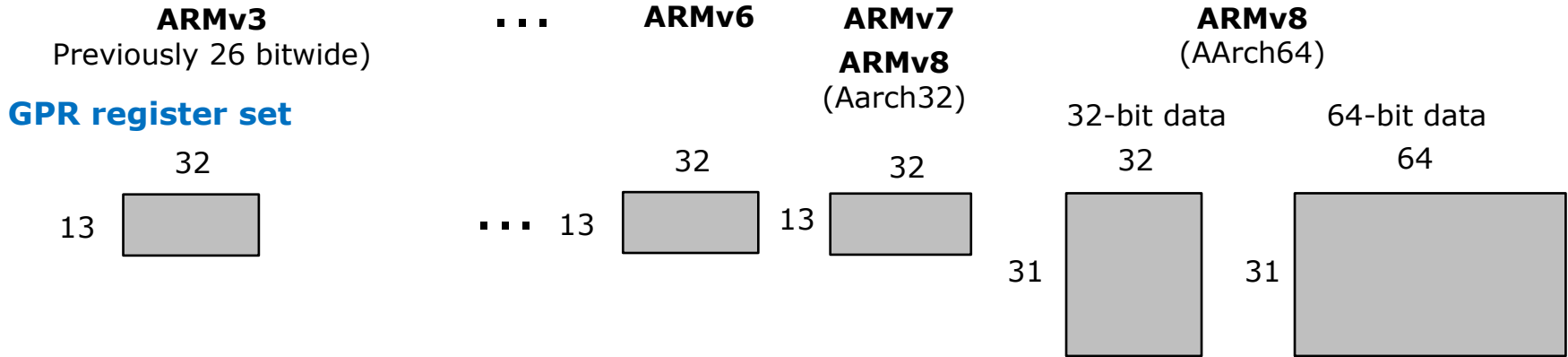
The GPR register sets in the ARMv3-ARMv7 and the ARM v8 ISA [63], [66]

In the AArch64 mode the ARMv8 ISA version expands the **number of GPRs** from **13 32-bit registers** to **31 64-bit wide registers**, as shown in the next Figure.



2.2.2 The GPR register set based SIMD extension (4)

Introduction of the GPR register set based SIMD extension in the ARMv6 ISA



Scalar data

FX32

Basic ISA

FX32

FX64

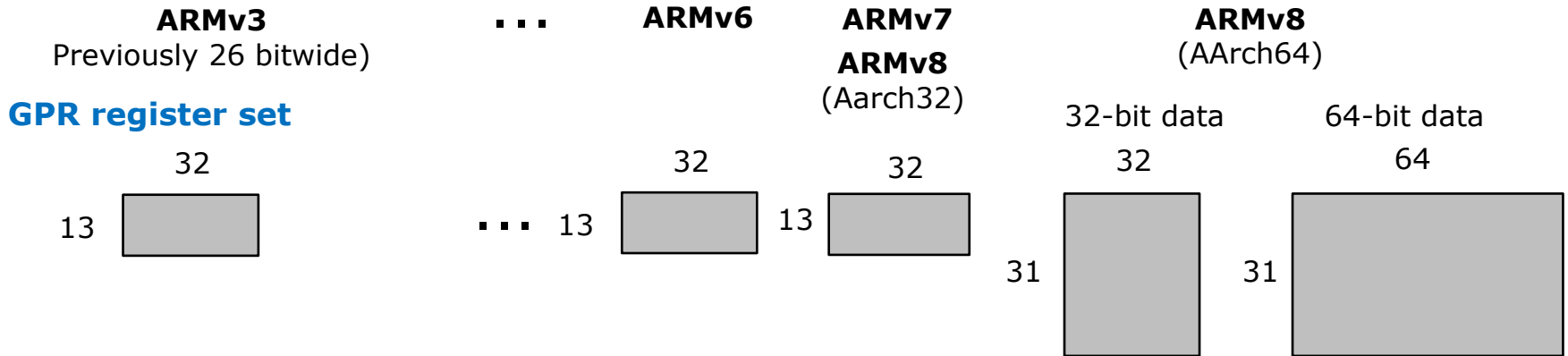
**SIMD
extension**

SIMD data

32-bit wide FX-SIMD
FX8/16 operations

2.2.2 The GPR register set based SIMD extension (4b)

Introduction of the GPR register set based SIMD extension in the ARMv8 ISA



Scalar data

FX32

Basic ISA

FX32

FX64

SIMD extension

SIMD data

32-bit wide FX-SIMD
FX8/16 operations

SIMD data

32-bit wide FX-SIMD
FX8/16 operations

SIMD data

64-bit wide FX-SIMD
FX8/16/32 operations

The GPR register set based SIMD extension

- It supports **FX-SIMD operations** on 32-bit wide SIMD data in the GPR registers.
- Available instructions perform **operations on 4xFX8 or 2xFX16 data in parallel**.
- It is similar to Intel' s MMX x86 ISA extension from 1997.
- The GPR register set based SIMD extension as introduced into the ARMv6 ISA version provides only a **modest performance boosting potential**.

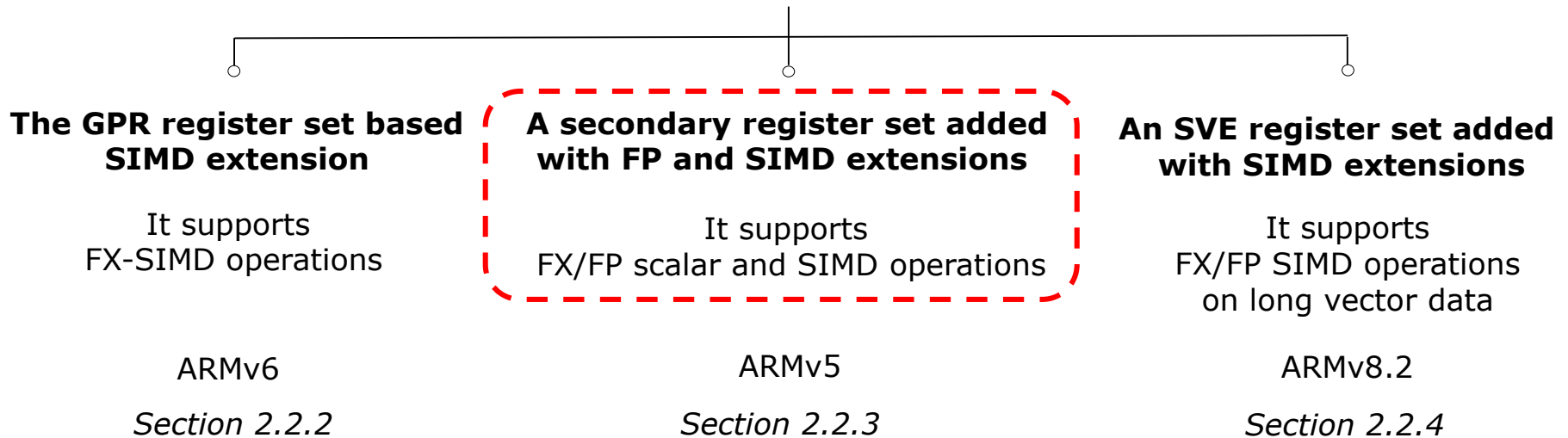
By contrast the subsequently, in the ARMv7 ISA version introduced, secondary register set based **advanced SIMD (NEON) extension** has a **much higher performance boosting potential**.

2. 2.2.3 Secondary register set based FP and SIMD extensions

2.2.3 Secondary register set based FP and SIMD extensions

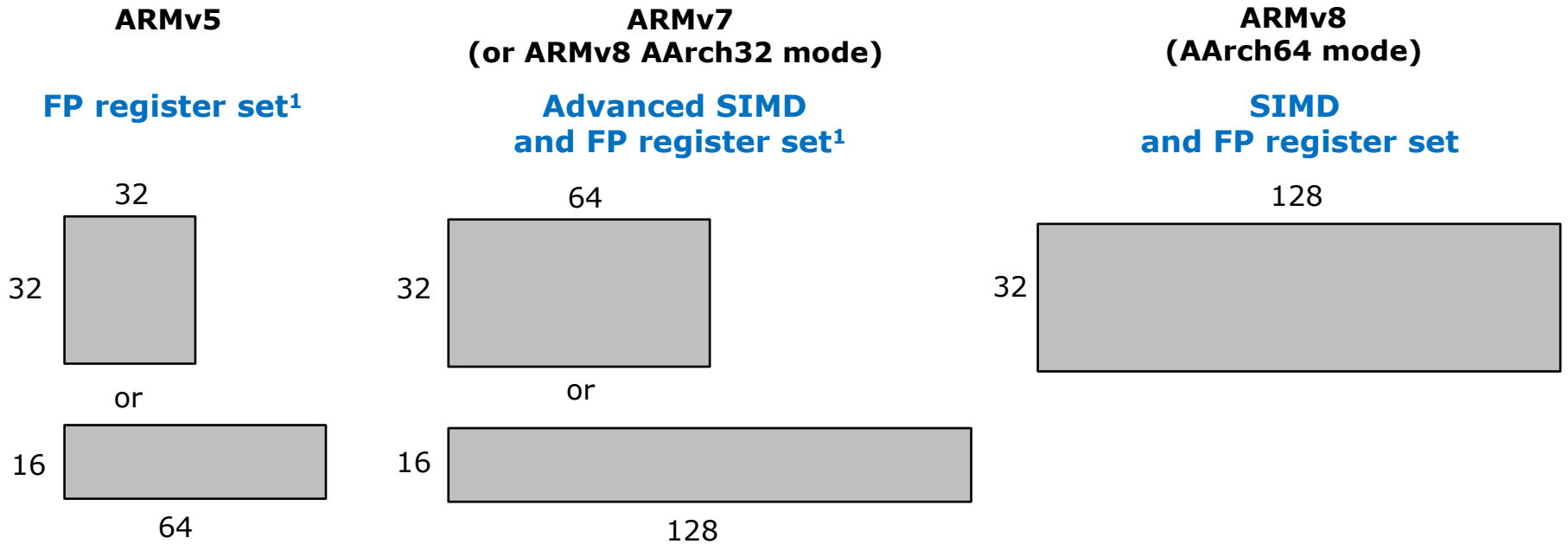
2.2.3.1 Secondary register set based FP and SIMD extensions - Overview

ARM's ISA extensions to enhance compute capabilities



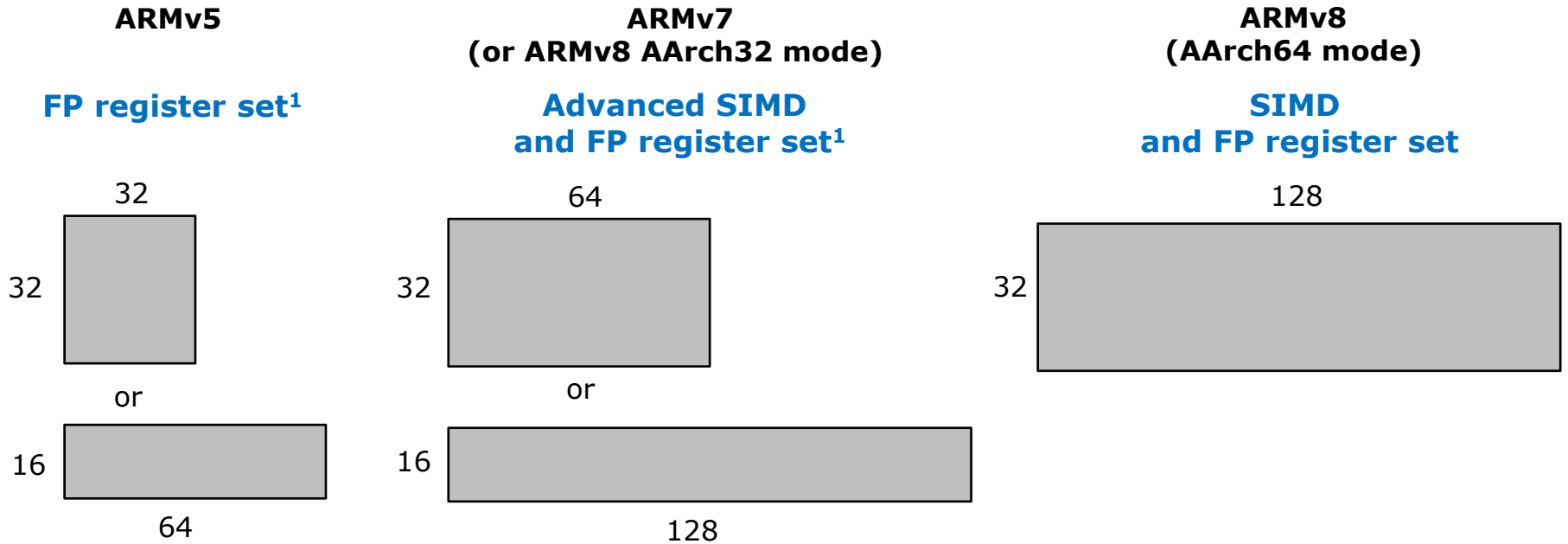
2.2.3.1 Secondary reg. set based FP and SIMD extensions - Overview (2)

Designation and size of the secondary register set in the ARMv5 - ARMv8 ISAs



2.2.3.1 Secondary reg. set based FP and SIMD extensions - Overview (3)

The secondary register set based FP and SIMD extensions - Overview



VFPv1/v2 extension

Scalar data

FP32/64

Serially processed FP vector data

Up to 8xFP32 or 4xFP64
operations serially executed

VFPv3/v4 extension²

Scalar data

FP16¹/32/64

Serially processed FP vector data

Up to 8xFP32 or 4xFP64
operations serially executed

SIMD and FP extension

Scalar data

FP16¹/32/64

SIMD (vector) data

64/128-bit wide SIMD data
FX8/16/32/64
FP16³/32/64 operations

Advanced SIMD (NEON) extension

SIMD data

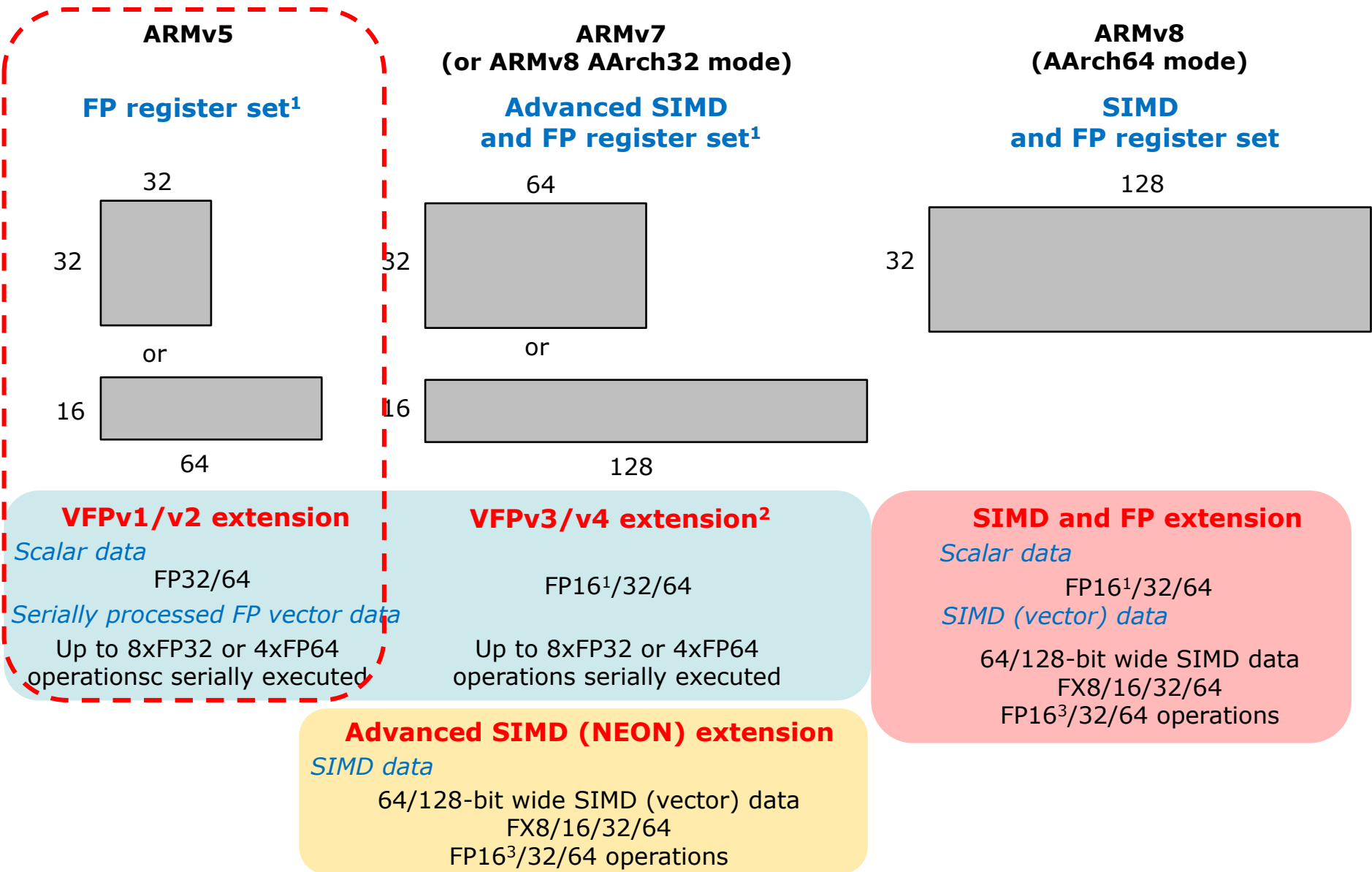
64/128-bit wide SIMD (vector) data
FX8/16/32/64
FP16³/32/64 operations

Remarks

- ¹Certain models implement only half of the specified register numbers.
- ²The implementation of the VFP3 or VFP4 extensions presumes the implementation of the Advanced SIMD (NEON) extension.
- ³In the VFP3/VFP4 extensions only conversions between FP16 and FP32 or FP64 are supported.

2.2.3.2 The VFPv1/v2 extensions (1)

2.2.3.2 The VFPv1/v2 extensions



2.2.3.2 The VFPv1/v2 extensions (2)

The FP register set (as introduced in the ARMv5) [65]

- The new secondary register set is 32 x 32-bit or 16x64-bit wide and is organized as **four register banks**, each including 8 registers, as seen below.

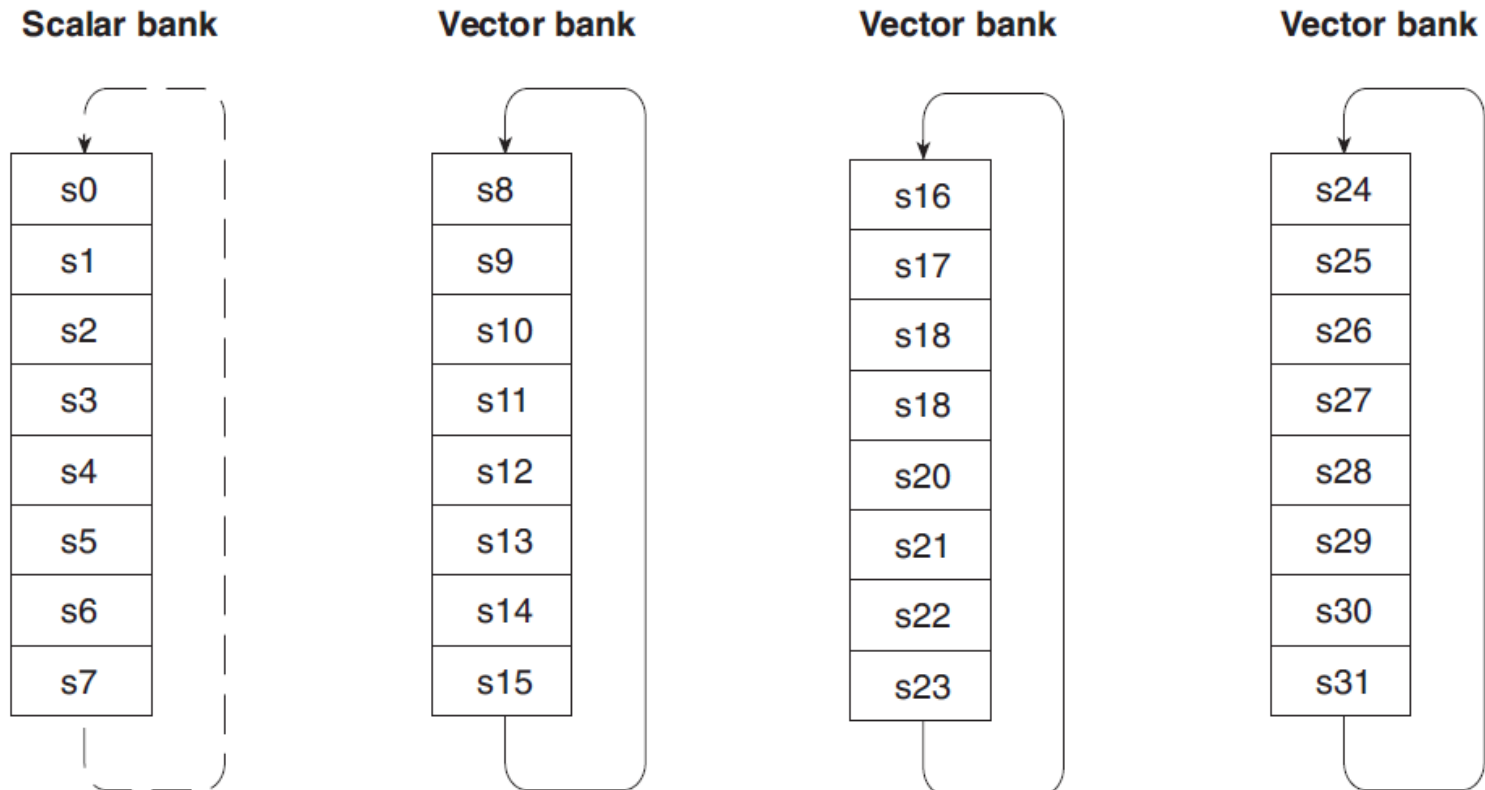


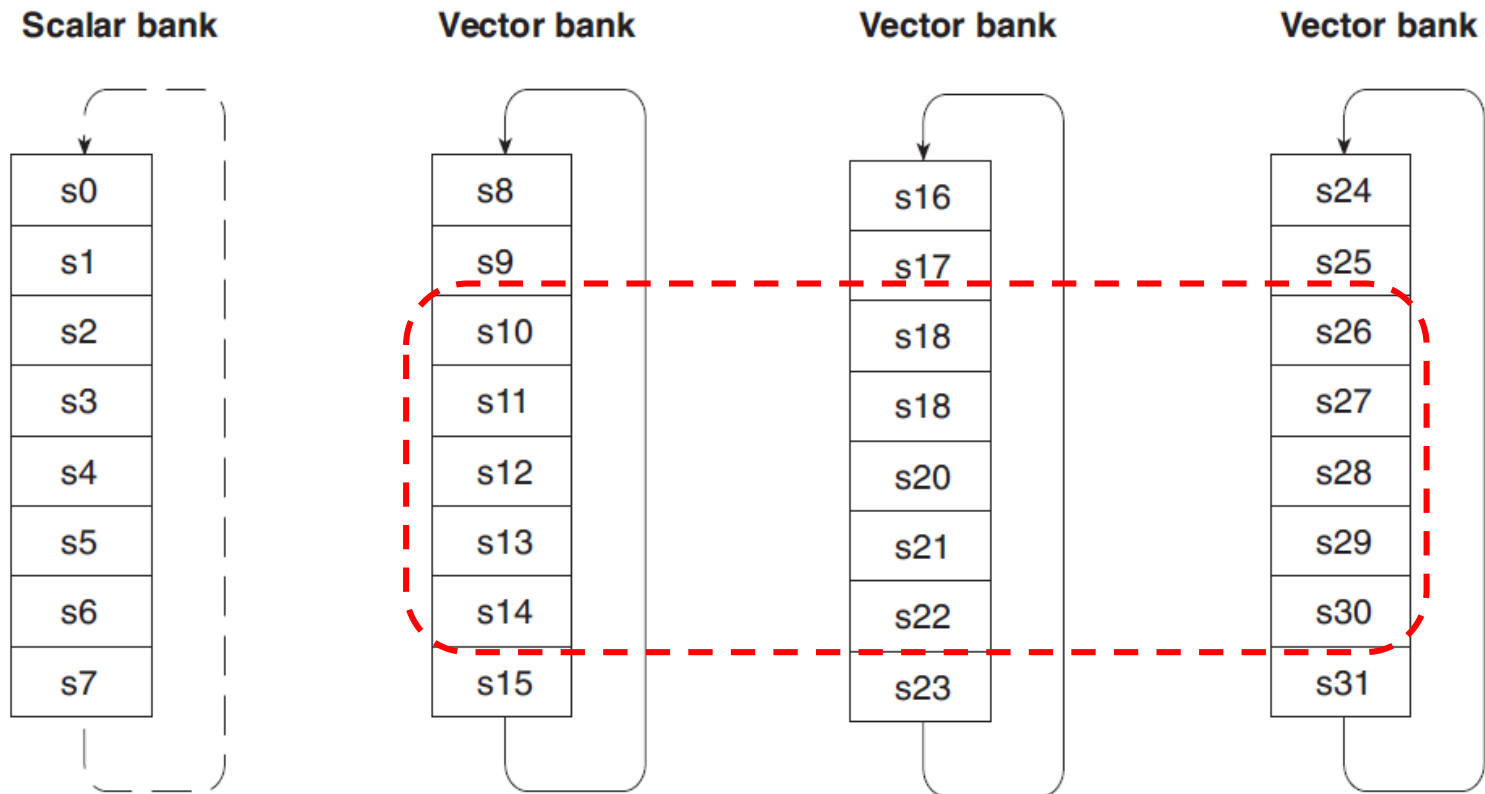
Figure: Register banks of the VFP1 and VFP2 extensions [65]

Use of the FP register set [65]

- The **first bank** is used to hold **scalar operands** whereas to **remaining three banks** **vector operands**.
- **FP vector operands** may refer to **2 to 8 registers** from the same bank.
- The **vector length** is given **in a specific field of a control register**.
- The **register numbers** given **in the instruction specify the first registers** that contain the first operands and specify the first destination register.
Each successive element of the vector is taken by incrementing appropriately the register numbers.
- The peculiarity of the VFP extension is that **the elements of the vector are processed sequentially** rather than parallel as for usual SIMD execution.

2.2.3.2 The VFPv1/v2 extensions (4)

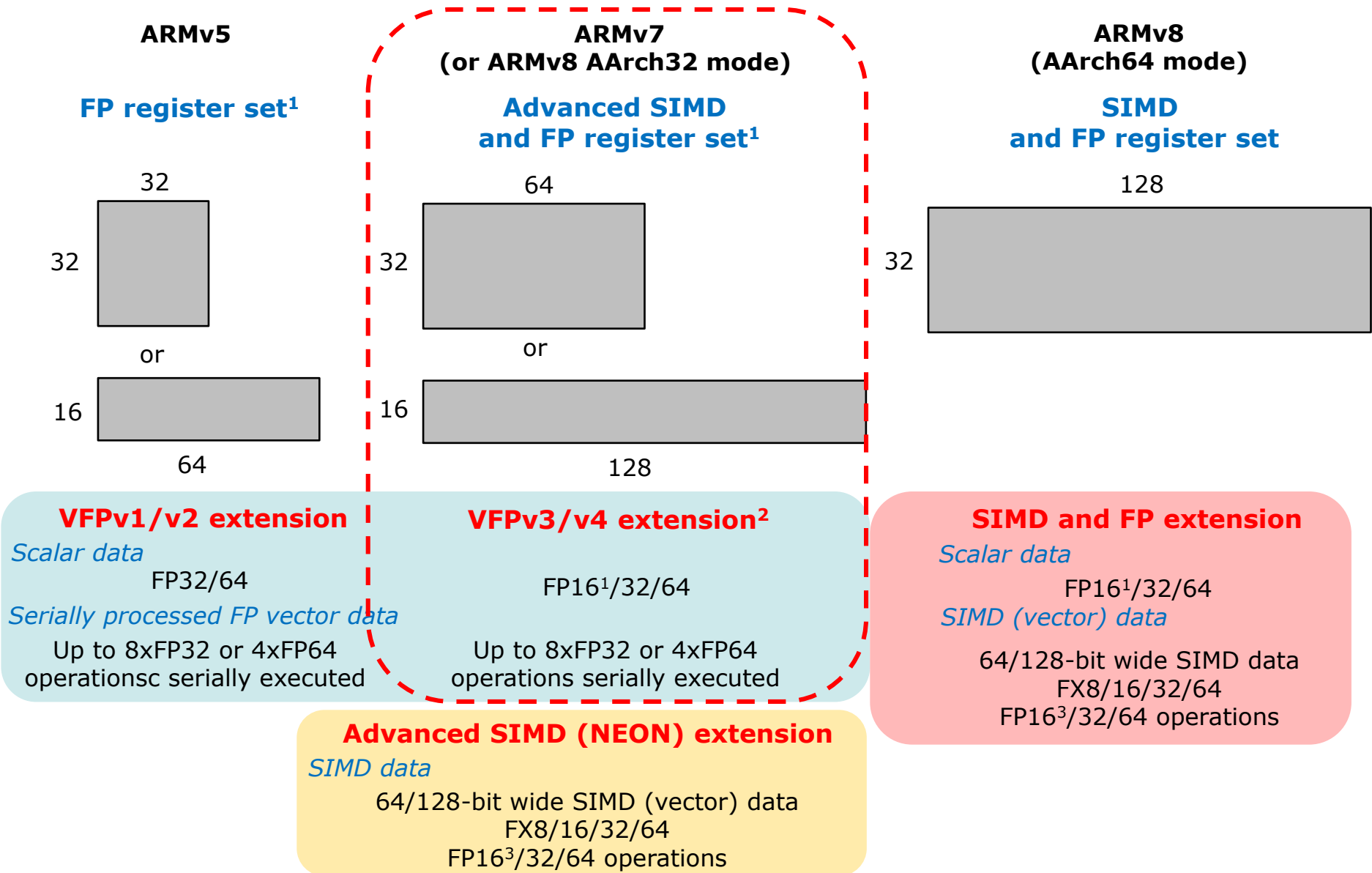
Example: Serially processed 5-element vector operation [65]



- In the example the **input operands** are taken from the registers s10...s14 and s18...s22, and the **result** is written into the registers s26...s30.
- The **execution is sequential** (like a hardware implemented subroutine).

2.2.3.3 The VFPv3/v4 extensions (1)

2.2.3.3 The VFPv3/v4 extensions -1



The VFPv3 and VFPv4 extensions -2

- The underlying register set (called the **Advanced SIMD and FP register set**) is an extension of the previous FP register set and is shared by the NEON (Advanced SIMD and FP) extension.

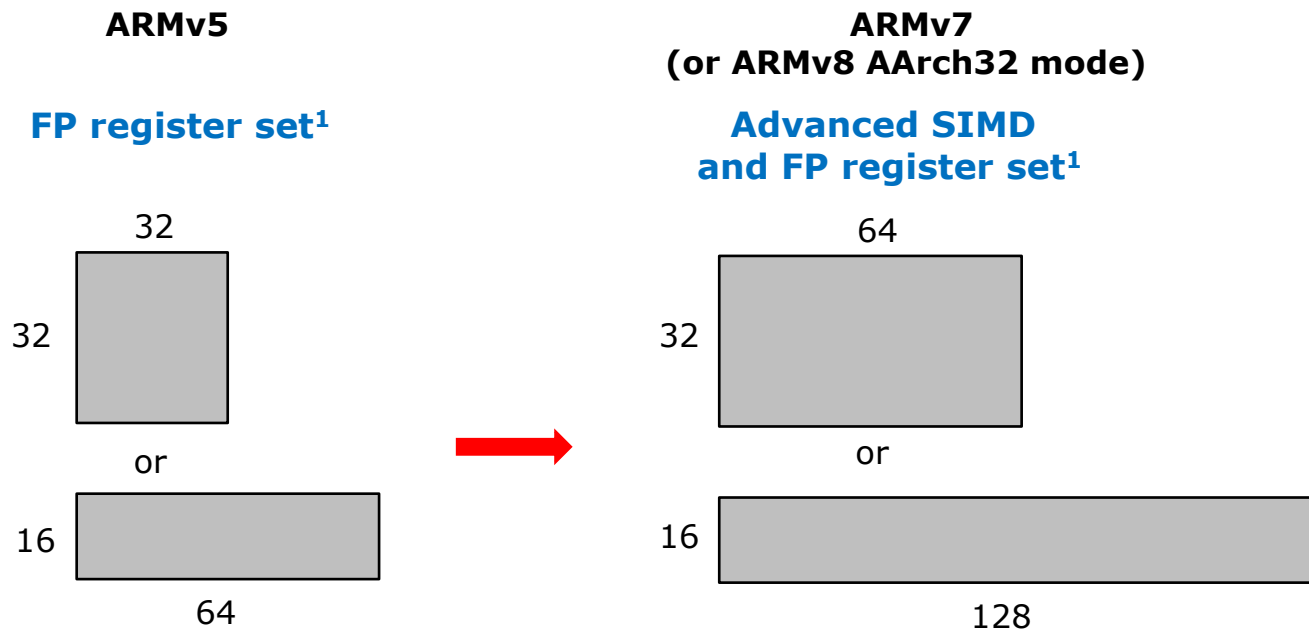


Figure: Extension of the secondary register set in the ARMv7 ISA

- It follows that the implementation of the VFPv3 or VFPv4 extensions presume the implementation of the Advanced SIMD (Neon) extension.

The VFPv3 and VFPv4 extensions -3

- The VFPv3/v4 extensions support basically the same operations as the previous VFPv2 extension.
- The main enhancements of VFPv3/v4 are:
 - VFPv3 support in addition FP16 (half-word FP) operations, nevertheless the supported operations are restricted only to conversions between FP16 and FP32 or FP64 data types.
 - VFPv4's main enhancement is support of FMA (Fused Multiply Add) operations.

2.2.3.3 The VFPv3/v4 extensions (4)

Contrasting main features of the VFPv1/v2 and VFPv3/v4 extensions

VFP (Vector Floating Point) extensions



VFPv1/v2

Introduced in the ISA version
ARMv5

- introduction of an **FP register set** (32x32 or 16x64 bit)
- supporting operations on
 - **scalar FP data** (FP32/FP64) and
 - **serially processed FP vector data** (up to 8xFP32 or 4xFP64)

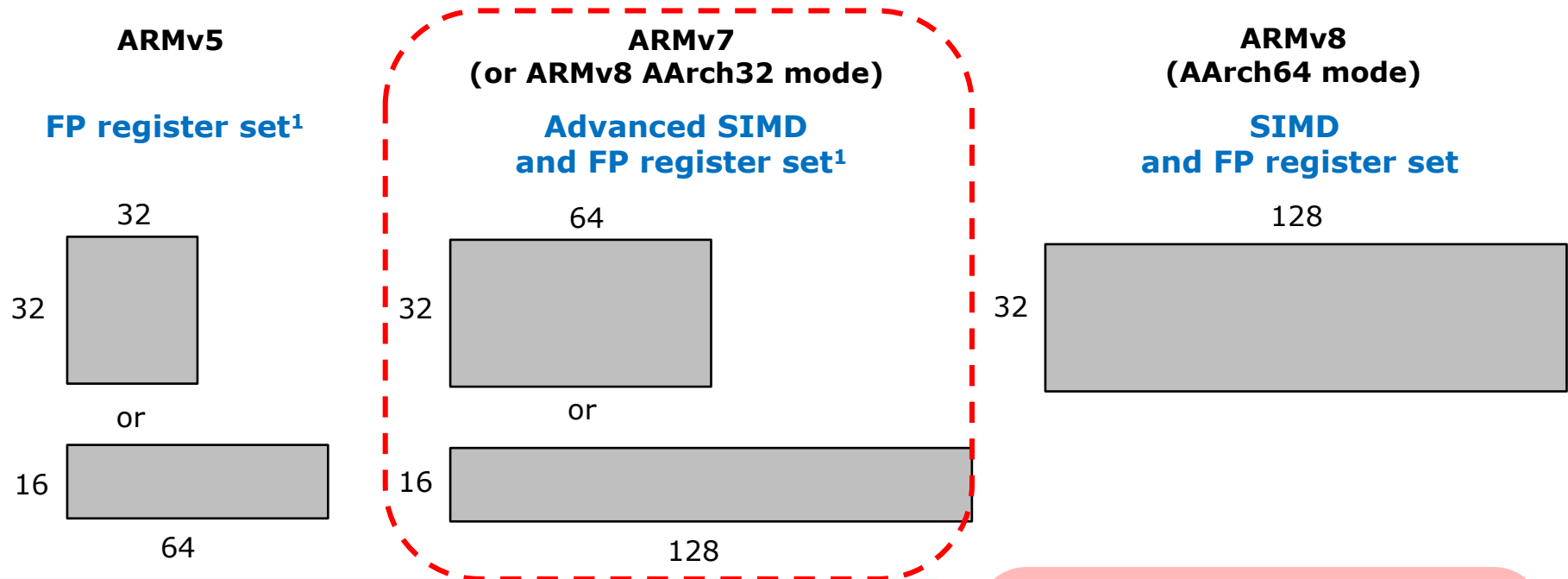
VFPv3/v4

Introduced in the ISA version
ARMv6

- **doubling the size of the FP register set to 32x64 or 16x128 bit registers**, and
- supporting operations as in case of the previous vFPv2 version plus
 - providing instructions that perform **conversions between FP16 and FP32 or FP64 data** (since the VFP3) and
 - **additionally FMA operations** (in the VFP4 extension).

2.2.3.4 Advanced SIMD (NEON) extension (1)

2.2.3.4 Advanced SIMD (NEON) extension -1



VFPv1/v2 extension

Scalar data

FP32/64

Serially processed FP vector data

Up to 8xFP32 or 4xFP64 operations serially executed

VFPv3/v4 extension²

FP16¹/32/64

Up to 8xFP32 or 4xFP64 operations serially executed

SIMD and FP extension

Scalar data

FP16¹/32/64

SIMD (vector) data

64/128-bit wide SIMD data
FX8/16/32/64
FP16³/32/64 operations

Advanced SIMD (NEON) extension

SIMD data

64/128-bit wide SIMD (vector) data
FX8/16/32/64
FP16³/32/64 operations

2.2.3.4 Advanced SIMD (NEON) extension (2)

2.2.3.4 The Advanced SIMD (NEON) extension of the ARMv7 -1

- NEON shares its register set with the VFPv3/v4 extension.
- The shared register set is called the **Advanced SIMD and FP register set**. It is an extension of the previous FP register set, as shown in the Figure below.

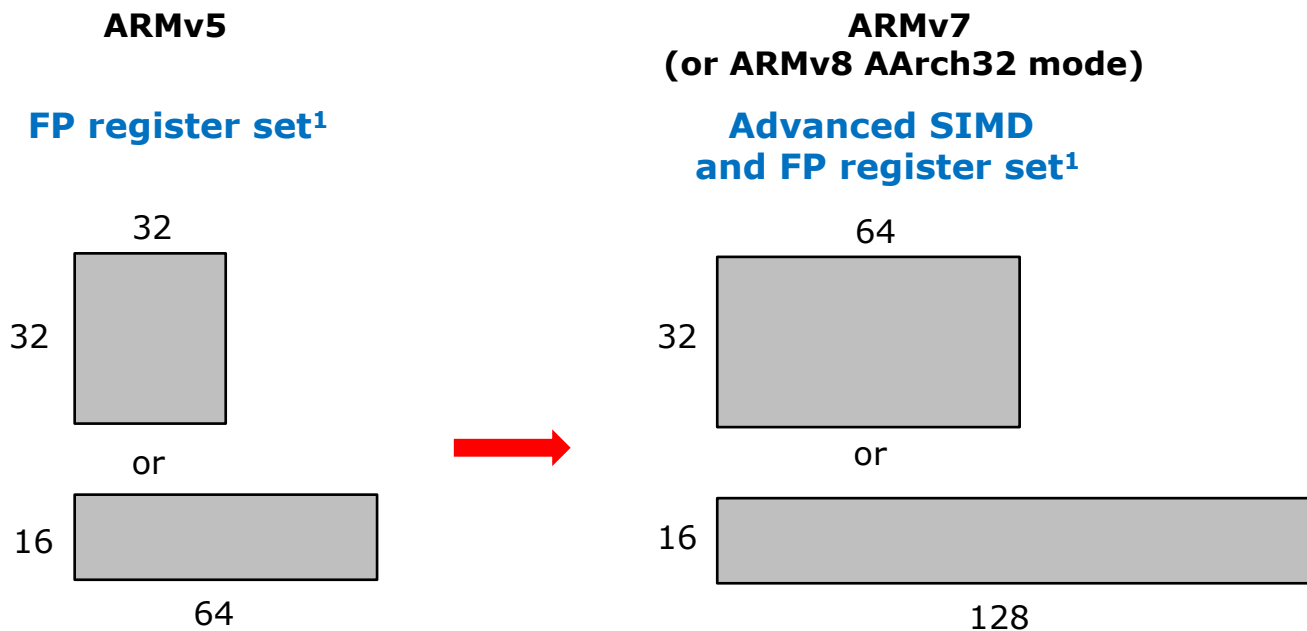


Figure: Extension of the secondary register set in the ARMv7 ISA

2.2.3.4 The Advanced SIMD (NEON) extension of the ARMv7 [11] -2

- NEON is intended to accelerate multimedia and signal processing algorithms such as video encode/decode, 2D/3D graphics, gaming, speech or image processing.

2.2.3.4 Advanced SIMD (NEON) extension (4)

The Advanced SIMD (NEON) extension of the ARMv7 ISA [11] -2

- **NEON instructions** operate on 64 or 128-bit wide vector data (SIMD data) held in the Advanced SIMD and FP register set.
- They perform the **same operation** on all data elements, as indicated below.

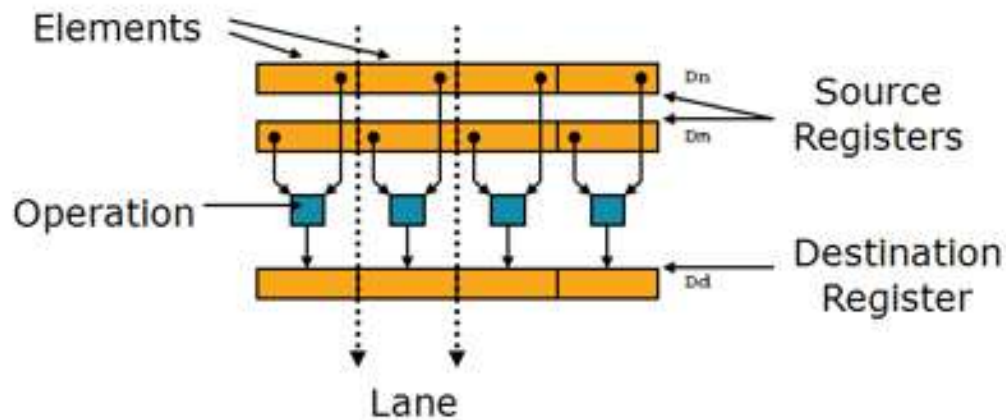


Figure: SIMD data [11]

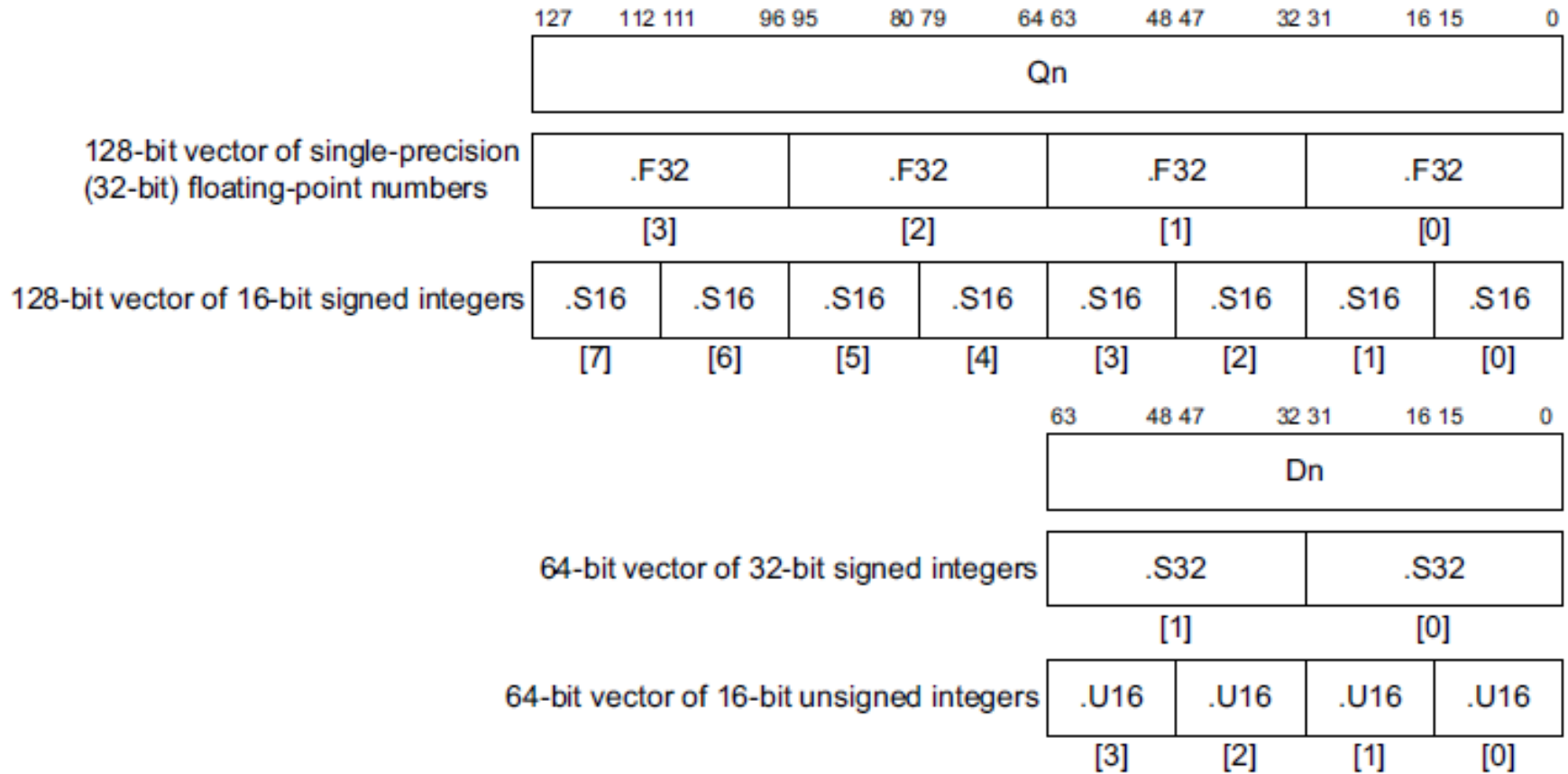
- **NEON instructions** operate on 64 or 128-bit wide vectors with
 - FX8/FX16/FX32/FX64 or
 - FP16/FP32/FP64data elements.

Remark

- For FP16 data NEON supports only conversions between FP16 and FP32/FP64 data.
- FP16 data operations are supported only in the Advanced SIMDv2 option.

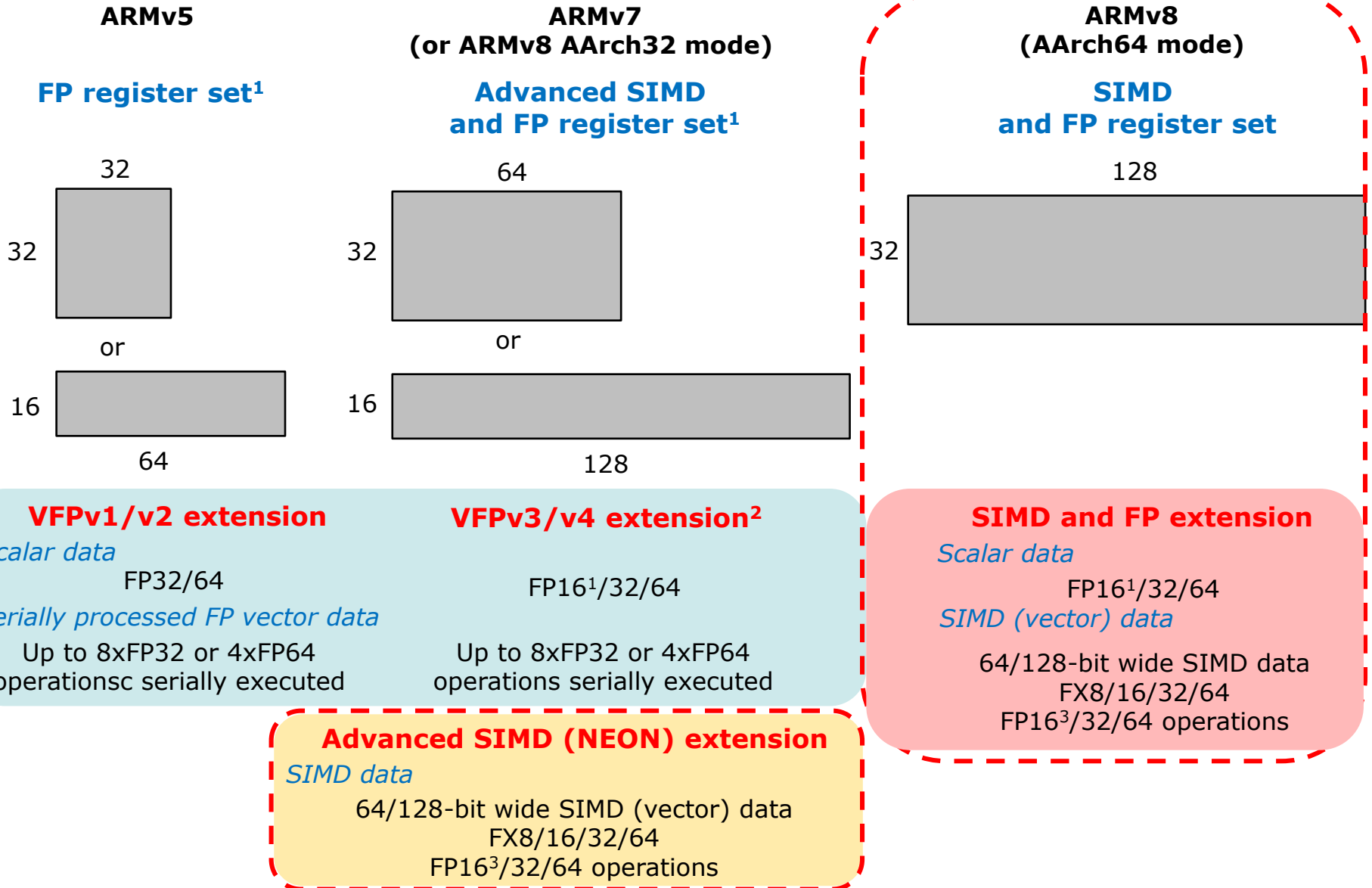
2.2.3.4 Advanced SIMD (NEON) extension (5)

Example vector data formats of the Advanced SIMD (NEON) extension [86]



2.2.3.5 The SIMD and FP extension of the ARMv8 ISA in AArch64 mode (1)

2.2.3.5 The SIMD and FP extension of the ARMv8 ISA in the AArch64 mode-1



2.2.3.5 The SIMD and FP extension of the ARMv8 ISA in AArch64 mode (2)

The SIMD and FP extension of the ARMv8 ISA in the AArch64 mode -2

- The ARMv8 ISA version introduces **major changes in** the ARM architecture while maintaining a high level of consistency with previous versions of the architecture.
- **ARMv8** has **two distinct execution modes**, as indicated below.

Execution modes of the ARMv8

```
graph TD; A[Execution modes of the ARMv8] --- B[Aarch32 execution mode]; A --- C[Aarch64 execution mode];
```

Aarch32 execution mode

It supports two 32-bit instruction sets, the **A32** and the **T32** (Thumb) instructions sets.

In this mode the processor can run programs developed for previous ISA versions.

Aarch64 execution mode

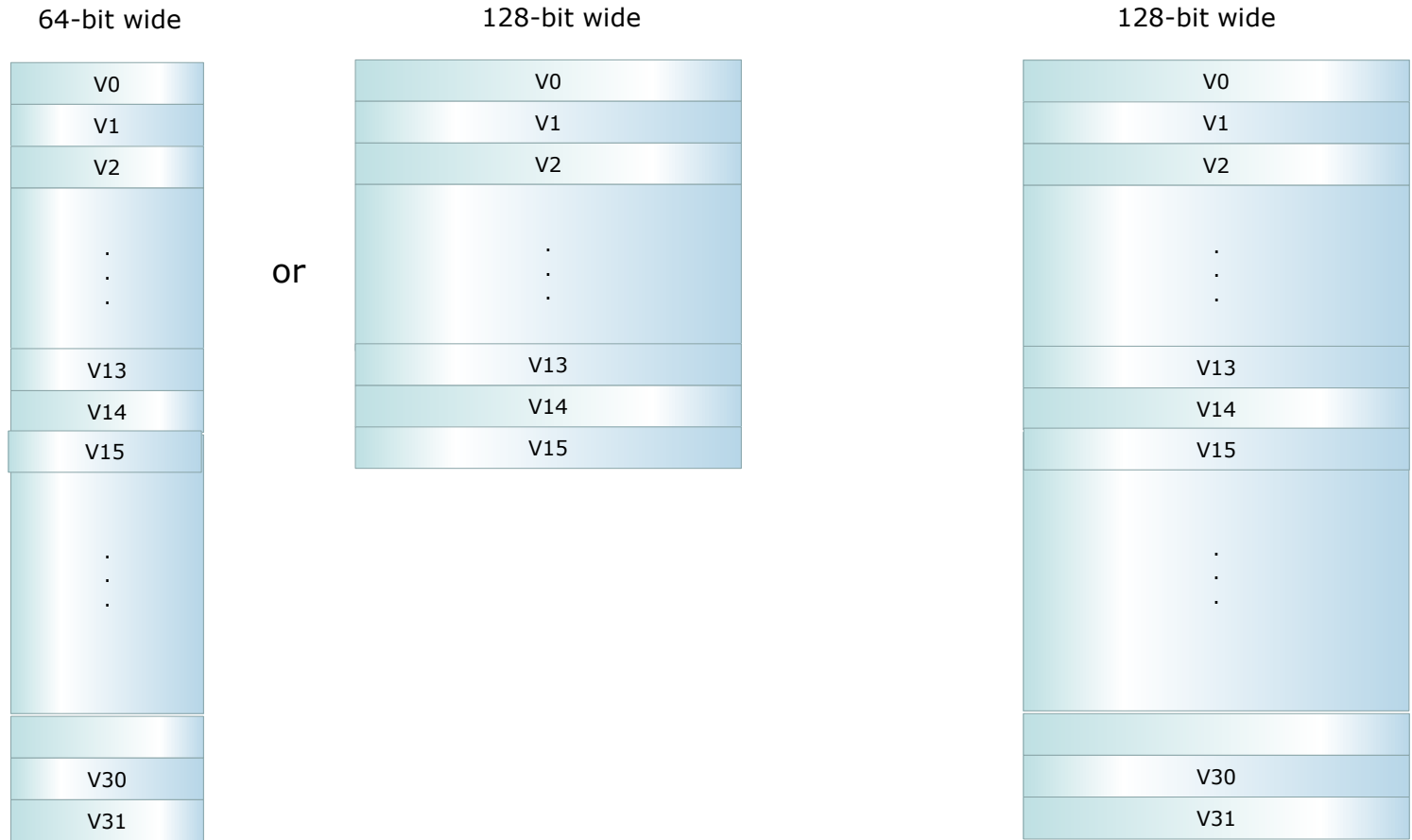
It supports a single 64-bit instruction set, called **A64**.

This is a fixed length powerful instruction set that uses 32-bit instruction encodings.

2.2.3.5 The SIMD and FP extension of the ARMv8 ISA in AArch64 mode (3)

Extension of the secondary register set in the ARMv8 ISA [66]

The AArch64 mode the ARMv8 ISA expands the number of the secondary registers from 32x64-bit or 16x64-bit (ARMv7 ISA) to 32x128-bit, as seen below.

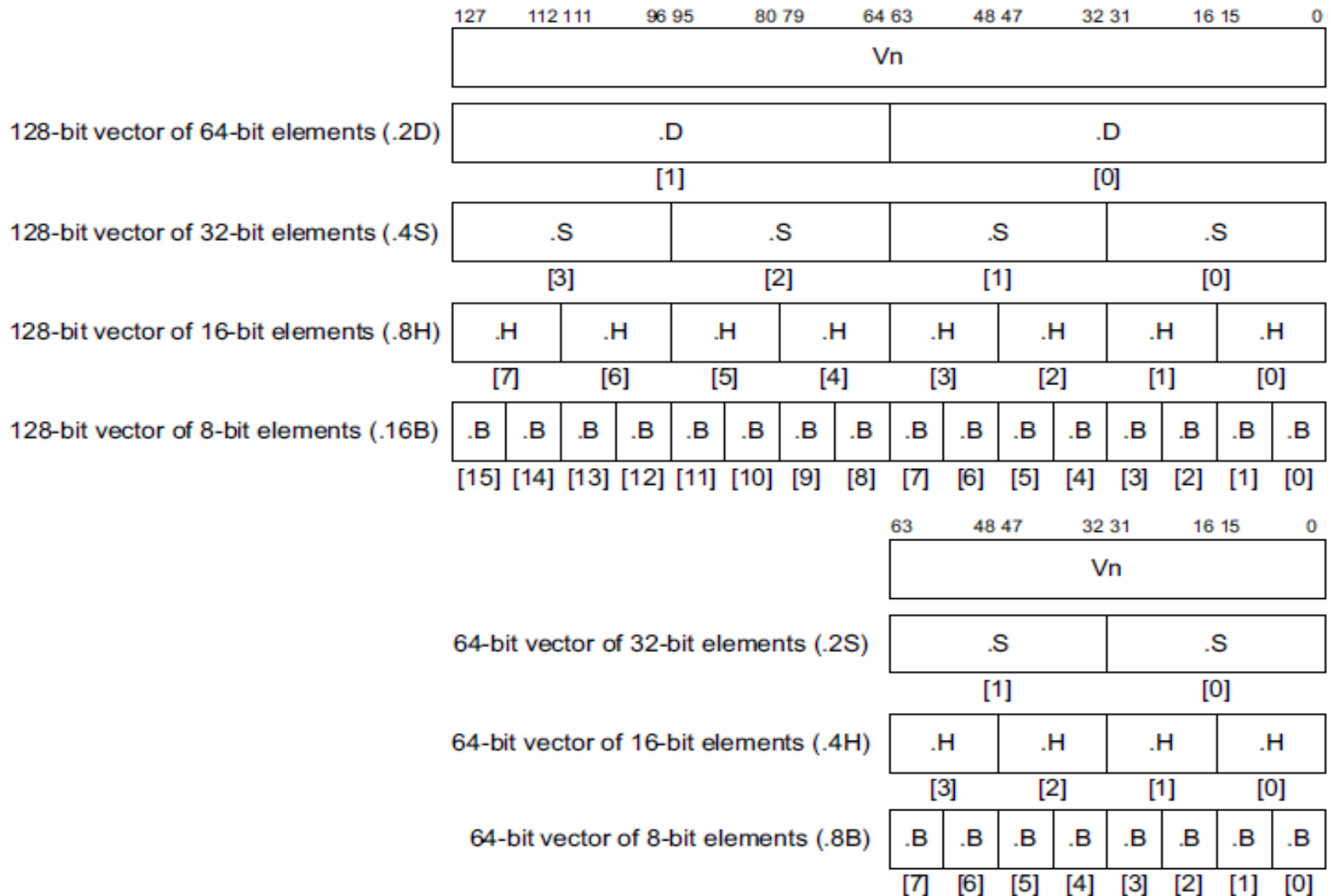


FP and advanced SIMD registers in the ARMv7 and ARMv8 AArch32 execution mode of the ARM ISA

SIMD and FP registers in the ARMv8 AArch64 execution mode of the ARM ISA

2.2.3.5 The SIMD and FP extension of the ARMv8 ISA in AArch64 mode (4)

Use cases of 64- and 128-bit wide SIMD data in the AArch64 mode [66]

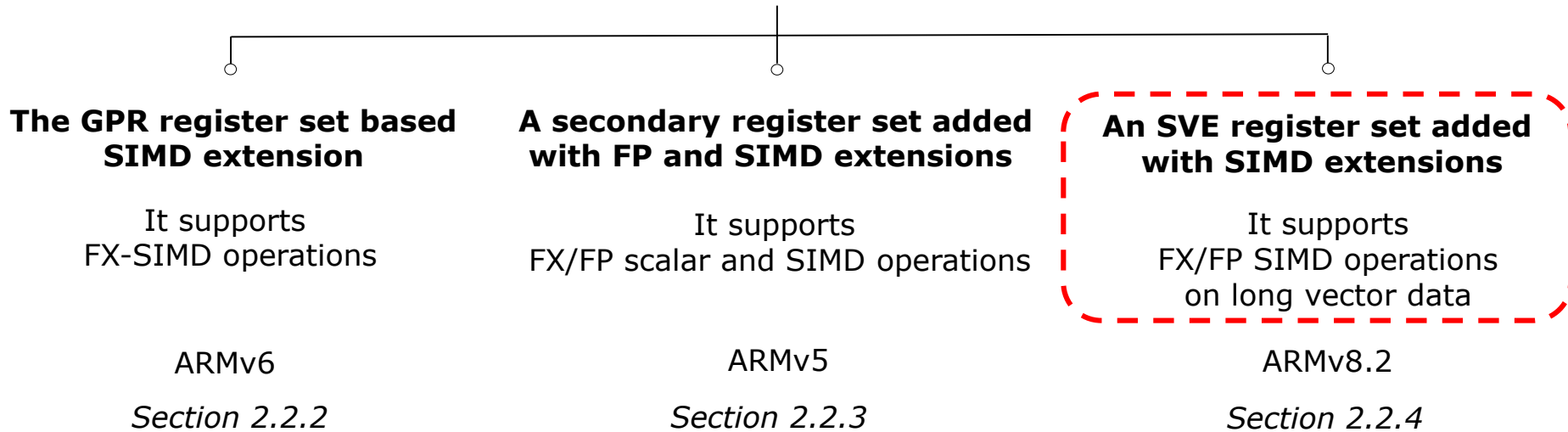


2. 2.2.4 The SVE register set based SVE extension

2.2.4 The SVE register set based SVE extension (1)

2.2.4 The SVE register set based SVE extension

ARM's ISA extensions to enhance compute capabilities



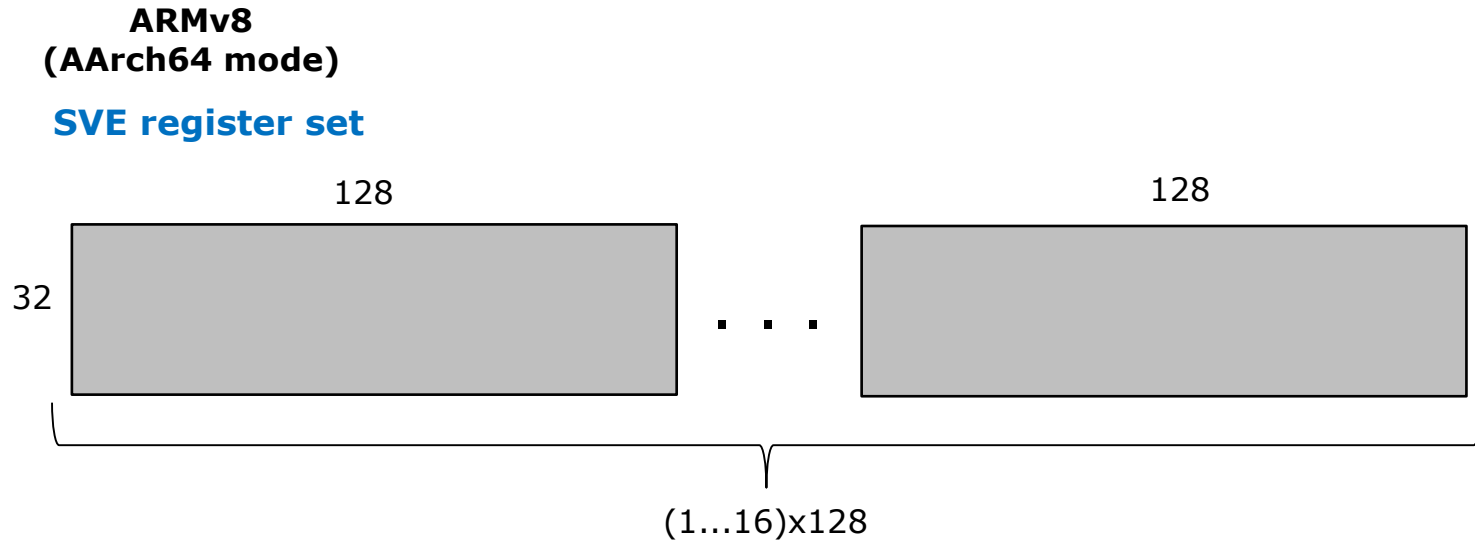
2.2.4 The SVE register set based SVE extension (2)

The SVE (Scalable Vector Extension) introduced into the AArch64 mode of the ARMv8 ISA [96]

- It was announced in [8/2016](#) (in the Hot Chips conference).
- General specification became available as the ARM Architecture Reference Manual Supplement - The Scalable Vector Extension (SVE) for ARMv8-A (β version in 3/2017).
- The [vector length](#) in SPE is a hardware choice [from 1x128 bit to 16x128 bit](#), first implementations (e.g. by Fujitsu) choose 4x128 bit = 512 bit vector length.
- SVE is implemented [only in the AArch64](#) version of ARMv8.
- SVE [aims at HPC](#) scientific workloads rather than media or image processing.
- It supports [both FX and FP processing](#).

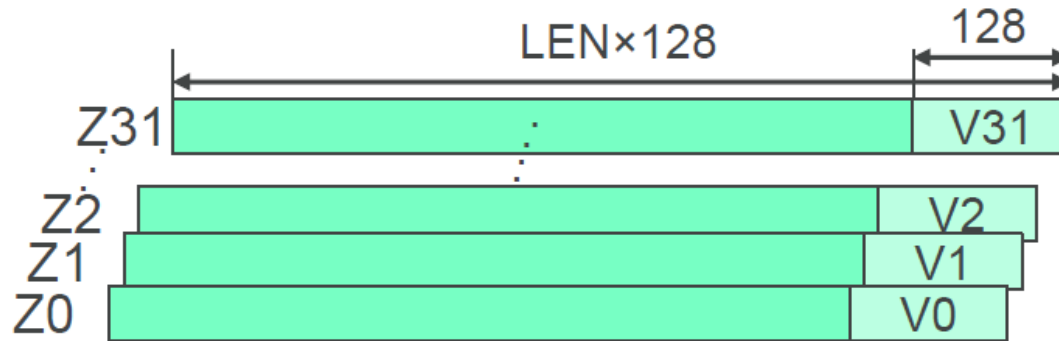
2.2.4 The SVE register set based SVE extension (3)

The SVE register set introduced in the AArch64 mode of the ARMv8 ISA



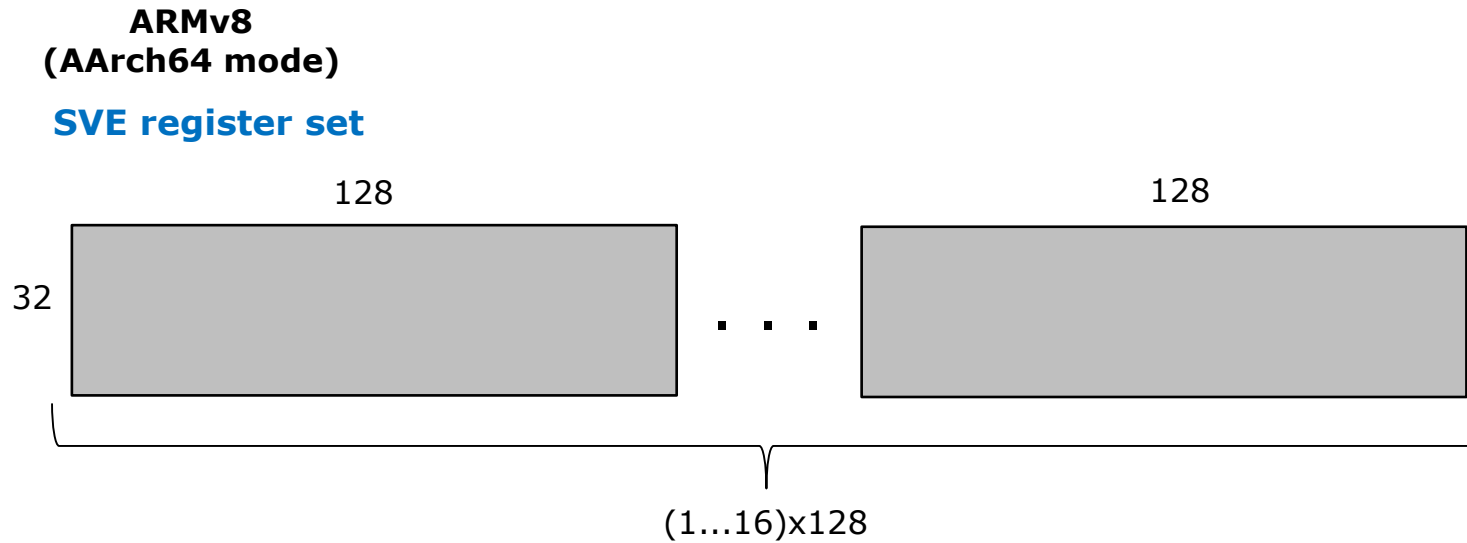
2.2.4 The SVE register set based SVE extension (4)

Available SVE registers [96]



LEN: 1 to 16

The SVE extension - Overview



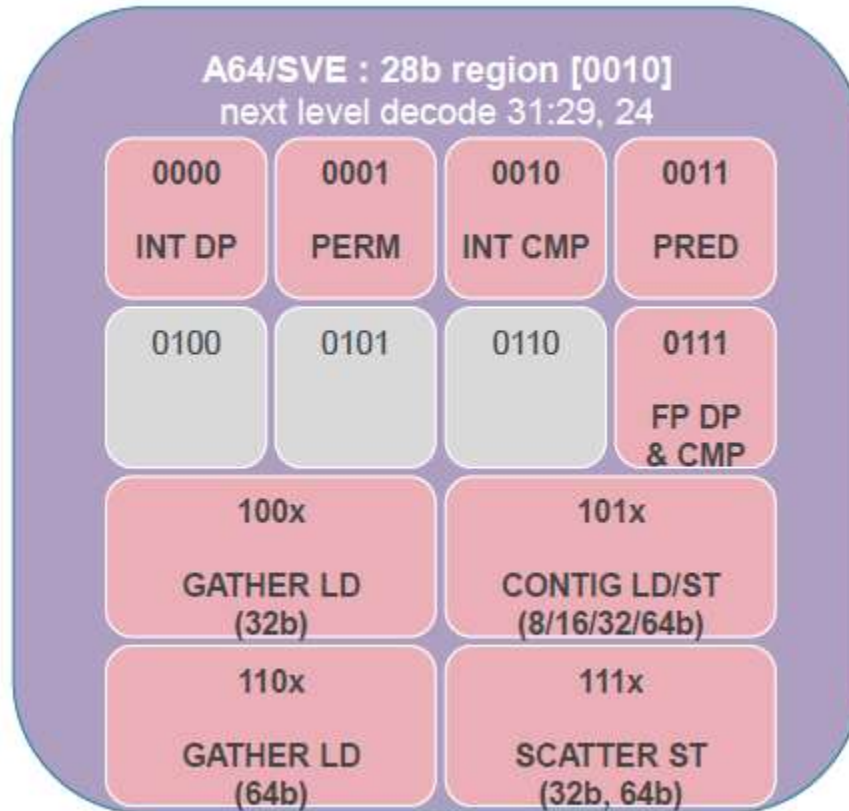
SVE extension

FX and FP datatypes
(Specified in the ARM Architecture
Reference Manual Supplement -
The Scalable Vector Extension (SVE)
for ARMv8-A)

(1...16)x 128 bit wide SIMD data
(Supported operations not yet specified)

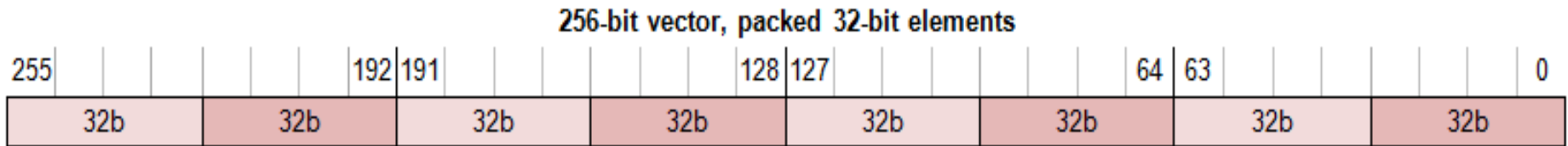
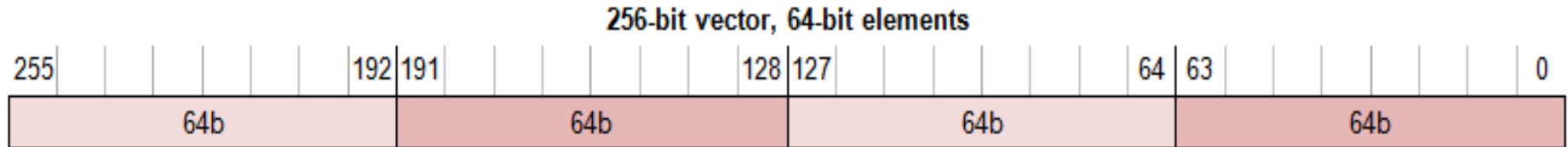
2.2.4 The SVE register set based SVE extension (6)

The SVE instruction set [96]



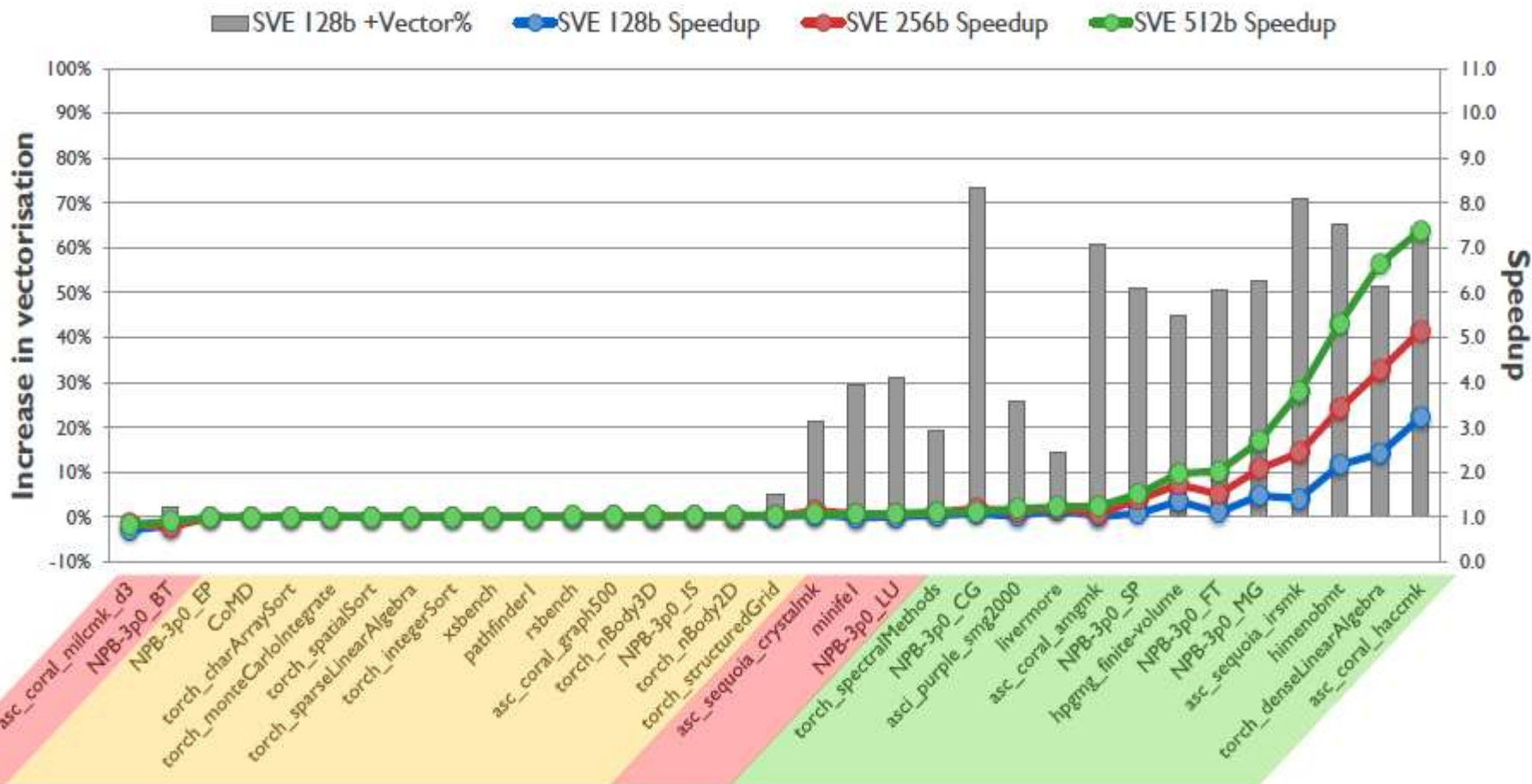
2.2.4 The SVE register set based SVE extension (7)

Example: Use of 256-bit vectors (either for FX or FP computations) [96]



2.2.4 The SVE register set based SVE extension (8)

Speed-up potential of SVE [96]



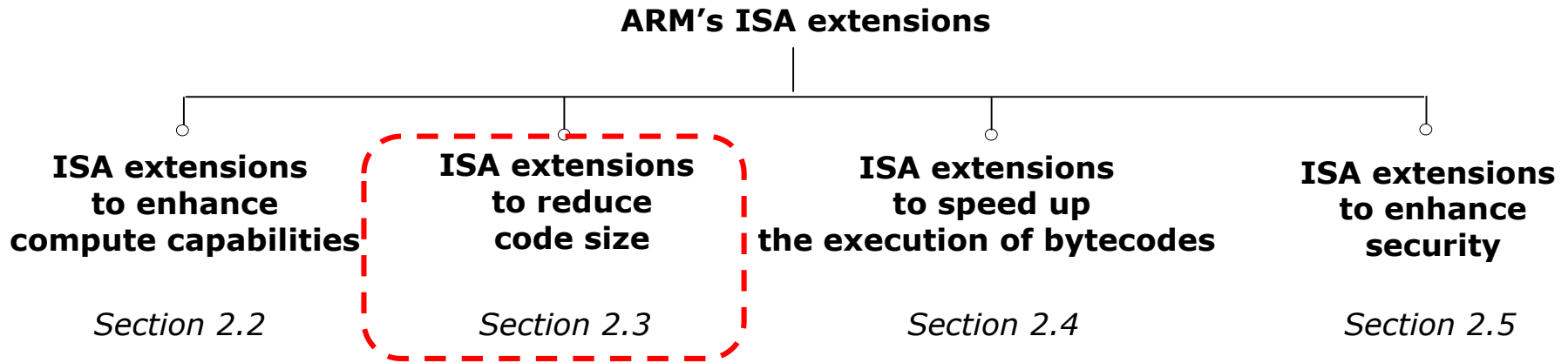
2.3 ISA extensions introduced to reduce code size

(Not discussed)

2.3.1 ISA extensions introduced to reduce code size - Overview (1)

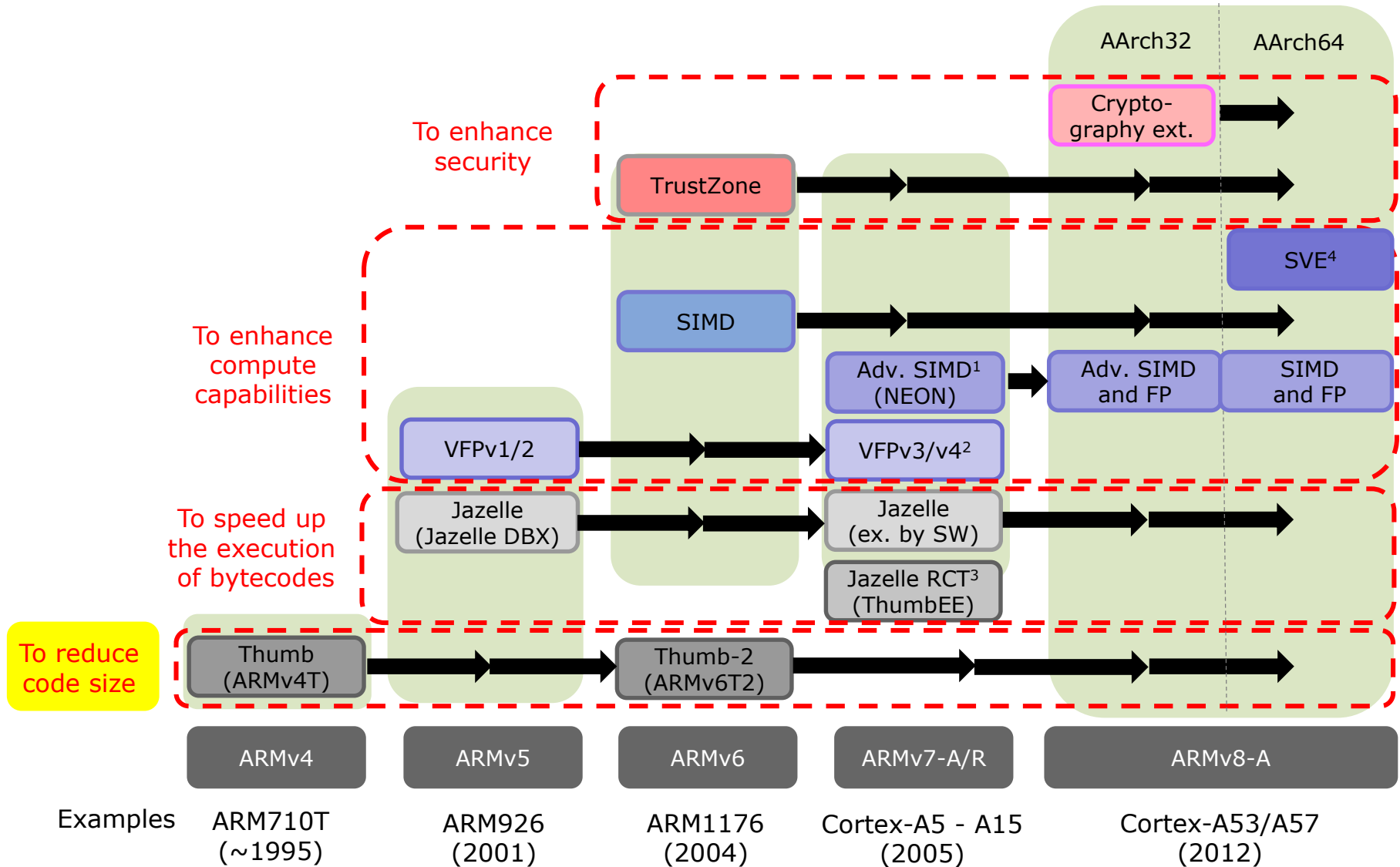
2.3 ISA extensions introduced to reduce code size

2.3.1 ISA extensions introduced to reduce code size - Overview (1)



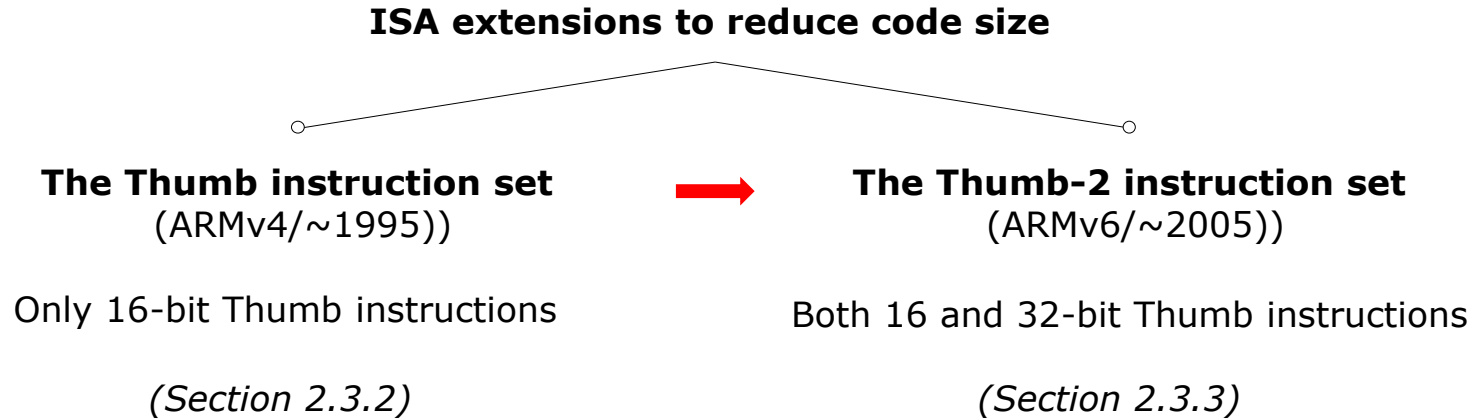
2.3.1 ISA extensions introduced to reduce code size - Overview (2)

ISA extensions introduced to reduce code size - Overview (2) (Based on [64])



Remarks: See on the slide 2.1 Overview (5)

ISA extensions introduced to reduce code size - Overview (3)

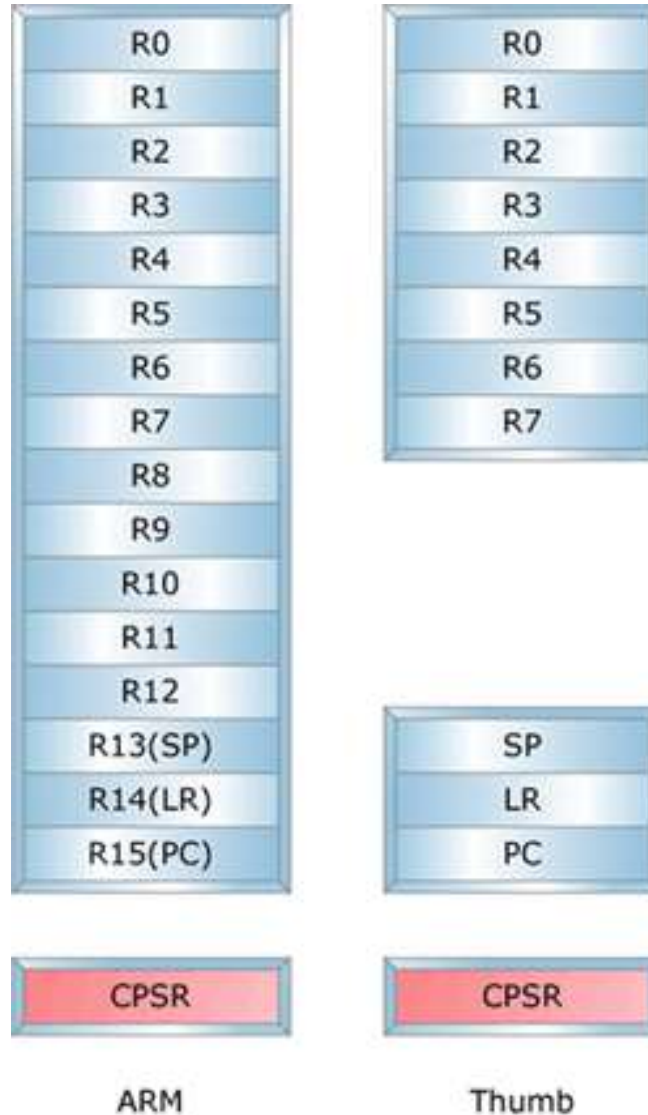


2.3.2 The Thumb instruction set [67]

- This is an **alternative 16-bit instruction set** that provides typically **35 to 40 % better code density** than traditional ARM code but **reduces performance slightly, e.g. by 10 %**.
- It has been introduced in the ARMv4 ISA.
- Processors with the **T suffix** provide beyond the default 32-bit ARM **also the 16-bit Thumb instruction set**.
- Thumb instructions are **16-bit long**, most 32-bit ARM instructions can be recoded to a single 16-bit Thumb instruction format.
- The Thumb instruction set is a **subset** of the ARM instruction set.
- There is a **special instruction for entering** the Thumb state (BX).
- Compilers are usually able to generate code optimally either for the ARM or the Thumb ISA.
- In the Thumb instruction set there are **only 8 GPRs** available for the programmer, as shown in the next Figure.

2.3.2 The Thumb instruction set (2)

Available GPR register sets in the ARM and the Thumb ISA [63]



2.3.3 The Thumb-2 instruction set [79]

- It was introduced along with the ARM1156T2-S processor in 2005.
- **Thumb-2** is a superset of the 16-bit ARMv6 Thumb ISA.
- It adds several new 16-bit instructions and also 32-bit instructions that can be freely intermixed in a program.
- The enhancements allow Thumb-2 to more efficiently cover the functionality of the ARM instruction set.
- In subsequent processors Thumb-2 replaced the Thumb instruction set.

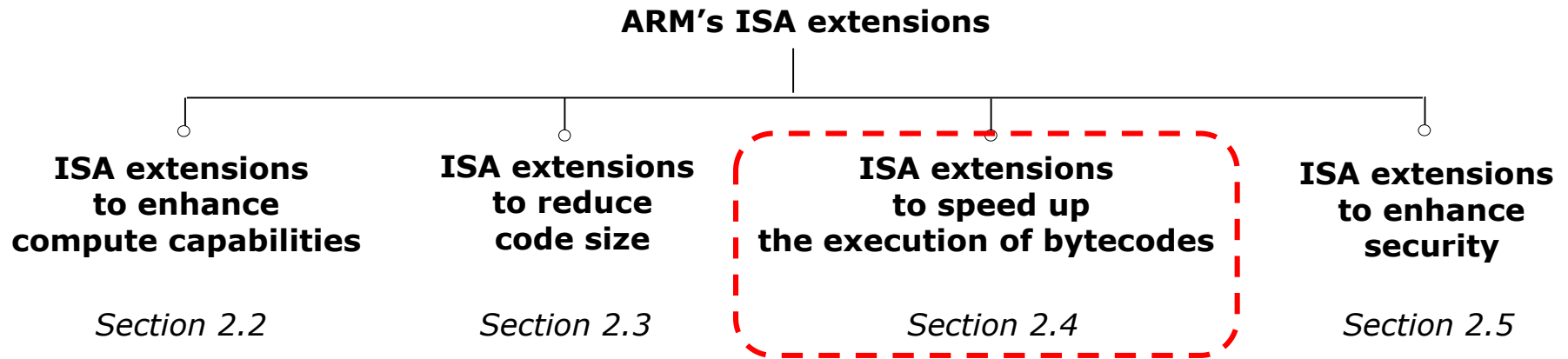
2.4 ISA extensions to speed up the execution of bytecodes

(Not discussed)

2.4.1 ISA extensions to speed up the execution of bytecodes - Overview (1)

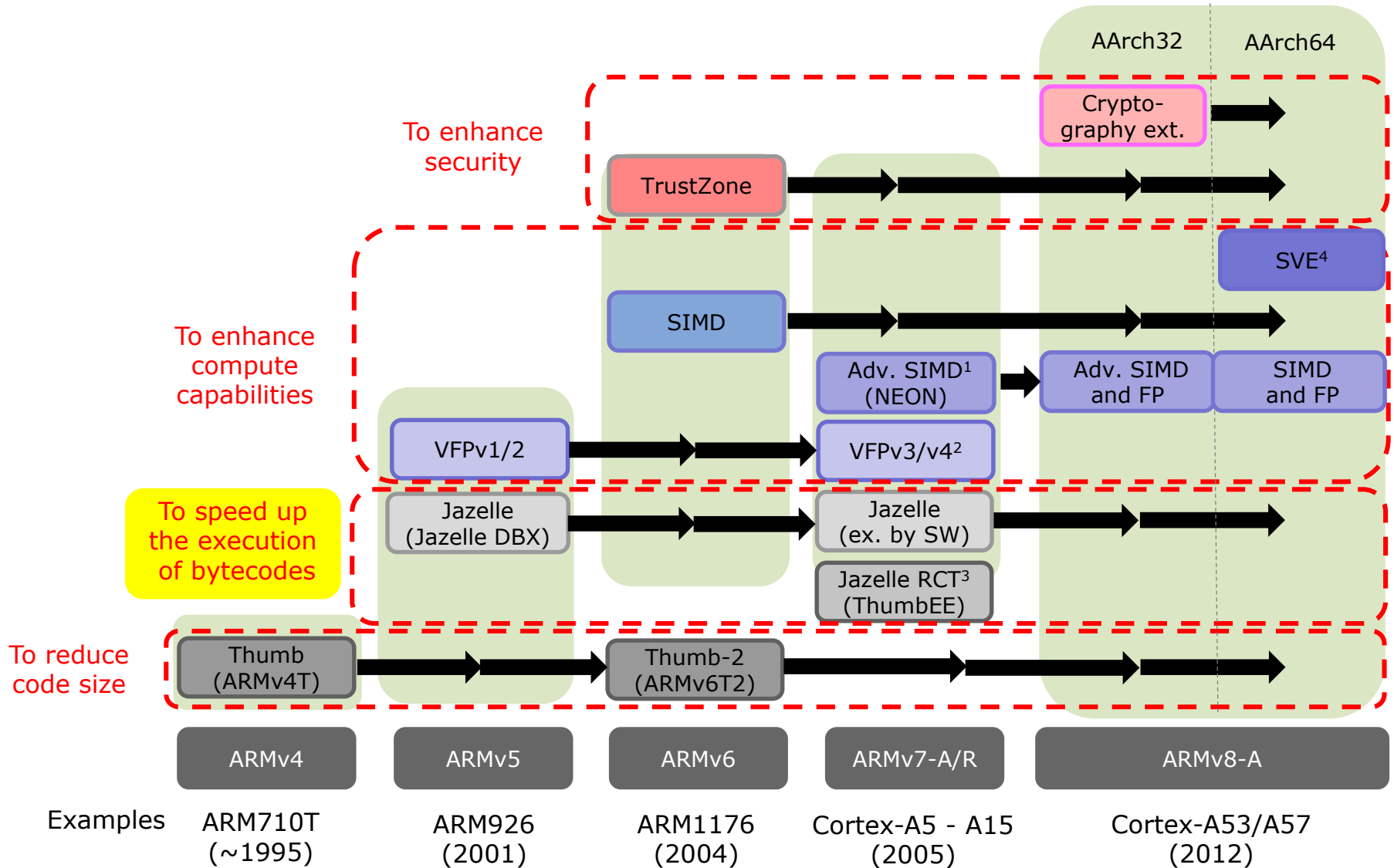
2.4 ISA extensions introduced to speed up the execution of bytecodes

2.4.1 ISA extensions introduced to speed up the execution of bytecodes
- Overview (1)



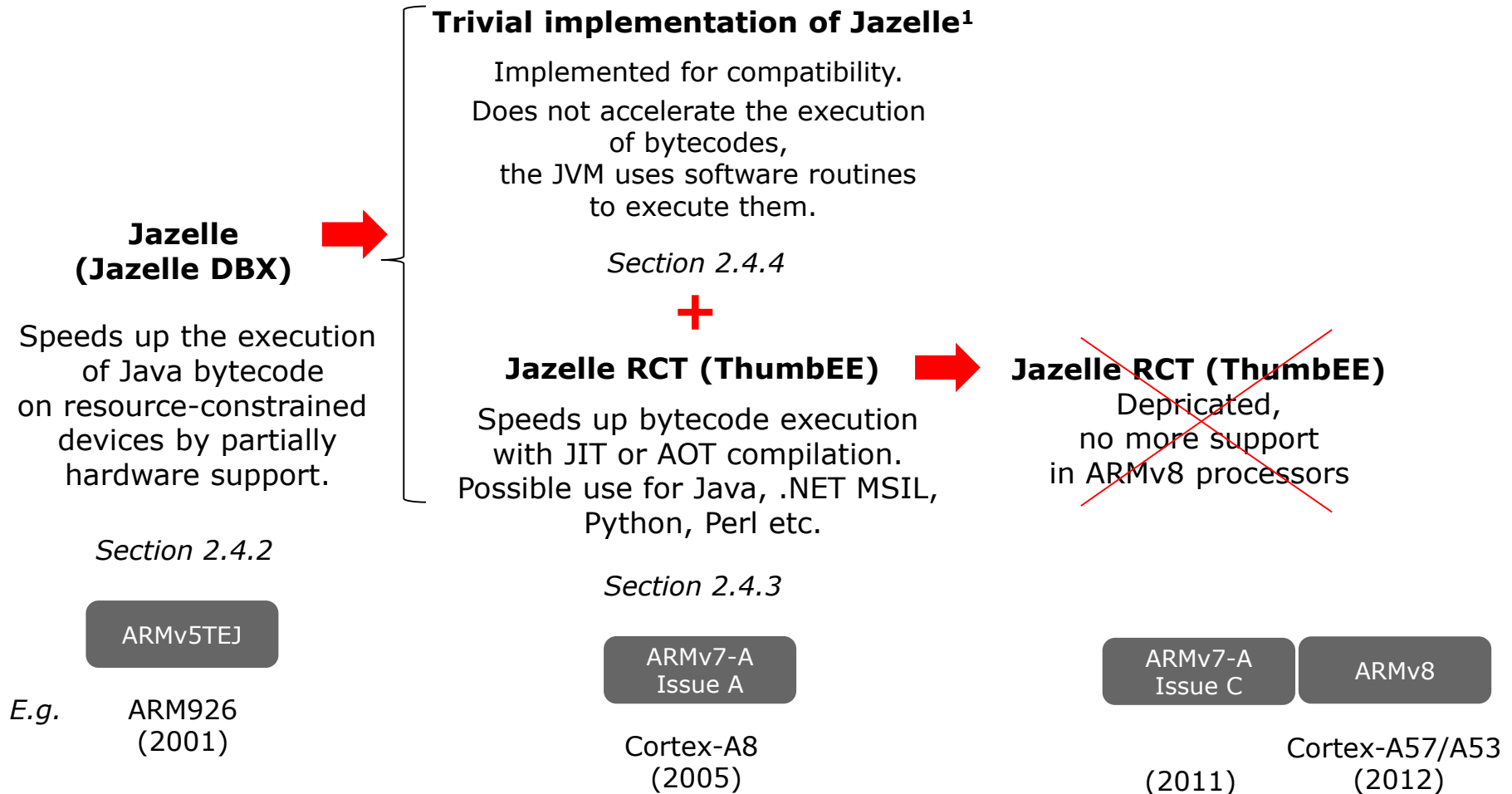
2.4.1 ISA extensions to speed up the execution of bytecodes - Overview (2)

ISA extensions introduced to speed up the execution of bytecodes Overview (2) (Based on [64])



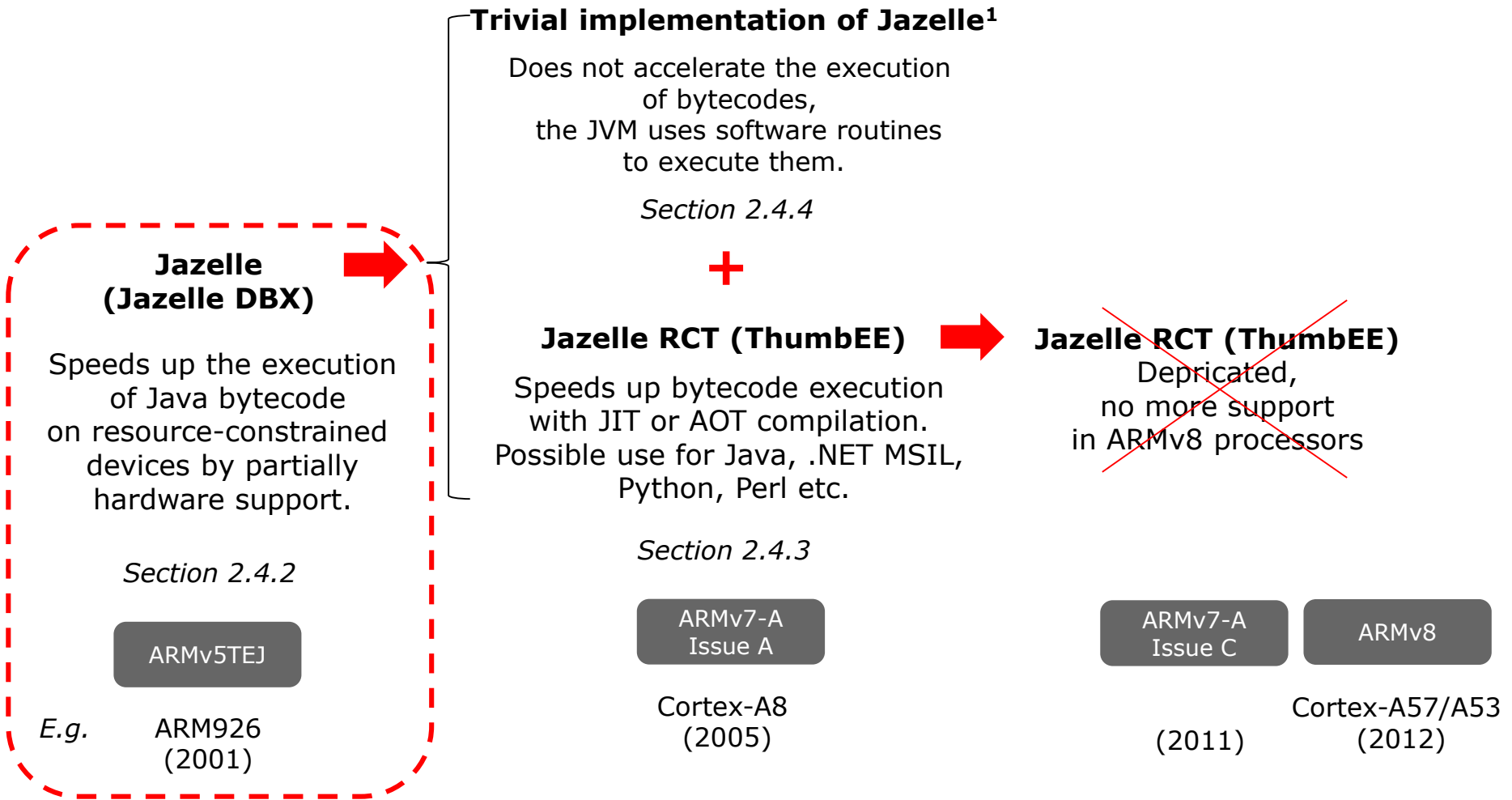
Remarks: See on the slide 2.1 Overview (5).

ISA extensions to speed up the execution of bytecodes - Overview (3)



¹A few ARMv7 processors, such as the Cortex-A5 and the Cortex-A9 support Jazelle DBX as an option.

2.4.2 Jazelle (Jazelle DBX) -1



¹A few ARMv7 processors, such as the Cortex-A5 and the Cortex-A9 support Jazelle DBX as an option.

2.4.2 Jazelle (Jazelle DBX) -2

It is ARM's [third ISA option](#), introduced in the ARMv5TEJ ISA (in 2001), as indicated below.

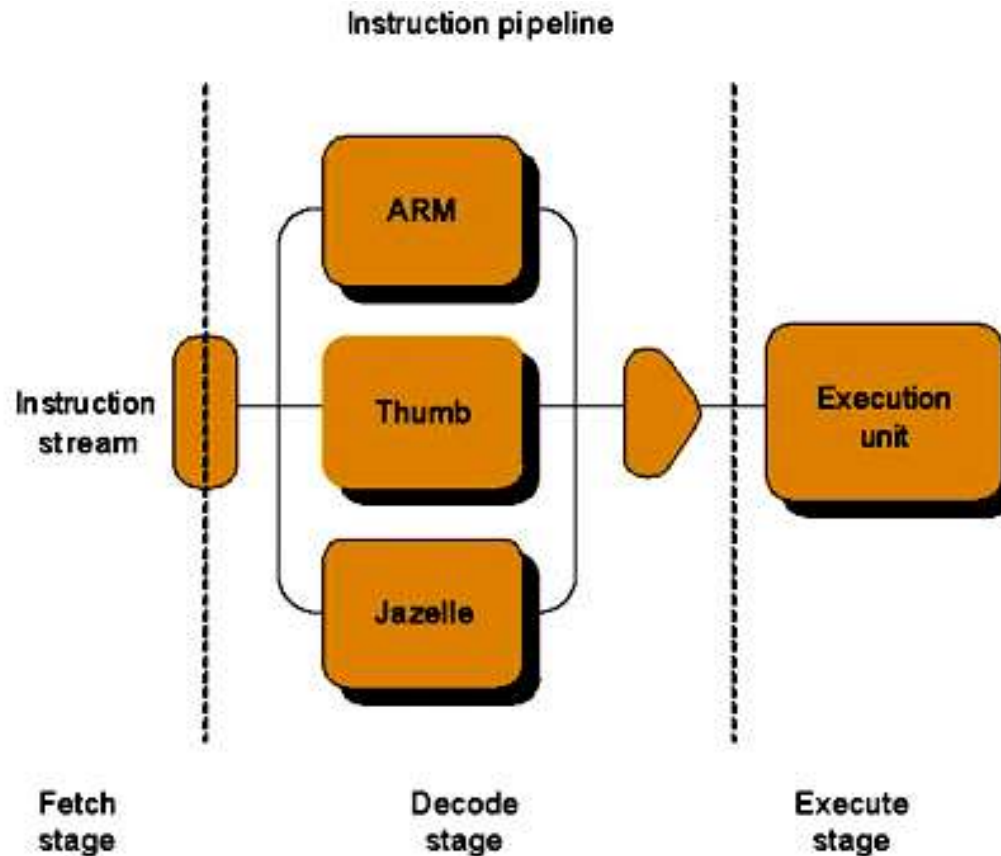


Figure: The Jazelle ISA extension as ARM's third ISA alternative [67]

It aims at [accelerating the execution of Java bytecode](#).

Java bytecode

- It is one kind of **bytecodes**.
- **Bytecodes** are **compact 1-byte codes** written for a stack based **virtual machine (virtual ISA)**.

All opcodes are one byte long followed by optional parameters.

2.4.2 Jazelle (Jazelle DBX) (4)

Part of a Java
bytecode [80]

```
; ----- section.constpool:
0x00000000      ca      breakpoint
0x00000001      fe      impdep1
0x00000002      babe000300 invokedynamic (48639)
0x00000007      2d      aload_3
0x00000008      00      nop
0x00000009      21      lload_3
0x0000000a      0a      lconst_1
0x0000000b      00      nop
0x0000000c      07      iconst_4
0x0000000d      00      nop
0x0000000e      1009    bipush 9
0x00000010      00      nop
0x00000011      110012  sipush 0x11 0x0
0x00000014      08      iconst_5
0x00000015      00      nop
0x00000016      130a00  ldc_w "Code"
0x00000019      110014  sipush 0x11 0x0
0x0000001c      0a      lconst_1
0x0000001d      00      nop
0x0000001e      1500    iload 0
0x00000020      1607    lload 7
0x00000022      00      nop
0x00000023      1707    fload 7
0x00000025      00      nop
0x00000026      1801    dload 1
0x00000028      00      nop
0x00000029      06      iconst_3
; ----- str.init:
0x0000002a      .string "init" ; len=6
0x00000030      01      aconst_null
0x00000031      00      nop
0x00000032      03      iconst_0
; ----- str.V:
0x00000033      .string "V" ; len=3
```


The computational model assumed for bytecodes

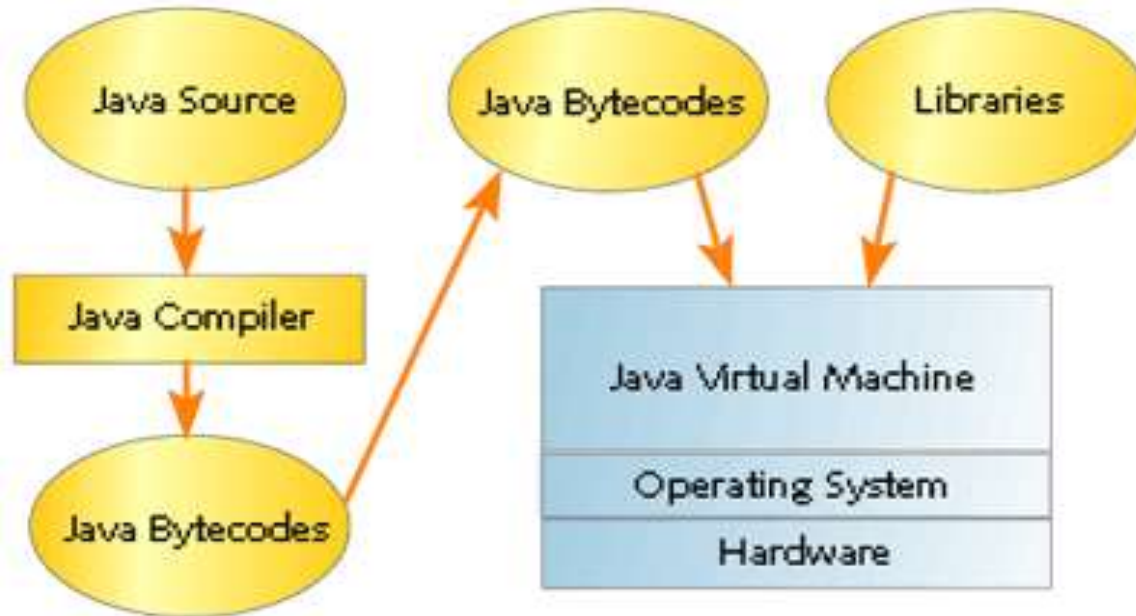
- The **computational model** of bytecodes presumes a **stack based execution**, i.e. operands are first loaded into the stack and operations will be executed on the operands being in the stack.

Principle of generating and executing Java bytecode -1 [81]

- Bytecodes are generated by compiling a source code assuming a virtual ISA that is underlying the considered bytecode and is not bound to any real ISA, like the ARM or x86 ISA.
- So bytecodes can not be directly run on any processor, i.e. they are a kind of pseudocode that can be executed by a virtual machine, as indicated in the next Figure for the Java bytecode.

2.4.2 Jazelle (Jazelle DBX) (7)

Principle of generating and executing Java bytecode -2 [81]



2.4.2 Jazelle (Jazelle DBX) (8)

Generating and executing Java bytecode on different platforms [82]

- It is the task of the **Java compiler** to generate the **Java bytecode** from a given source language.
- The **Java compiler** runs under a given OS on a particular platform, as seen in the Figure below.

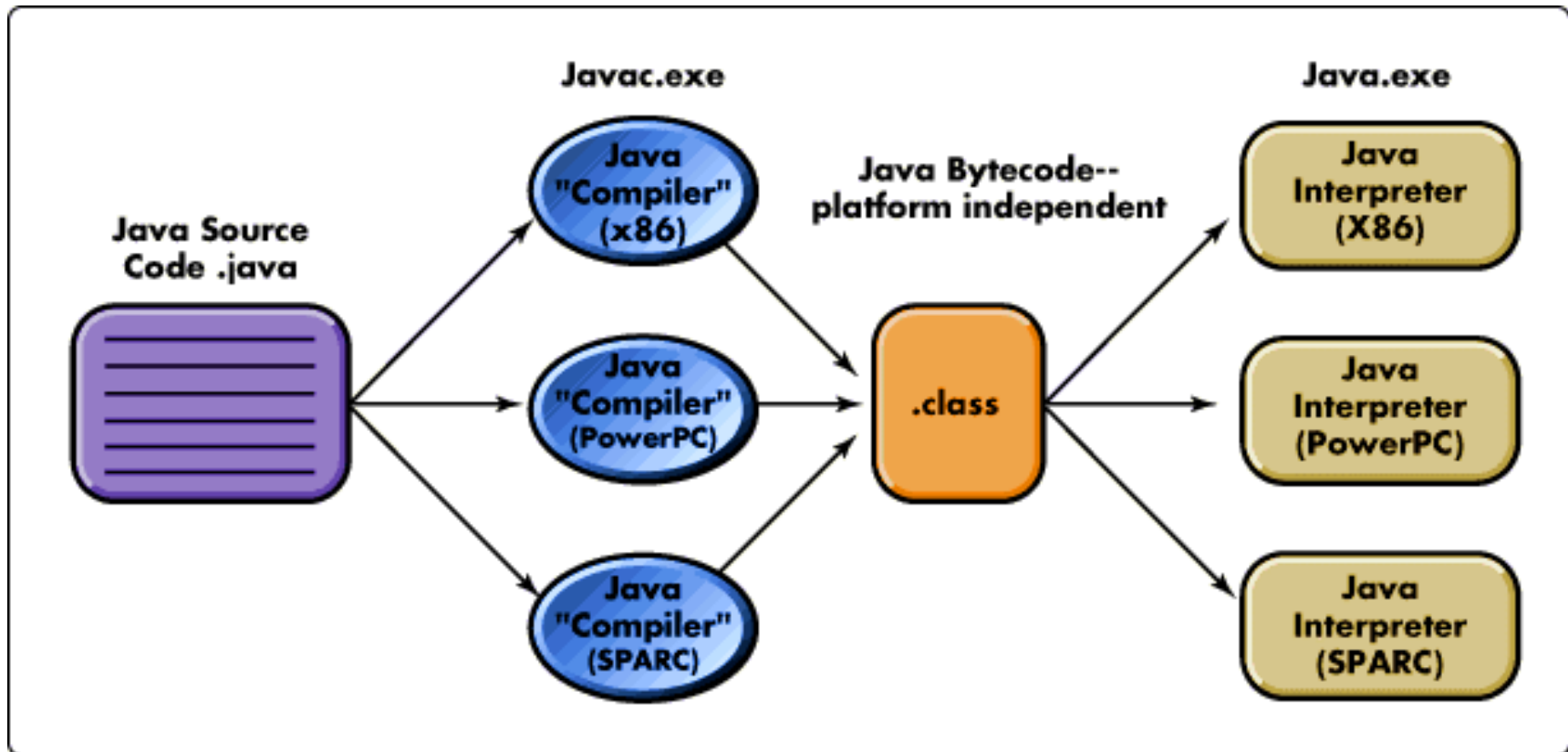


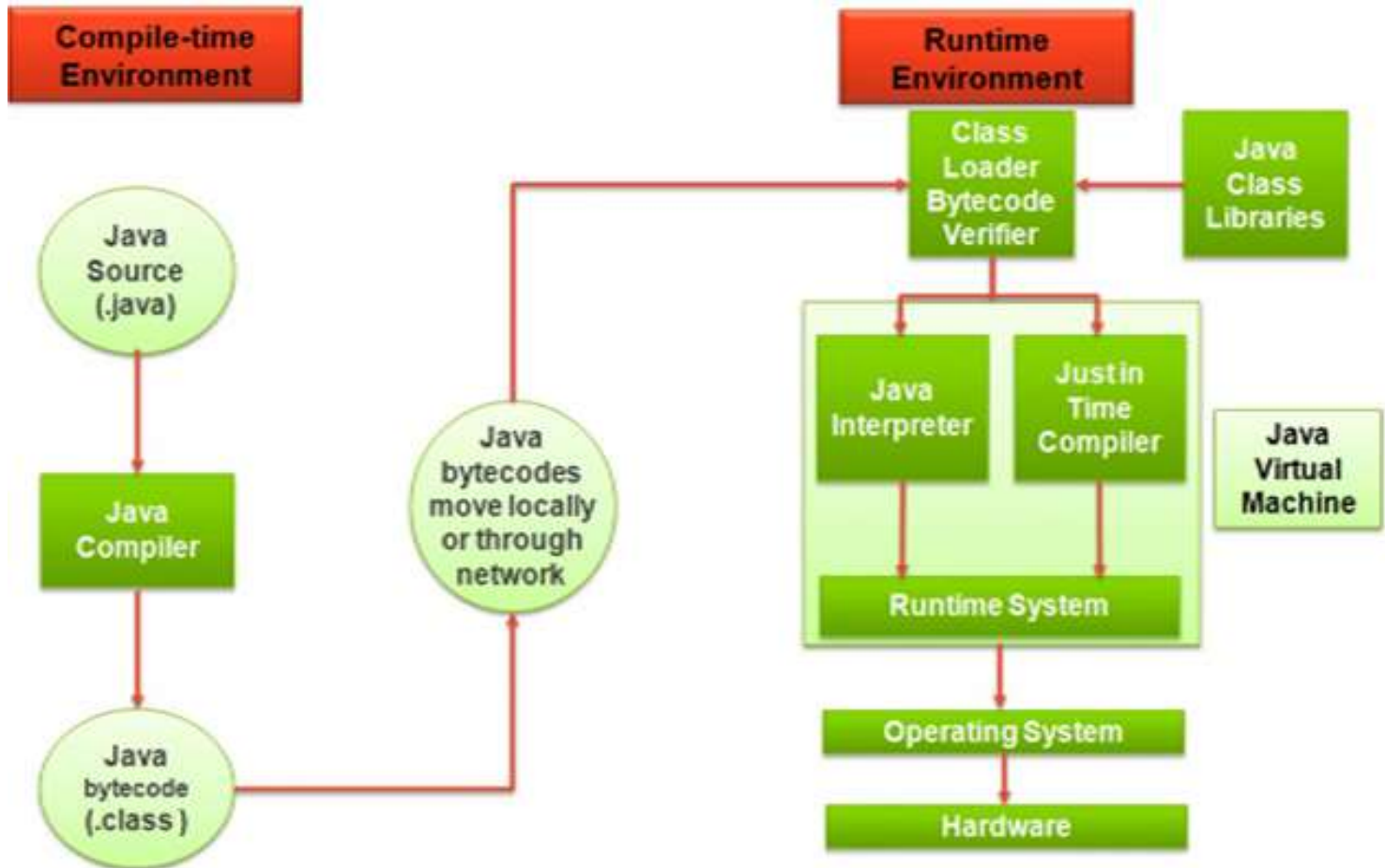
Figure: Principle of generating Java bytecode [82]

Principle of executing Java bytecode [83]

- The most straightforward way to implement a Java Virtual Machine or virtual machines at all is to use a **software interpreter** as a virtual machine.
- A **faster implementation** can be achieved by **compiling the bytecode** by a **JIT or AOT compiler** (as seen in the next Figure and discussed in Section 3.3.3) on the processor itself and/or by using **hardware support** for the execution, such as ARM's **Jazelle DBX** or a **coprocessor**.

2.4.2 Jazelle (Jazelle DBX) (10)

Implementing a Java Virtual Machine by including a JIT compiler into it [83]

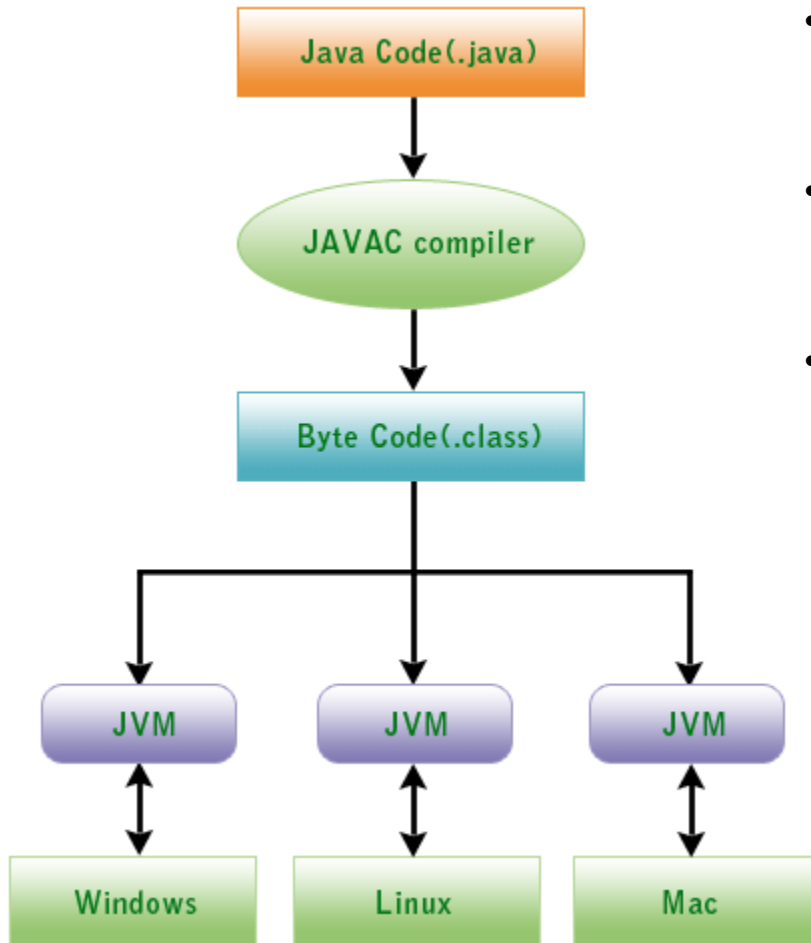


Portability of the Java bytecode or bytecodes at all

- Since **bytecodes are not bound to a particular real ISA** but will be executed by a virtual machine that suits the target ISA and target OS, **bytecodes are portable**.
- It follows that **for a given type of bytecode**, e.g. Java bytecode, there are **many possible virtual machines**, as the next Figure indicates.

2.4.2 Jazelle (Jazelle DBX) (12)

Example: Executing Java bytecode on different platforms



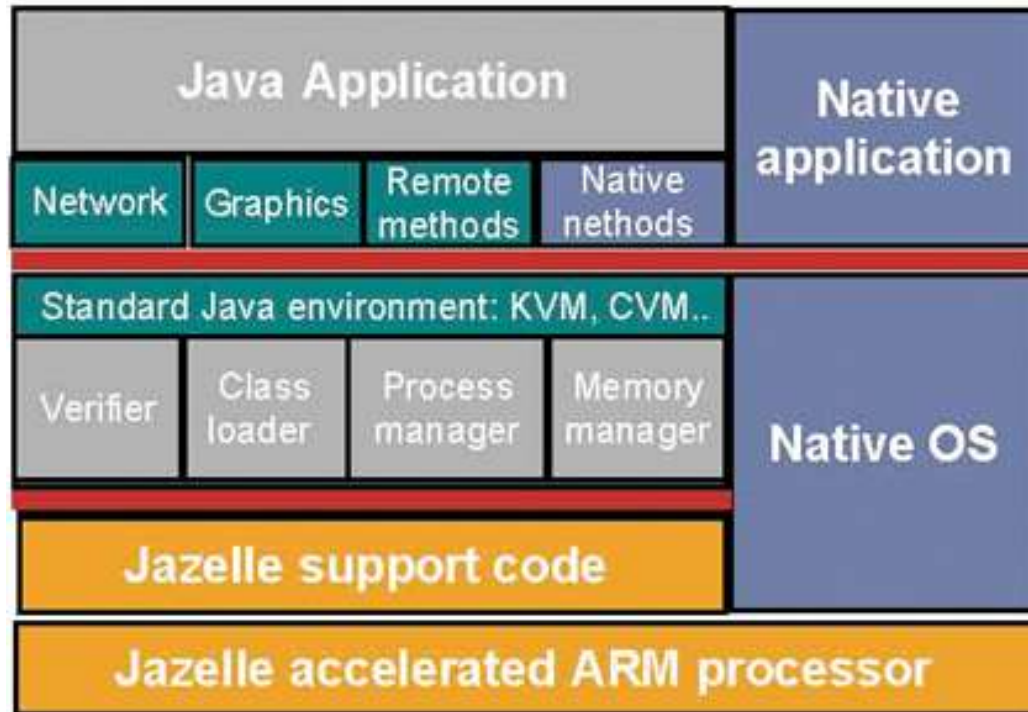
- The bytecode is processed by the **JVM** component of the **Java Runtime Environment (JRE)**.
- Each platform needs one or more JVMs that suits the OS and the target ISA, as indicated in the Figure.
- It is then the task of the JVM to execute the bytecode e.g. by interpreting each bytecode instructions.

Figure: Different JVMs for different OSs and CPU ISA [84]

Jazelle DBX support for executing Java bytecode on ARM processors

- An inherent **drawback of interpreting Java bytecode** vs. compiled execution is **slower speed**.
- **Jazelle DBX** aims at mitigating this drawback by providing **hardware support** for executing Java bytecode.
- For this reason Jazelle DBX provides a **third execution state**, called the **Jazelle processor state**, where **the processor decodes Java bytecode to ARM instructions** as follows:
 - about 2/3 of Java bytecode will directly be mapped to ARM instructions, and
 - the remainder will be trapped as exceptions and emulated by multiple ARM instructions via roughly 8 kB of microcode memory.
- **A special instruction** (the BXJ instruction) is used **to switch the processor from the ARM state to the Jazelle state**.
- Jazelle DBX was introduced in the ARMv5 based ARM926 in 2001 and was used in ARMv5 and ARMv6 processor implementations typically with very limited memory.

Implementation of Jazelle DBX [67]



Remark 1: The Java programming environment [85]



- **JDK (Java Developer Kit)** contains tools needed for developing Java programs, such as the compiler (`javac.exe`), etc.
- The compiler converts Java code into **byte code**.
- **JRE (Java Runtime Environment)** contains **JVM** that actually runs the Java program, class libraries and other supporting files.
- **JVM (Java Virtual Machine)** runs the program, and it uses the class libraries, and other supporting files provided in JRE.
- To run Java programs, the fitting JRE needs to be installed on the system.

- **JVM** itself is not platform independent.
- When JVM has to interpret the byte codes to machine language, then it has to interact with the OS and interpretes Java bytecode on the ARM ISA.

Figure: The Java programming environment [85]

Remark 2: Code generation in .NET

A further example for making use of the virtual machine principle is Microsoft's **MSIL bytecode** (officially known as **CIL (Common Intermediate language)**), that is utilized by the **.NET framework** (see the Figure below).

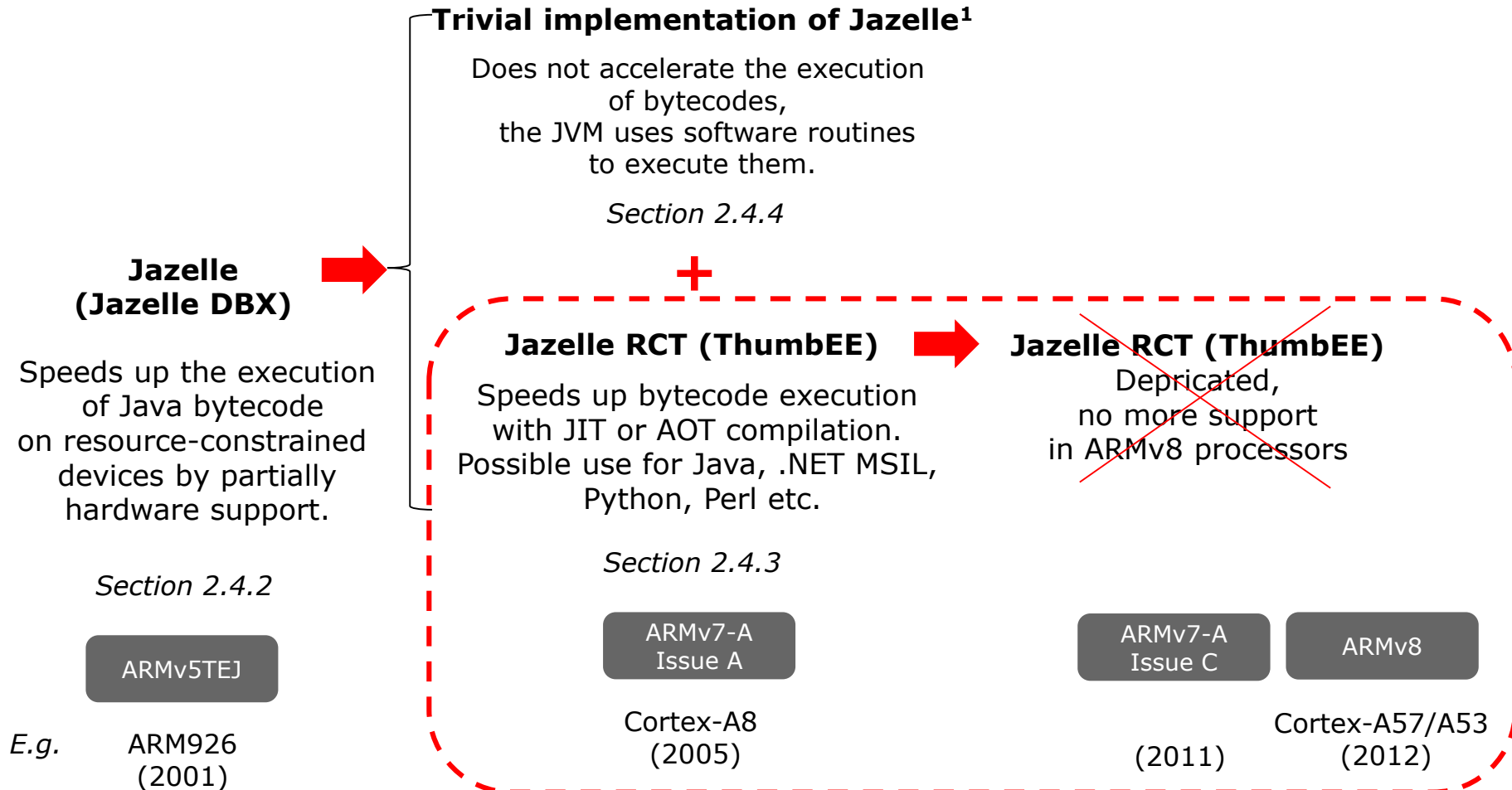


Figure: Code generation in .NET [88]

2.4.3 Jazelle RCT (Runtime Compilation Target) (1)

2.4.3 Jazelle RCT (Runtime Compilation Target) -1

It was **introduced in 2005** (along with the first Cortex processor (Cortex-A8), based on the ARMv7 Issue A,) but **withdrawn in 2011** (with the ARMv7 Issue C,) as seen below.



¹A few ARMv7 processors, such as the Cortex-A5 and the Cortex-A9 support Jazelle DBX as an option.

Jazelle RCT (Runtime Compilation Target) [86] -2

- **Jazelle RCT** is dubbed also as the **ThumbEE** extension, since it makes use of the ThumbEE execution environment that is based on an enhanced version of the Thumb2 instruction set.
- It aims at **speeding up the execution of dynamically generated code**.
- This is code that is **compiled e.g. from a portable bytecode** e.g. Java, Perl, Python bytecode or .Net MSIL, **on the processor**
 - **either while downloading the bytecode** (by an **Ahead-Of-Time or AOT compiler**) or
 - **during execution of the code** (by an **Just-In-Time compiler**).

2.4.3 Jazelle RCT (Runtime Compilation Target) (3)

Example: Just-in-time (JIT) compilation in Java environment [73]

- Java bytecode is loaded, compiled to native code each time a method is called, compiled code is cached in memory and will be executed by the Java Virtual Machine (JVM), as shown below.

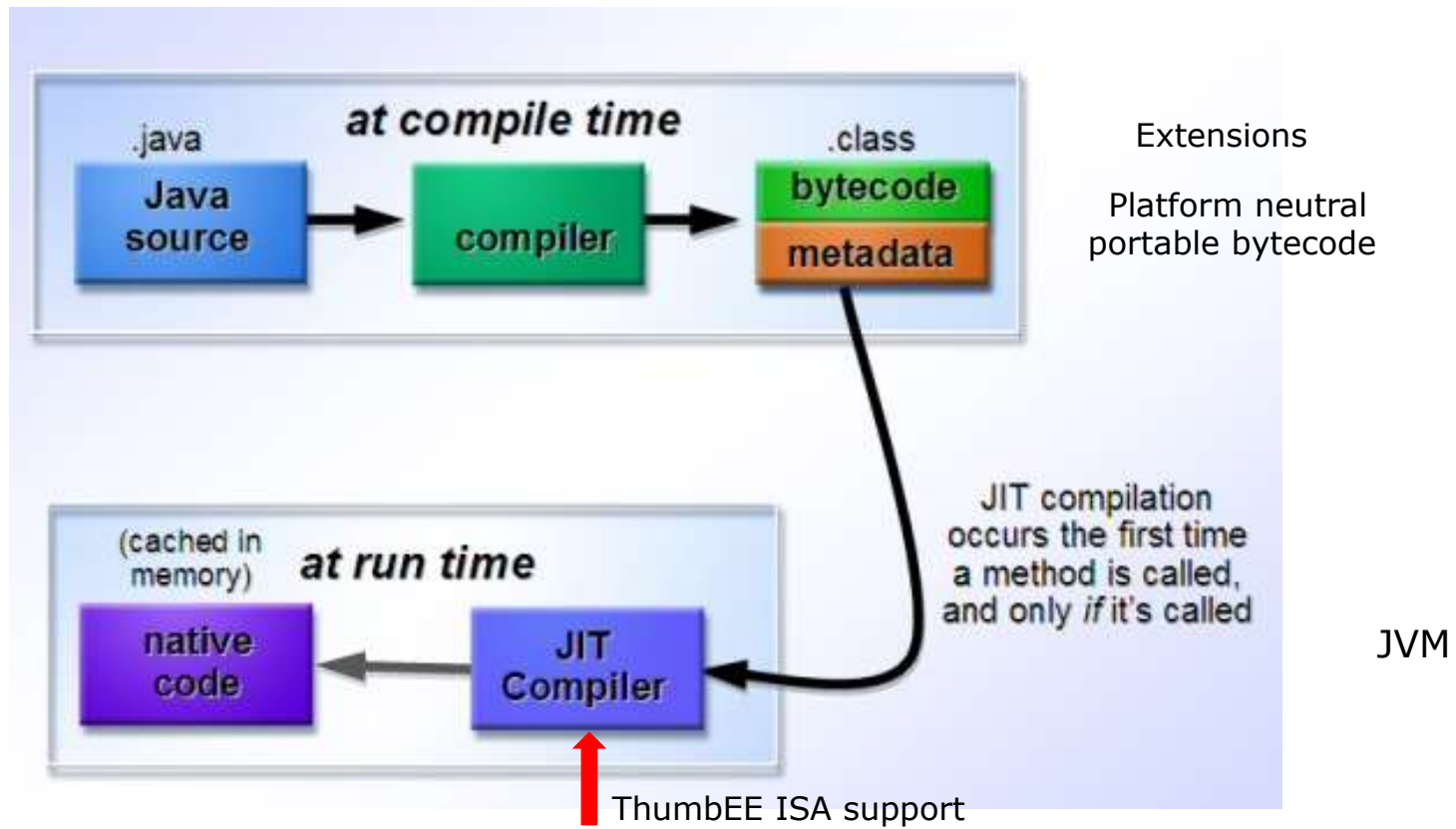


Figure: Just-in-time compilation in Java environment [73]

Benefits of using Jazelle RCT [67]

- It provides a **higher speed up** in executing bytecode than Jazelle DBX but requires **more memory**.
- It follows that **the introduction of Jazelle RCT (ThumbEE) made Jazelle DBX more or less superfluous**.

As a consequence, when ARMv7 based Cortex-A processors introduced Jazelle RCT(ThumbEE), at the same time they **typically stopped supporting Jazelle DBX**.

(In fact, the Cortex-A8/A7 or A15 processors as well as the subsequent ARMv8 based processors provided instead only a trivial (software emulated) Jazelle implementation (see Section 2.4.4) whereas the Cortex-A5 and A9 processors supported Jazelle DBX as an option).

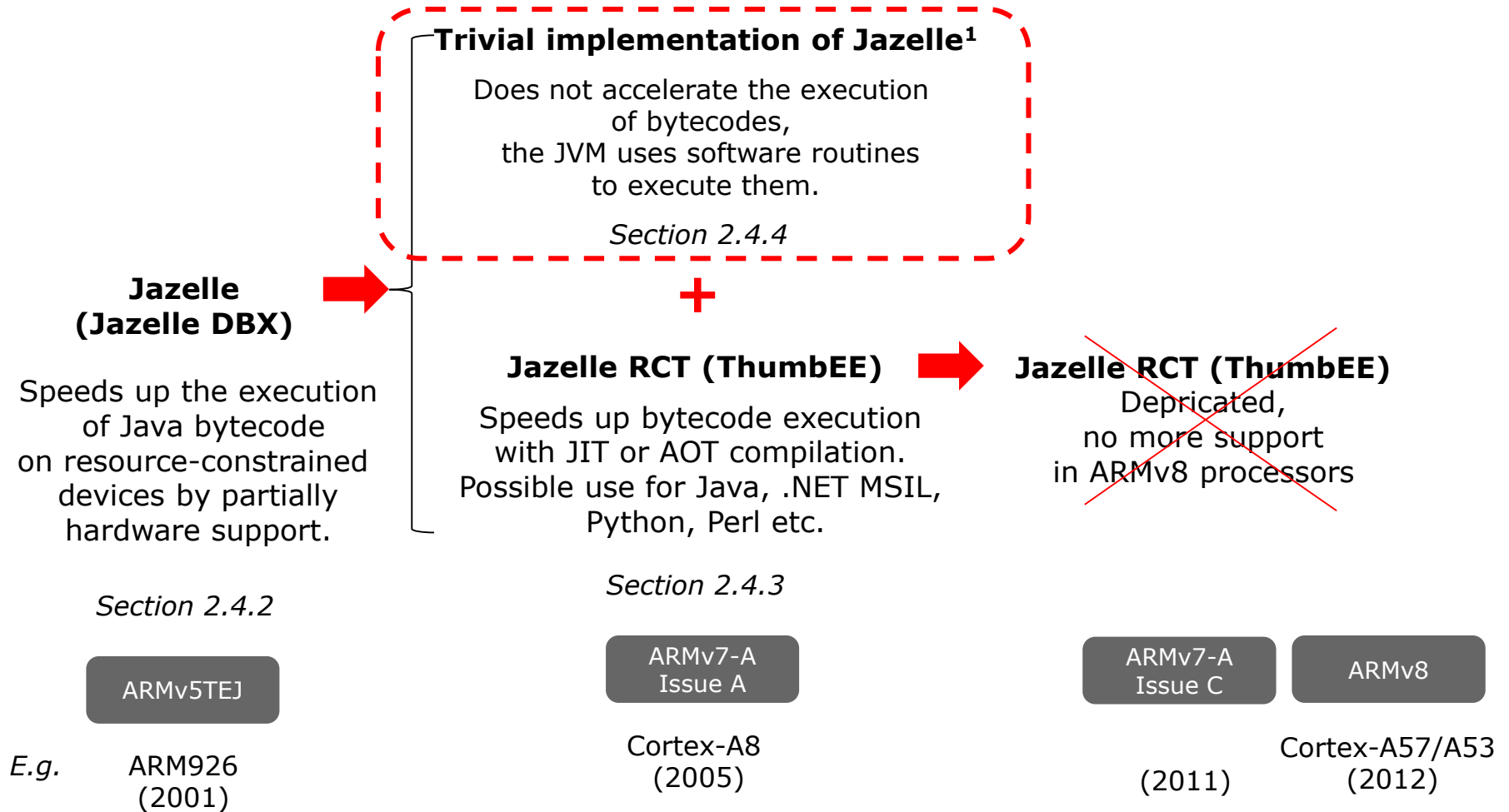
Drawbacks of dynamic compilation [67]

- AOT or JIT compilation causes a [delay between an application's launch and its actual run](#), this can prevent their use in real time application.
- Further on, [dynamically compiled code expands four to six times](#).

So, in addition to delaying the startup of an application, AOT and JIT compilation requires extra memory for the code compiled.

- As a consequence, [ARM deprecated the use of the Jazelle RCT \(ThumbEE\) ISA extension in the ARMv7 Issue C \(10/2011\)](#) and subsequent ARMv8 ISA.

2.4.4 Trivial implementation of Jazelle -1



¹A few ARMv7 processors, such as the Cortex-A5 and the Cortex-A9 support Jazelle DBX as an option.

Trivial implementation of Jazelle -2 [87]

- The introduction of Jazelle RCT in Cortex-A processors made Jazelle DBX obsolete, but for reasons of compatibility, in subsequent processors ARM provided a so called **trivial implementation of Jazelle**.
- In the trivial Jazelle implementation, the processor does not accelerate the execution of any bytecodes, JVM uses software routines to execute any bytecode.

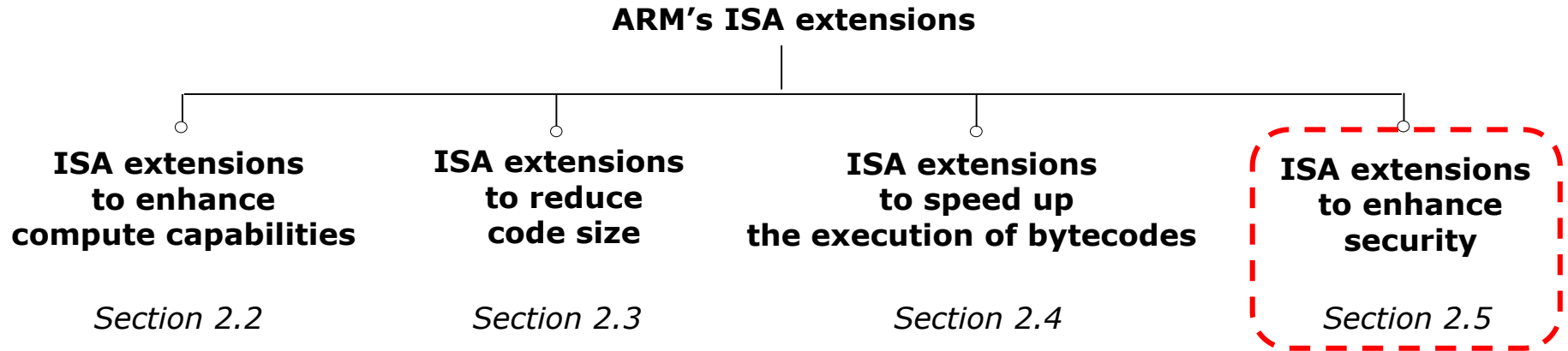
2.5 ISA extensions introduced to enhance security

(Not discussed)

2.5.1 ISA extensions introduced to enhance security - Overview (1)

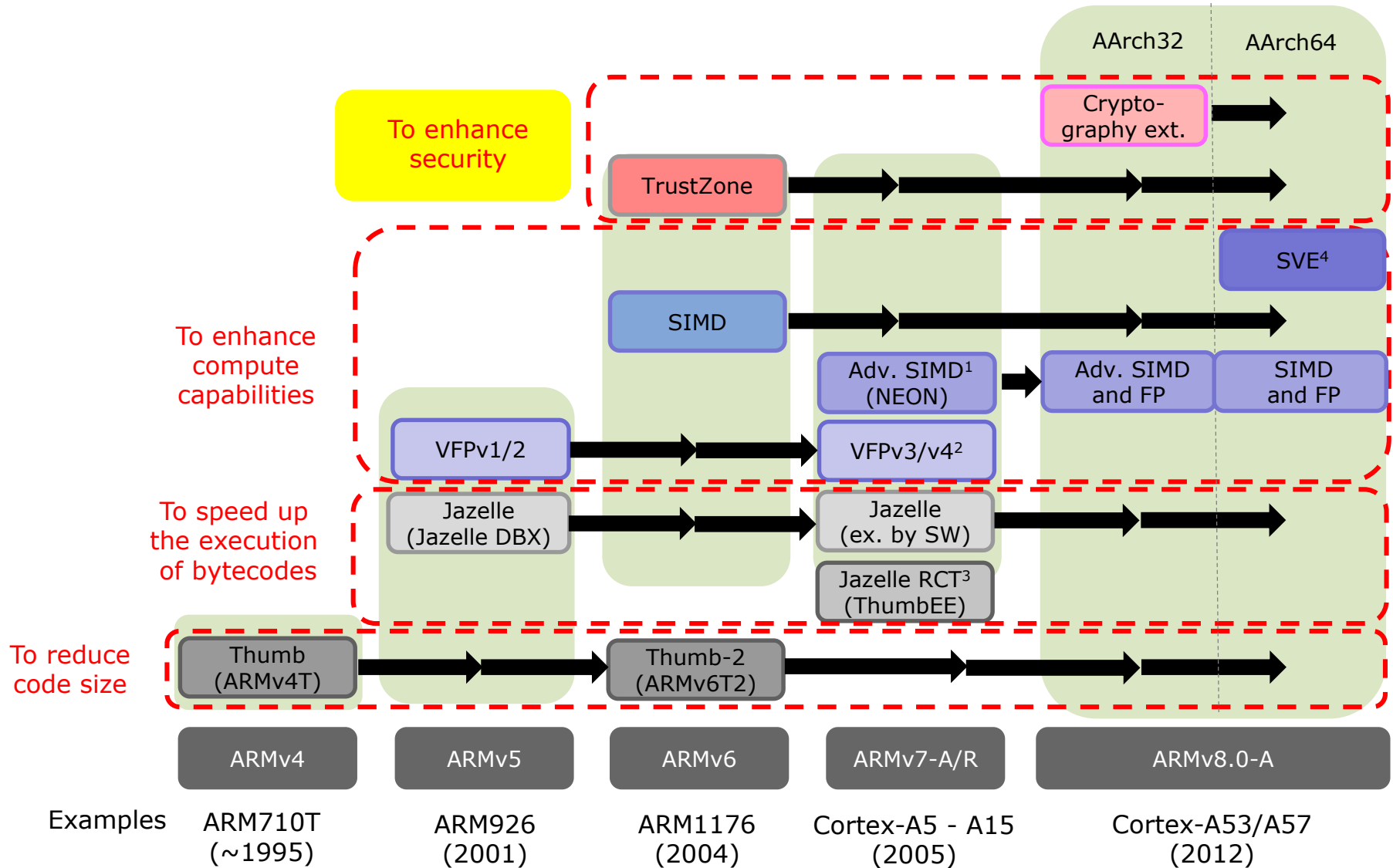
2.5 ISA extensions introduced to enhance security

2.5.1 ISA extensions introduced to enhance security - Overview (1)



2.5.1 ISA extensions introduced to enhance security - Overview (2)

ISA extensions introduced to enhance security - Overview (2) (Based on [64])



Remarks: See on the slide 2.1 Overview (5).

ISA extensions introduced to enhance security - Overview (3)

ISA extensions introduced to enhance security

```
graph TD; A[ISA extensions introduced to enhance security] --- B[The TrustZone extension]; A --- C[The Cryptography extension];
```

The TrustZone extension

ARMv6 (2001)

Section 2.5.2

The Cryptography extension

ARMv8 (2012)

Section 2.5.3

2.5.2 The TrustZone extension [68]

- It provides a **system-wide protection** against possible attacks.
- It is achieved **by partitioning system wide hardware and software resources** so that **they exist in one of two worlds**;
 - in the **Secure world** for the security subsystem and
 - the **Normal world** for everything else.
- **Hardware logic of the TrustZone ensures that no Secure world resources can be accessed from the Normal world.**
- The trustZone extension was introduced as part of the ARMv6 ISA.
- For details see e.g. [68].

2.5.3 The Cryptography extension [69]

- It adds **new instructions to accelerate the execution of cryptographic algorithms**, like
 - Encryption/decryption according to the Advanced Encryption Standard (AES),
 - Secure Hash Algorithm (SHA) functions SHA-1, SHA-224, and SHA-256.
- It was introduced as part of the ARMv8 ISA.
- It is similar to Intel's AES-NI ISA extension introduced along with the Westmere basic architecture (2009).

3. Overview of ARM's processor series

- 3.1 Overview
- 3.2 Processors implementing the ARMv1 – ARMv2 ISA
- 3.3 Processors implementing the ARMv3 – ARMv6 ISA
- 3.4 Processors implementing the ARMv7 – ARMv8 ISA

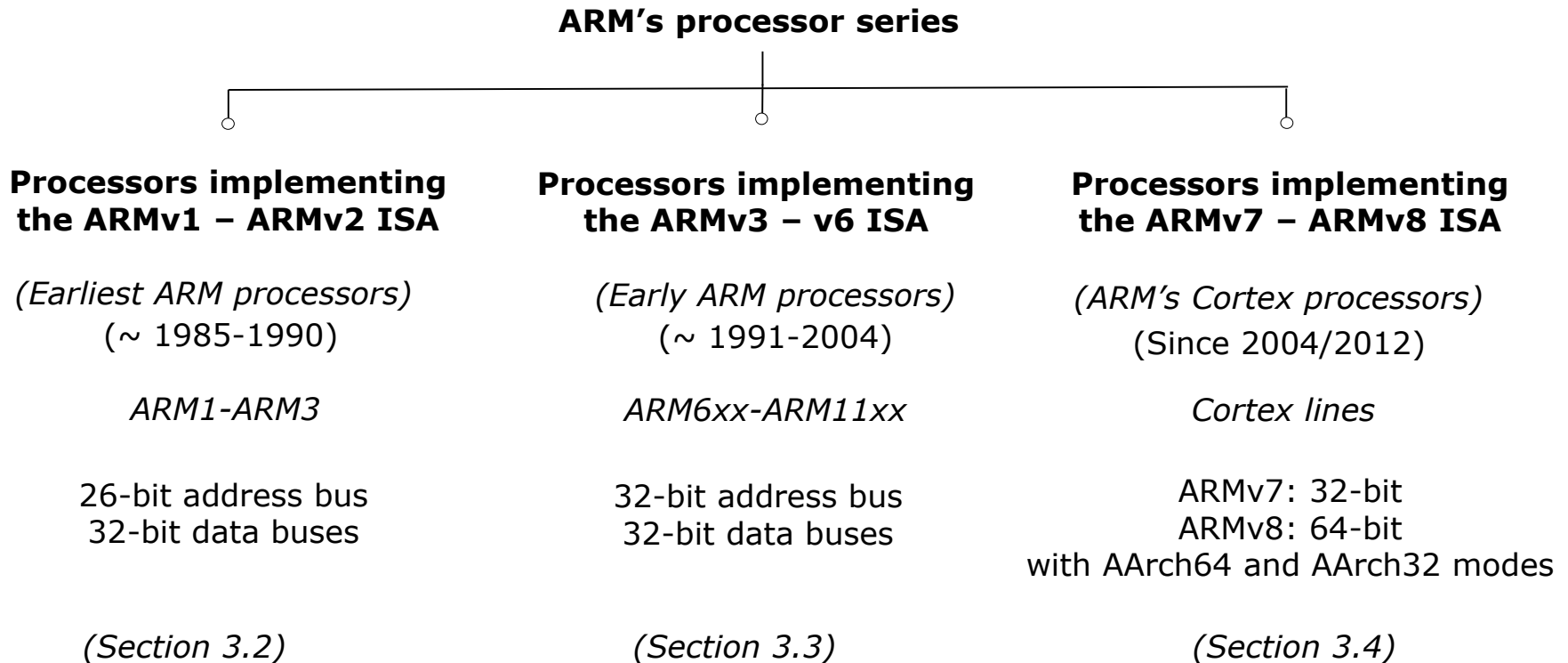
Only Section 3.4 will be discussed.

3.1 Overview of ARM's processor series

3.1 Overview of ARM's processor series (1)

3.1 Overview

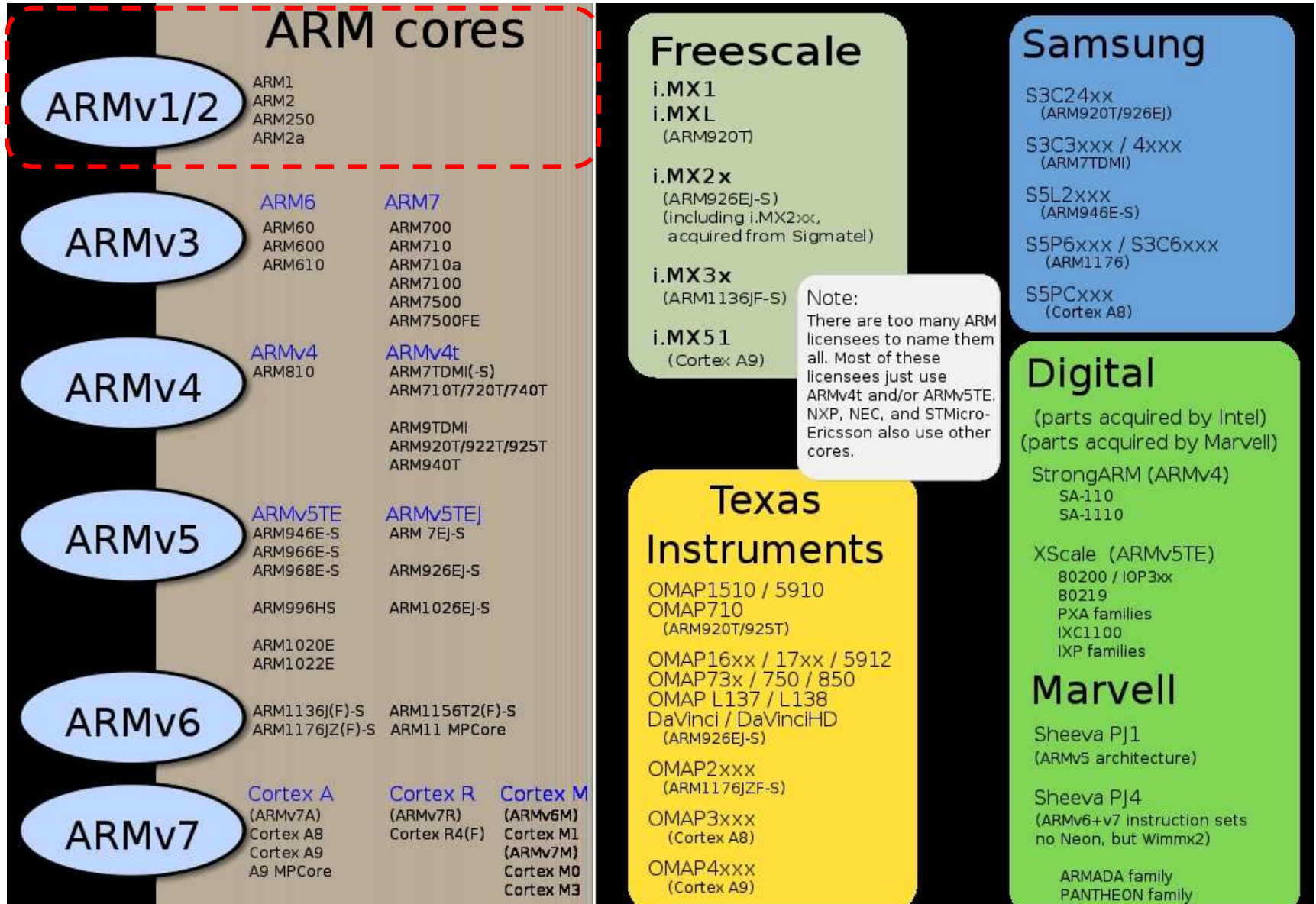
Subsequently, we give an overview of ARM's processor series **subdivided into three sections, according their underlying ISAs**, as follows.



3.2 Processors implementing the ARM v1 - ARM v2 ISA

3.2 Processors implementing the ARM v1 - ARM v2 ISA (1)

3.2 Processors implementing the ARMv1 – ARMv2 ISA -1 [5]



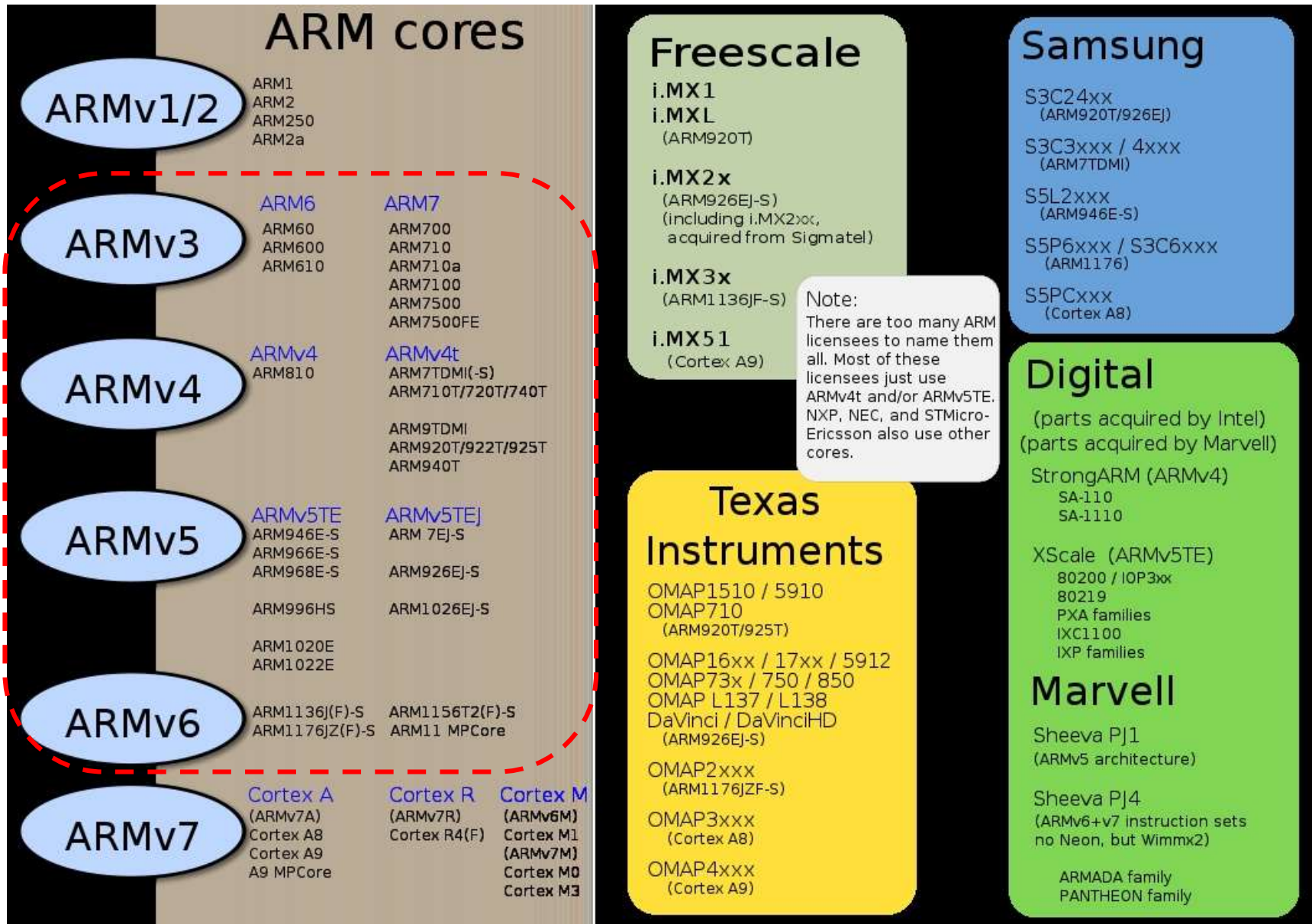
Remarks

- As already discussed in Section 2.1
 - processors based on the ARM ISA versions ARMv1 and ARMv2 (like ARM1, ARM2 or ARM3) implement only a 26-bit address bus and a 32-bit data bus.
 - They are called 26-bit architectures.
 - By contrast, processors based on the ARMv3 or higher ISA version support already a 32-bit address space and are known as 32-bit architectures up to the ARMv7 ISA.
 - In the ARMv8 ISA the AArch64 mode supports already 64-bit addressing.

3.3 Processors implementing the ARM v3 - ARM v6 ISA

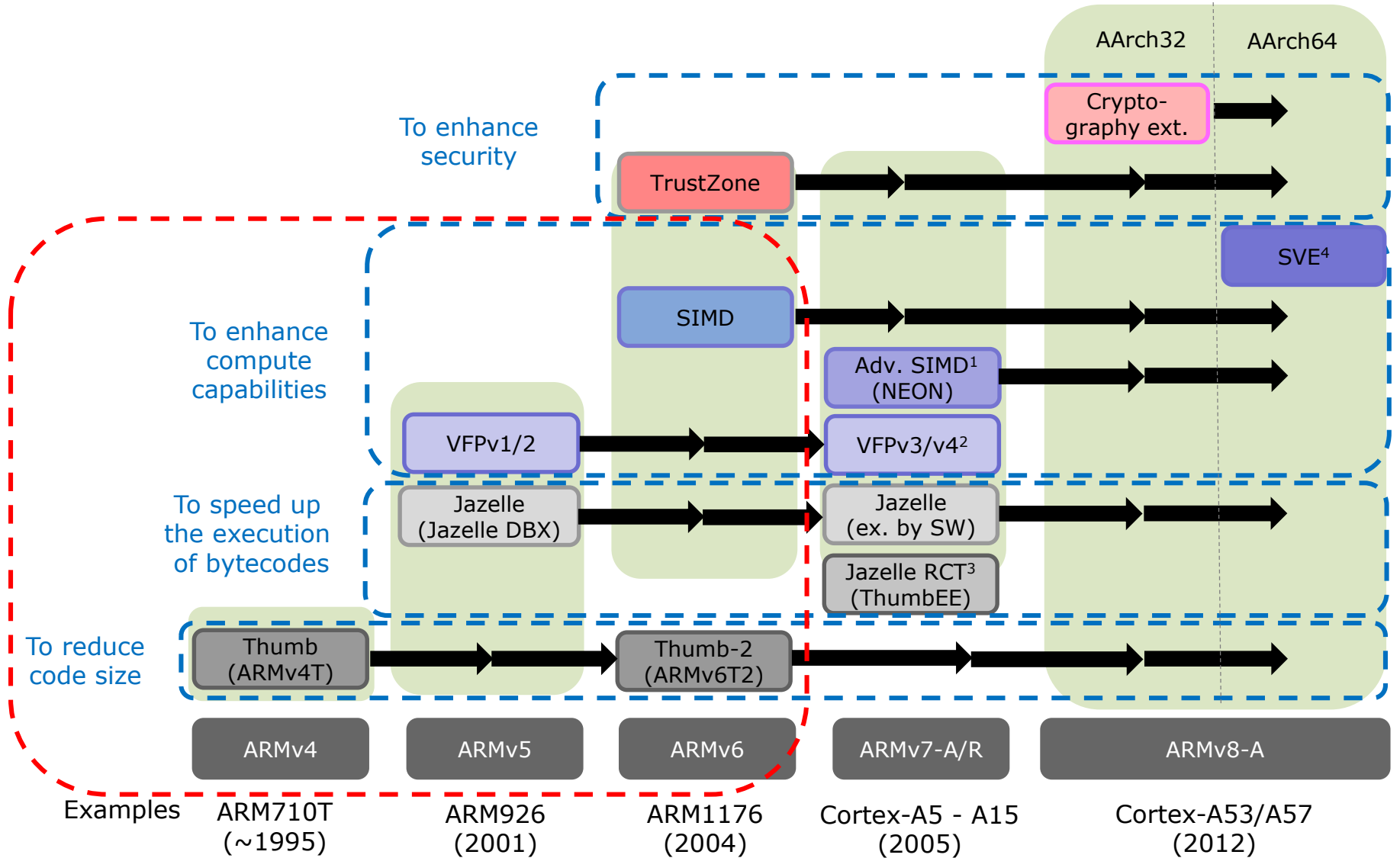
3.3 Processors implementing the ARM v3 - ARM v6 ISA (1)

3.3 Processors implementing the ARMv3 – ARMv6 ISA [5]



3.3 Processors implementing the ARM v3 - ARM v6 ISA (2)

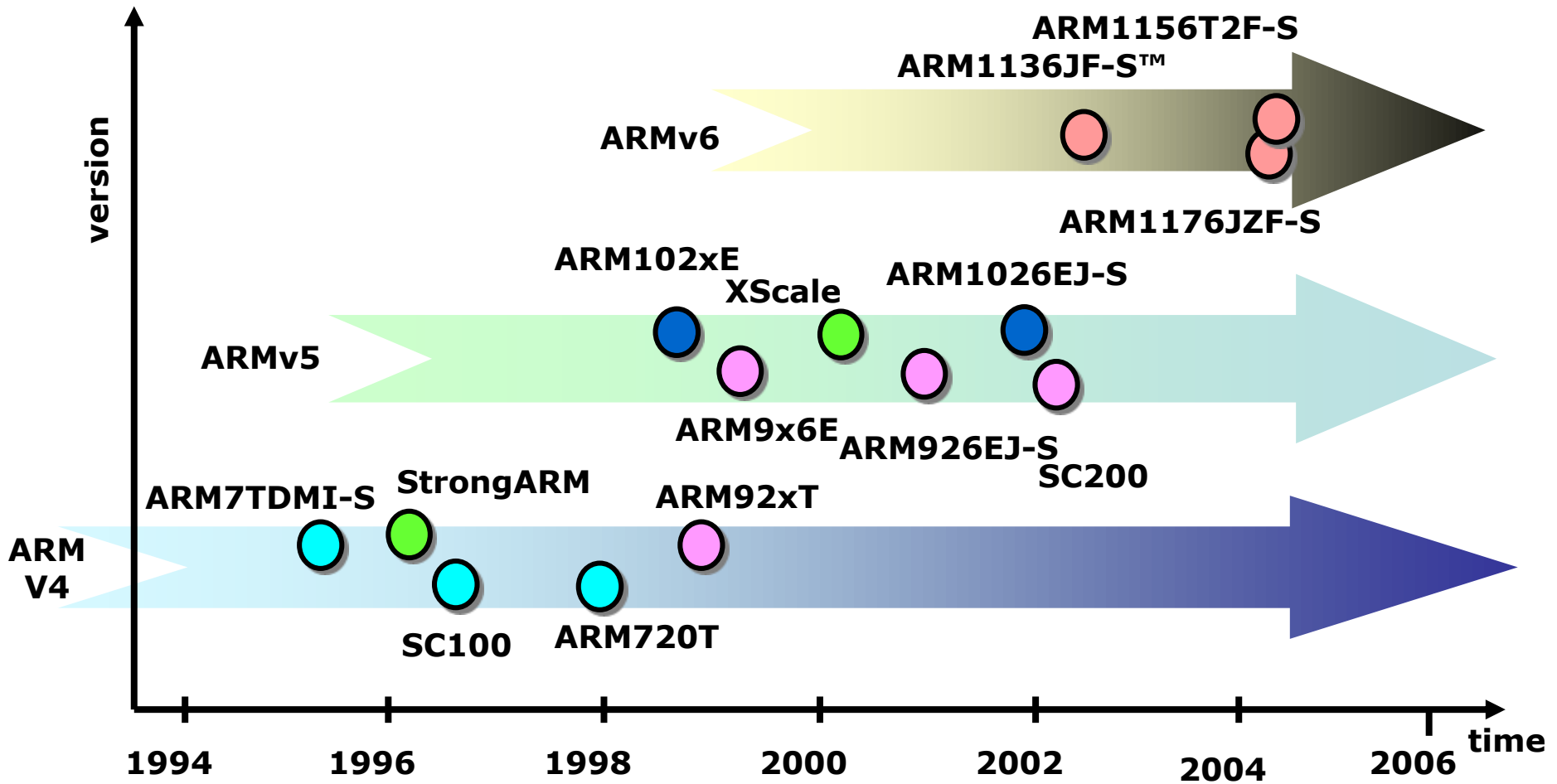
Main extensions introduced in the ARMv4 – ARM v6 ISA versions (simplified)



Remarks: See on the slide 2.1 Overview (5).

3.3 Processors implementing the ARM v3 - ARM v6 ISA (3)

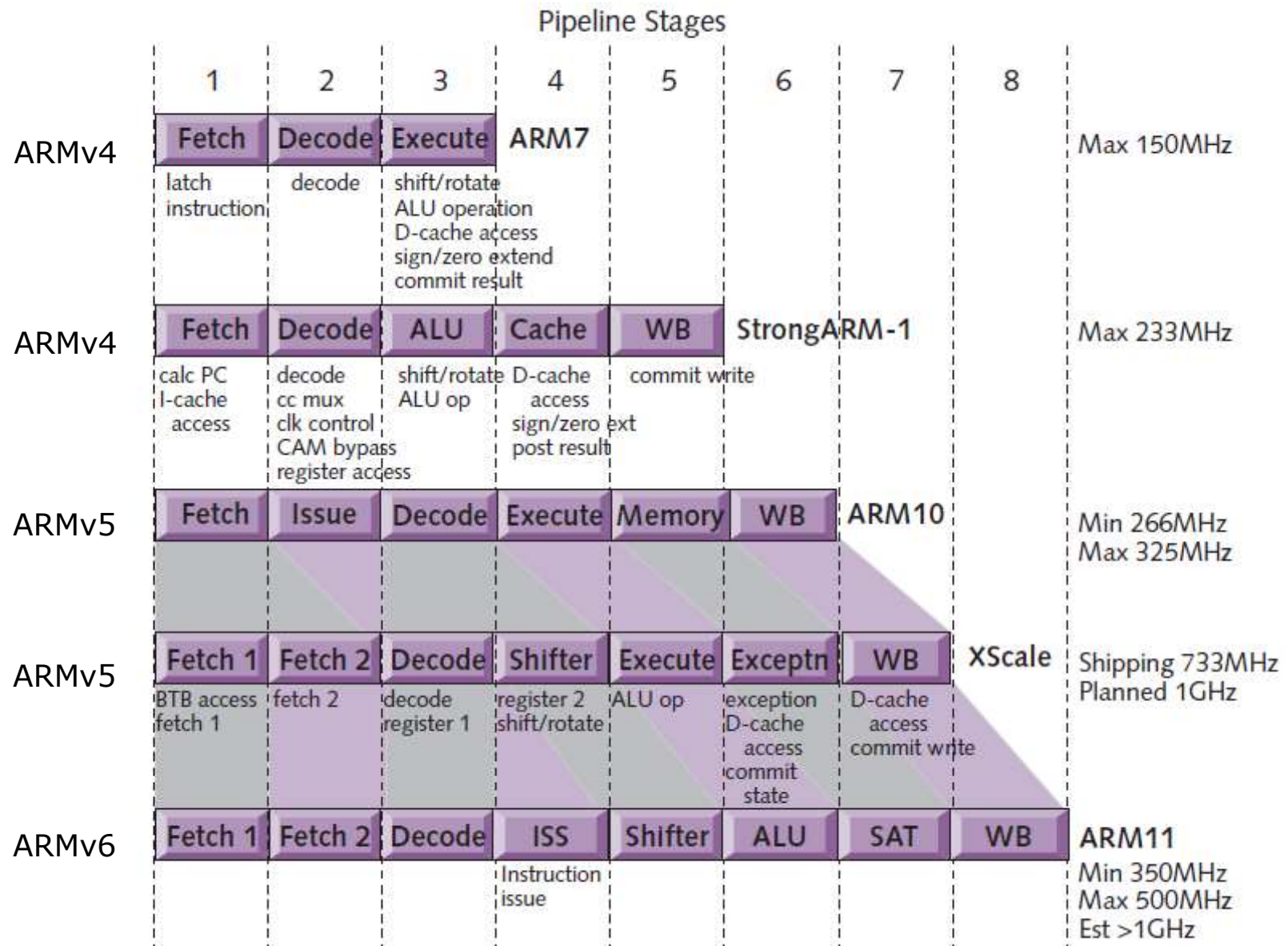
Overview of ARM's processors implementing the ARMv4 – ARMv6 ISA [5]



XScale is a trademark of Intel Corporation

3.3 Processors implementing the ARM v3 - ARM v6 ISA (4)

Evolution of the pipeline length of ARM v4 – ARM v6 based processors [8]



3.3 Processors implementing the ARM v3 - ARM v6 ISA (5)

Main features of the ARMv5 – ARMv6 processors (\approx 1999-2003) [57]

Feature	ARM9E™	ARM10E™	Intel® XScale™	ARM11™
Architecture	ARMv5TE(J)	ARMv5TE(J)	ARMv5TE	ARMv6
Pipeline Length	5	6	7	8
Java Decode	(ARM926EJ)	(ARM1026EJ)	No	Yes
V6 SIMD Instructions	No	No	No	Yes
MIA Instructions	No	No	Yes	Available as coprocessor
Branch Prediction	No	Static	Dynamic	Dynamic
Independent Load-Store Unit	No	Yes	Yes	Yes
Instruction Issue	Scalar, in-order	Scalar, in-order	Scalar, in-order	Scalar, in-order
Concurrency	None	ALU/MAC, LSU	ALU, MAC, LSU	ALU/MAC, LSU
Out-of-order completion	No	Yes	Yes	Yes
Target Implementation	Synthesizable	Synthesizable	Custom chip	Synthesizable and Hard macro
Performance Range	Up to 250MHz	Up to 325MHz	200MHz – >1GHz	350MHz - >1GHz

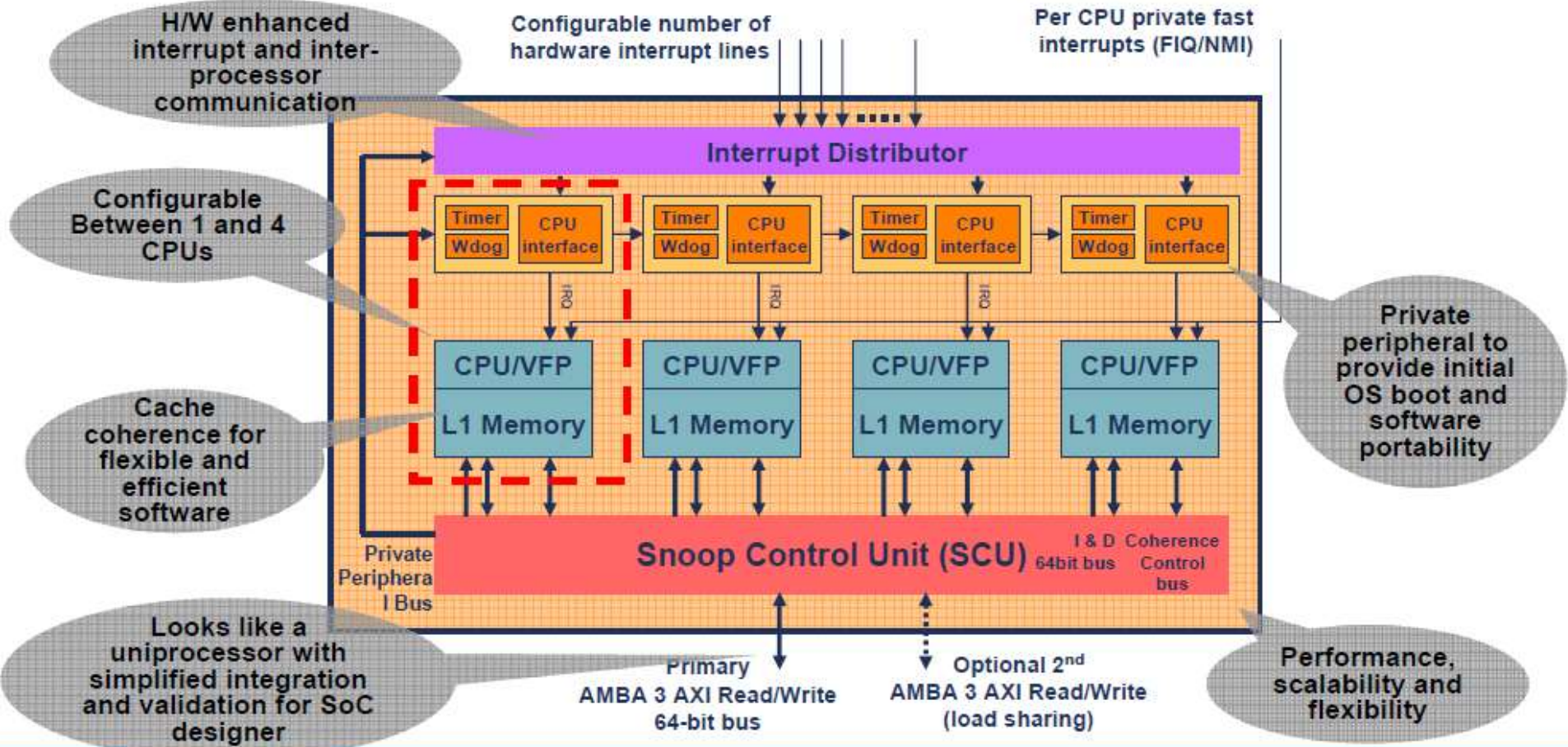
Introduction of multicore processors by ARM: the ARM11 MPCore

- 07/2003: First public disclosure of the ARM11 MPCore.
- 05/2004: **ARM11 MPCore** (**multi core ARM11 with 1-4 cores**) available for licensing with an evaluation system for early software development.
- It is basically an **up to four way (4-socket) cache coherent symmetric multicore processor**
- The ARM11 MPCore incorporates:
- **IEM (Intelligent Energy Manager)**

It dynamically predicts the required performance and lowers the voltage and the frequency accordingly.

3.3 Processors implementing the ARM v3 - ARM v6 ISA (7)

Block diagram of the ARM11 MPCore [9]



Note that the ARM11 MPCore **does not include an L2**.

3.3 Processors implementing the ARM v3 - ARM v6 ISA (8)

Remark: Emergence of multi core processors

Year of launching	Dual core design
10/2001	IBM launches dual core POWER4
11/2002	IBM launches dual core POWER4+
03/2004	Sun releases the UltraSPARC IV (Jaguar) dual core processor
05/2004	ARM announces the availability of the synthetizable ARM11 MPCore quad core processor
05/2004	IBM launches dual core POWER5
08/2004	AMD demonstrates first x86 dual core (Opteron) processor
04/2005	ARM demonstrates the ARM11 MPCore quad core test chip in cooperation with NEC
04/2005	Intel launches dual core Pentium processors (Pentium D)
04/2005	AMD launches dual core Opteron server processors
06/2006	Intel launches their Core 2 line of multicore processors
10/2007	ARM launches their first multicore Cortex model (the quad core Cortex A9 MPcore)

3.4 Processors implementing the ARM v7 - ARM v8 ISA

3.4 Processors implementing the ARMv7 - ARMv8 ISA

- Processors implementing the ARMv7 or ARMv8 ISA are designated **Cortex family** processors.
- The **Cortex-family** along with its first member, the **Cortex-M3** processor was announced in **10/2004**, without disclosing the ARMv7 ISA.
- AMD revealed the **technical specification of the ARMv7 ISA** finally in **3/2005**.

Profiles of ARM's Cortex family [6], [7], [113]

Along with revealing the technical specifications for the **ARMv7 ISA** in 3/2005 ARM introduced **three family profiles** to optimize its Cortex family processors for specific market segments:

The Cortex-A (application) profile

- It aims at **application processors for complex OS and user applications**, like processors in smartphones, tablets, netbooks, eBook readers etc.
- The Cortex-A family supports a **Virtual Memory System Architecture** based on a **Memory Management Unit (MMU)**.
- It supports **three instruction sets**:
 - the **A64** (AArch64),
 - the **A32** (AArch32) and
 - the **T32** (Thumb32)

instruction set.

The Cortex-R (Real-time) profile [113]

- It marks **embedded processors for real time applications**, like mass storage or printer controllers.
- It implements a **Protected Memory System Architecture (PMSA)** based on a **Memory Protection Unit (MPU)**.
- Supports the **A32 and T32** instruction sets.

The Cortex-M (Microcontroller) profile [113]

- Processors of the M profile are optimized **for deeply embedded processors aimed at microcontroller and cost sensitive applications**, like automotive body electronics, or smart sensors.
- Supports **writing interrupt handlers in high-level languages**.
- It implements a **variant of the Protected Memory System Architecture (PMSA)**.
- It supports a variant of the **T32** instruction set.

Extending the ARM Cortex family with the SecurCore profile in 10/2007

- In 10/2007 ARM introduced the **SecurCore profile**.
- It is optimized for smart card and secure applications.

3.4 Processors implementing the ARM v7 - ARM v8 ISA (5)

Recent ARM profiles and processors [6]

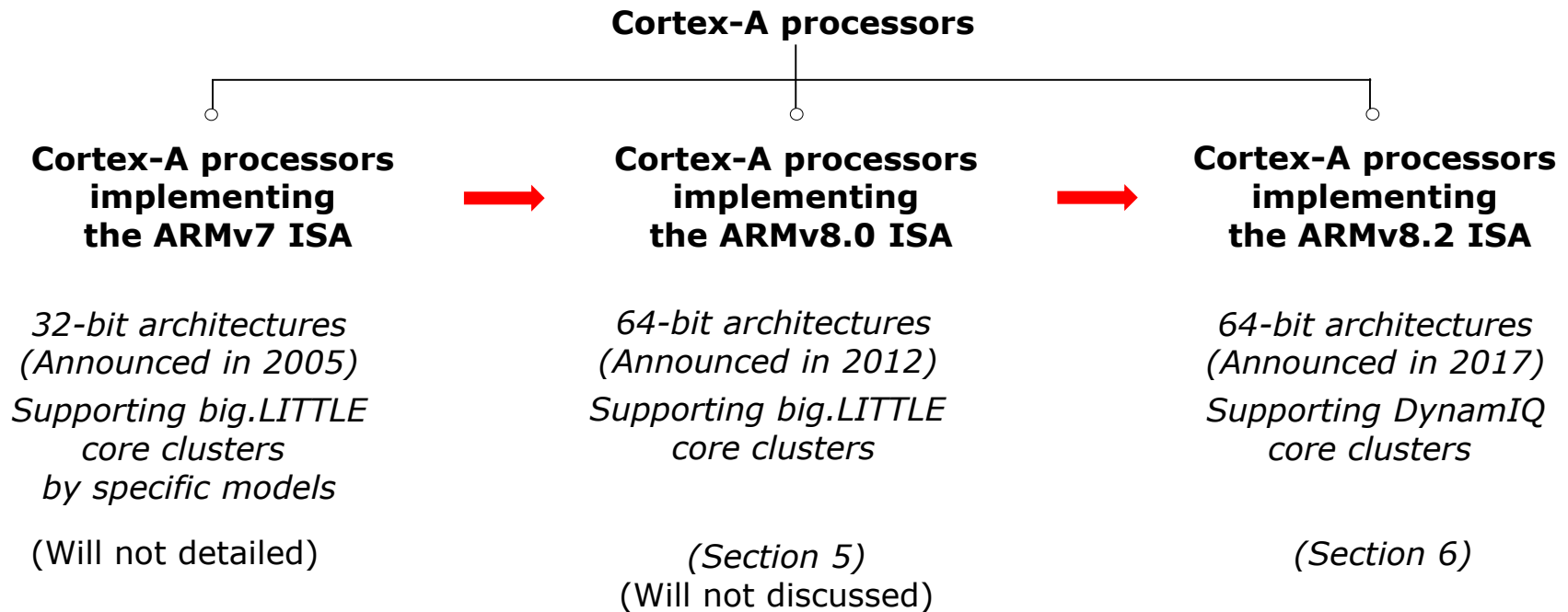
CORTEX-A	Cortex-A72
	Cortex-A57
	Cortex-A53
	Cortex-A17
	Cortex-A15
	Cortex-A9
	Cortex-A7
	Cortex-A5
CORTEX-R	Cortex-R7
	Cortex-R5
	Cortex-R4
CORTEX-M	Cortex-M7
	Cortex-M4
	Cortex-M3
	Cortex-M1
	Cortex-M0+
	Cortex-M0
SECURCORE	SC000
	SC100
	SC300

Remark

According to the general scope of this Lecture Notes, subsequently we will be concerned only with the Cortex-A series.

Overview of the evolution of the Cortex-A series processors

- In 2012 ARM expanded their Cortex family by processors implementing the **64-bit ARMv8 ISA**, and subsequently
- in 5/2017 ARM enhanced its ARMv8 ISA by the **ARMv8.2 revision** and introduced first models implementing the new ISA release (the Cortex-A55 and Cortex-A75 models, as indicated below).



Note

Subsequently, we designate the **original ARMv8 ISA** release as the **ARMv8.0 ISA**.

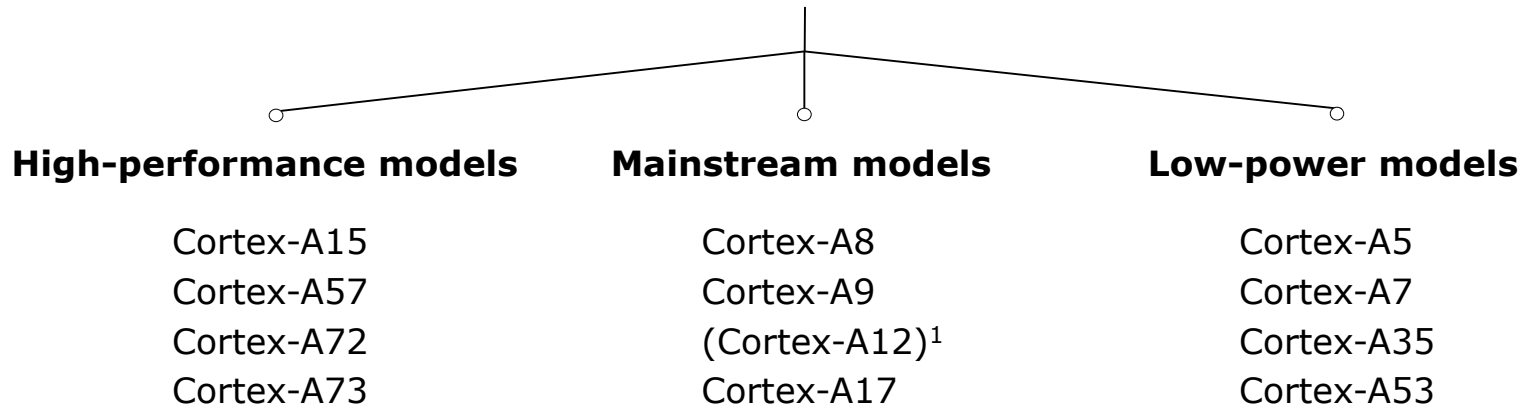
4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (1)

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models

Performance classes in the Cortex-A processors (only ARMv7/8.0 ISA-based processors) -1

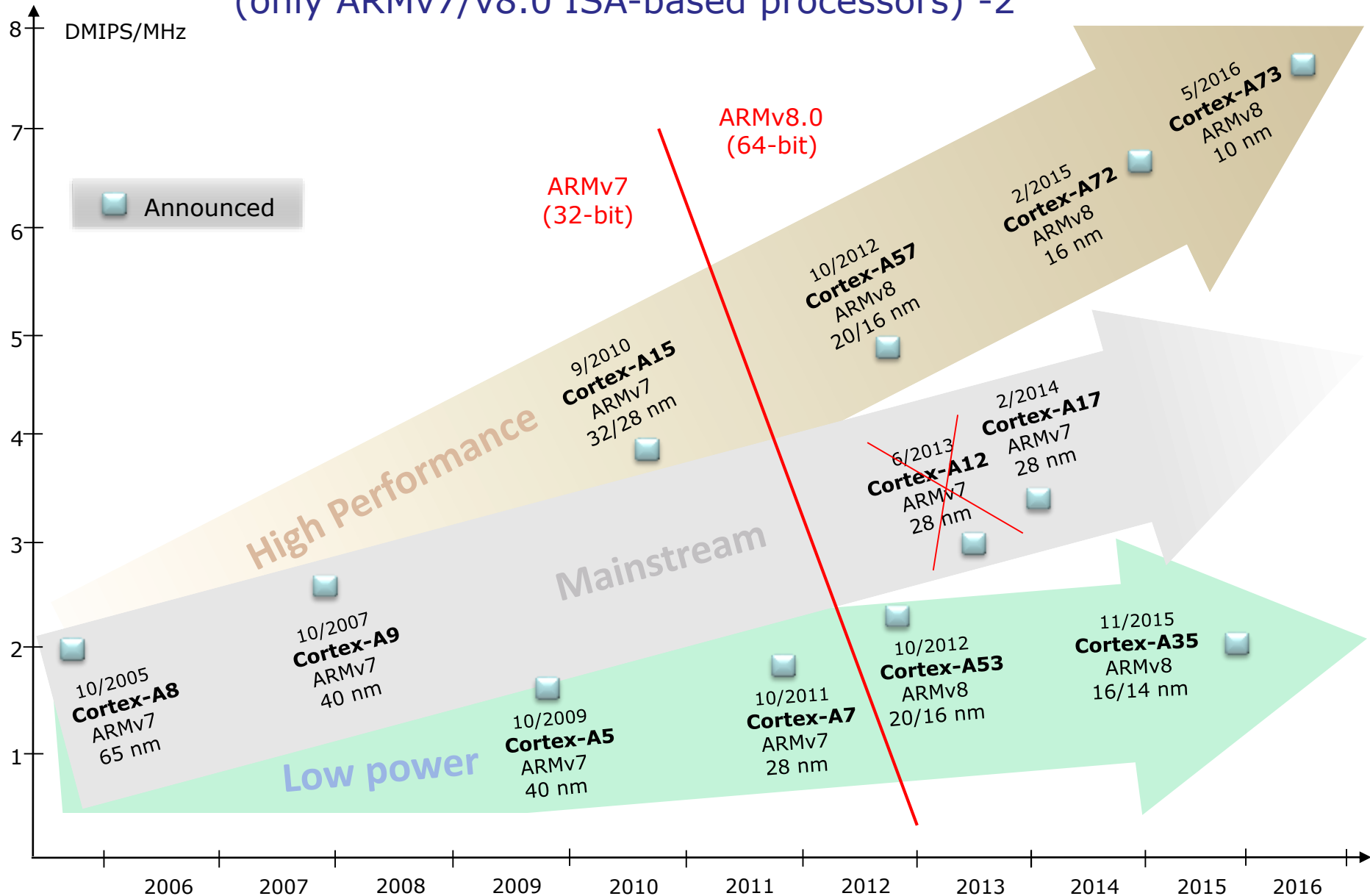
Performance classes of the Cortex-A series processors implementing the ARMv7/8.0 ISA



¹The model Cortex-A-12 was introduced in 6/2013 but became withdrawn in 10/2014 since its parameters were too close to that of the Cortex-A17, so subsequently we will leave it out in most parts from our discussion.

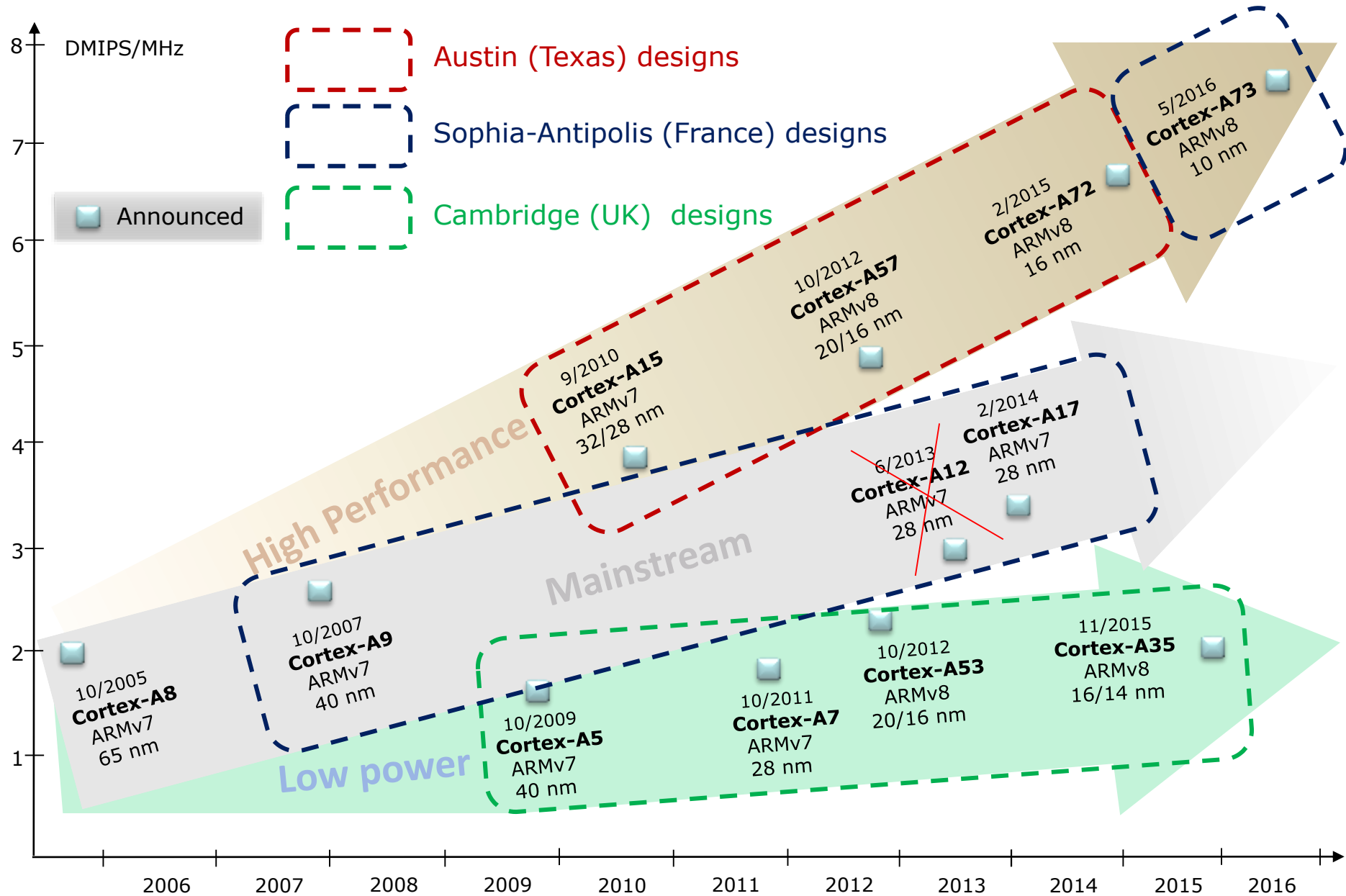
4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (2)

Performance classes in the Cortex-A series (only ARMv7/v8.0 ISA-based processors) -2



4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (3)

Three design teams working in parallel on Cortex-A processors []



4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (4)

Remarks on the designs of the Cortex-A series (taken from [99])

"...The A15, A57, A72 all belong to the **Austin family** of microarchitectures, and as one would have guessed from the name, this is because they originated from ARM's Austin CPU design centre.

The A5, A7 and A53 belong to the **Cambridge family** while the Cortex A12, A17 and today's new A73 belong to the **Sophia family**, owing its name to the small city of Sophia-Antipolis which houses one of Europe's largest technology parks as well as ARM's French CPU design centre. Referring to their design location is however not enough to disambiguate microprocessor families, as we'll see completely new designs come out from each R&D center. In fact, this has already happened as the A12/17/73 can be seen as a new generation over the preceding the A9 microarchitecture and as such can be referred to as a "second generation Sophia family". This is an important notion to consider as in the future we'll be seeing completely new microarchitectures come out of ARM's various design teams."

"The **Cortex A73** being still in the same **Sophia family** effectively means the design is very much a 64-bit successor to the Cortex A17. The new core effectively inherits some of the main characteristics of its predecessor such as overall μ arch philosophy as well as higher-level pipeline elements and machine width. And herein lies the biggest surprise of the Cortex A73 as a A72 successor: Instead of choosing to maintain A72's 3-wide, or increase the microarchitecture's decoder width, ARM opted to instead go back to a 2-wide decoder such as found on the current Sophia family. Yet the A73 positions itself a higher-performance and lower-power design compared to the larger A72."

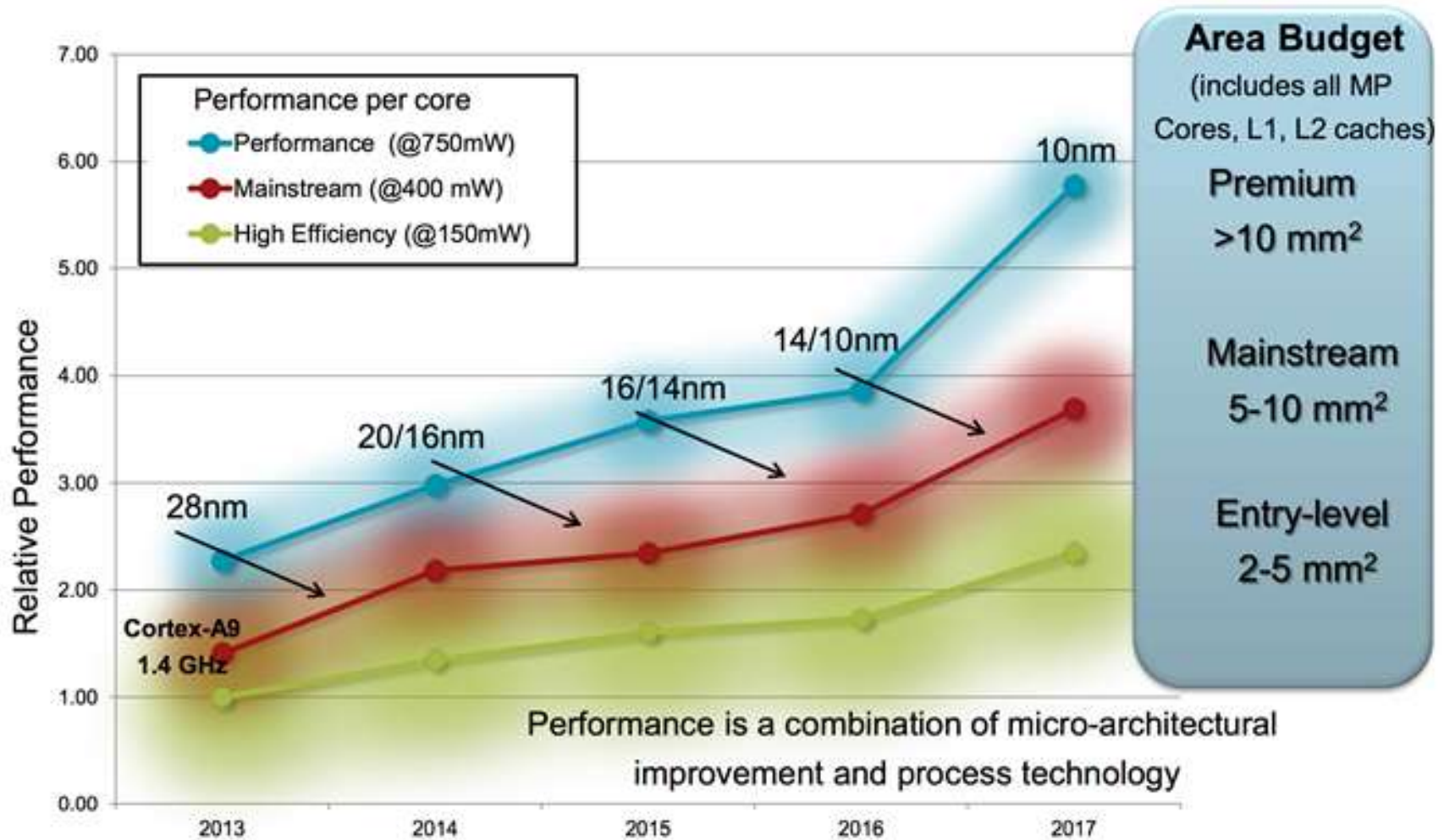
Addendum

The subsequent Cortex-A models were designed by the following teams:

- The **Cortex-A76** by the **Sophia-Antipolis (France)** team and
- the **Cortex A-55** by the **Cambridge (UK)** team.

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (5)

Relative per core performance of Cortex-A processors with different power budget [16]



4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (6)

Announcement dates and efficiency (DMIPS/MHz) of the Cortex-A models implementing the ARMv7 and ARMv8.0 ISA

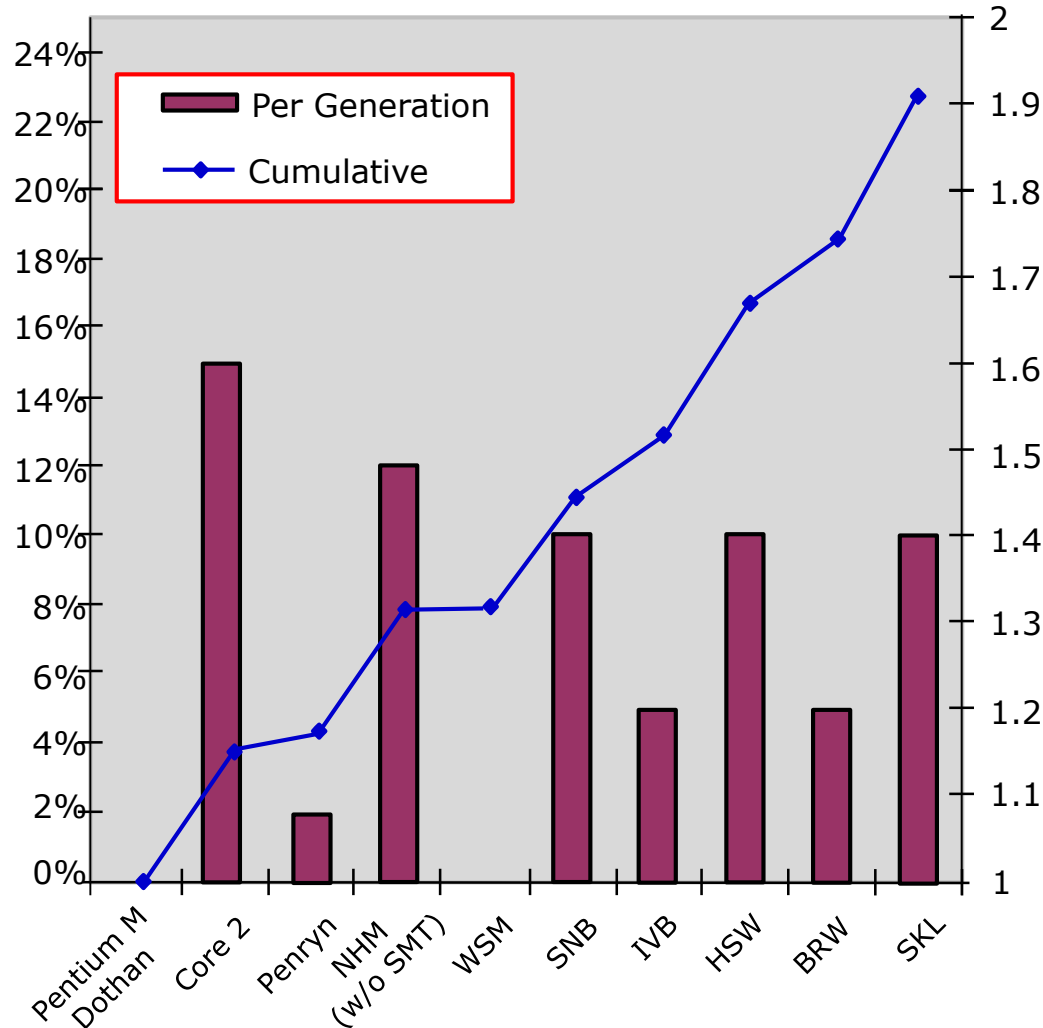
Announced	Cortex-A model	DMIPS/MHz
10/2005	Cortex-A8	2.0
10/2007	Cortex-A9/A9 MPCore	2.5
10/2009	Cortex-A5/A5 MPCore	1.6
9/2010	Cortex-A15/A15 MPCore	3.5-4.0
10/2011	Cortex-A7 MP Core	1.7
6/2013	(Cortex-A12/A12 MPCore)	3.0
2/2014	Cortex-A17/A17 MPCore	3.1-3.3
11/2015	Cortex-A35 MPCore	~ 2.1
10/2012	Cortex-A53 MPCore	2.3
10/2012	Cortex-A57 MPCore	4.1-4.7
2/2015	Cortex-A72 MPCore	6.3-7.35
5/2016	Cortex-A73 MPCore	7.4-8.5

Source: ARM

DMIPS: Dhrystone MIPS (A synthetic benchmark that indicates integer performance)

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (6b)

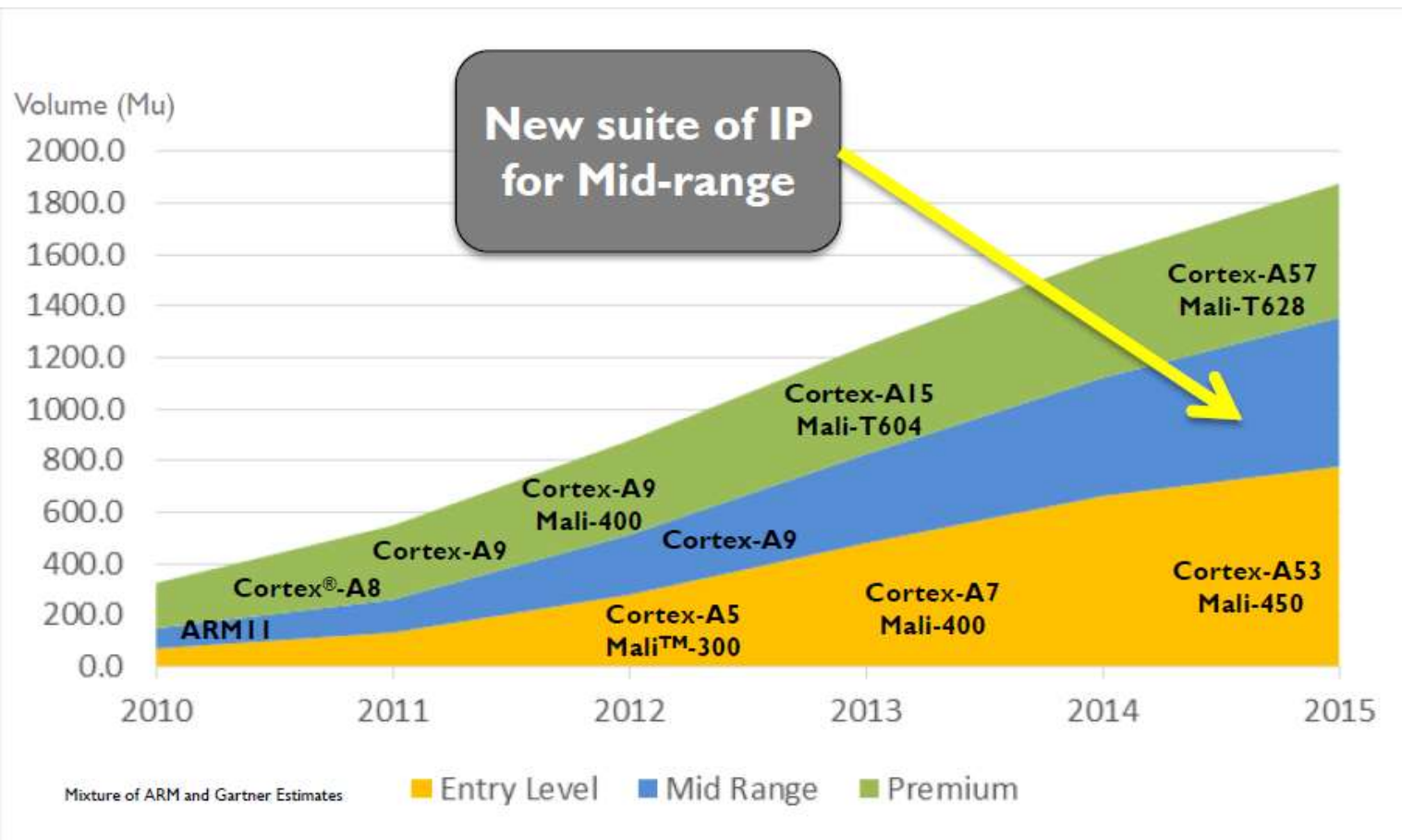
Single thread IPC in Intel's basic architectures (Based on [124])



Note that Intel raised IPC in the Core family **only less than 2-times** in about 10 years, whereas **increased the efficiency of the Cortex-A line by more than 3-times**. *

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (7)

Yearly shipment of ARM Cortex-A chips in different performance classes
(data from 2013) [13]



Relative
SoC Die
Sizes

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (8)

Target markets of the 64-bit Cortex-A series [49]

Entry-level Computing

Extend OS capabilities to sub-\$100 devices



'Desktop Class' Computing

Performance apps
Enhanced multimedia processing



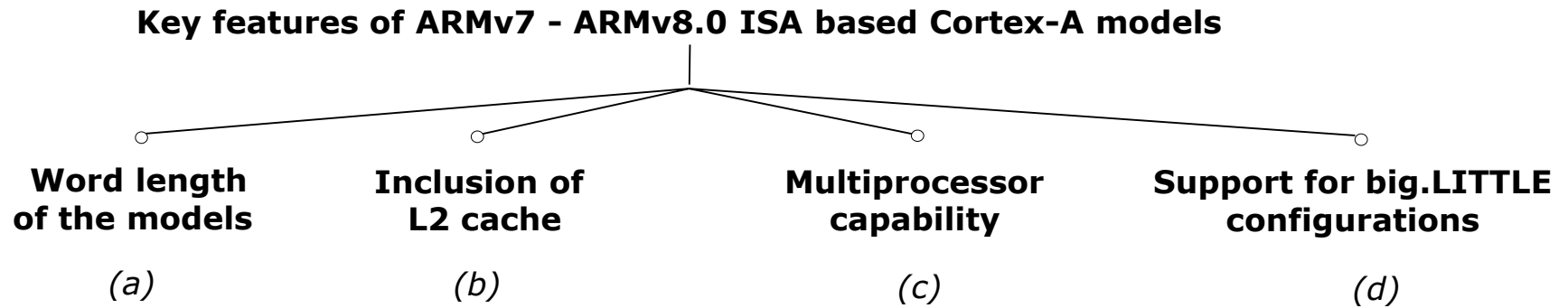
High-end Enterprise

64-bit memory addressing
Virtualisation
High bandwidth
Enable innovation for hyperscale operators



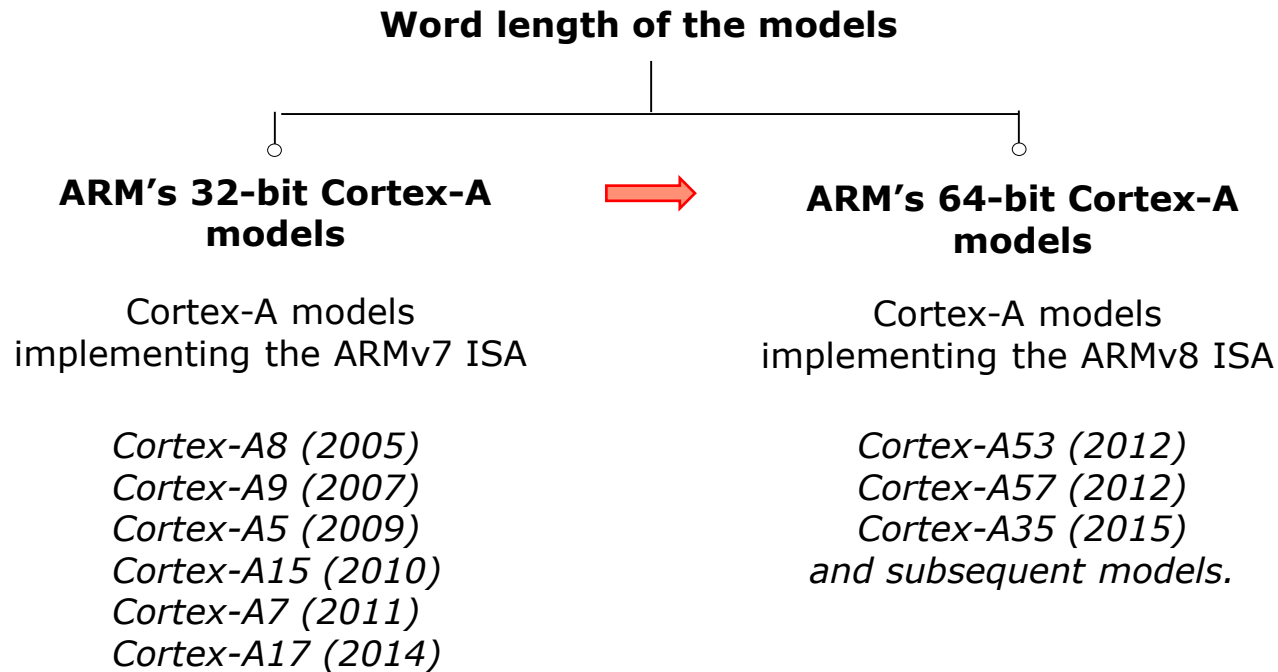
4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (9)

Key features of ARMv7 - ARMv8.0 ISA based Cortex-A models



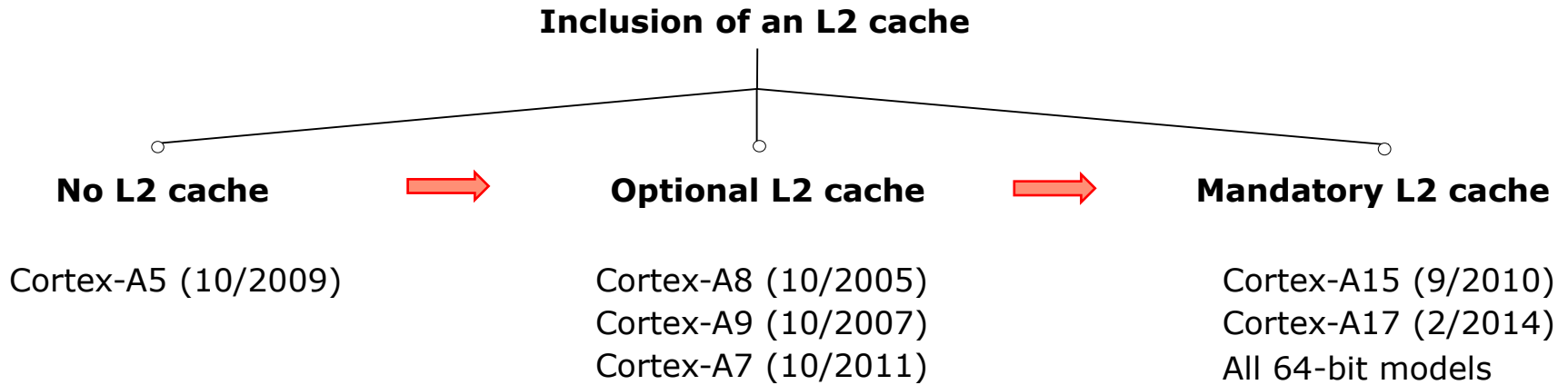
4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (10)

a) Word length of the models



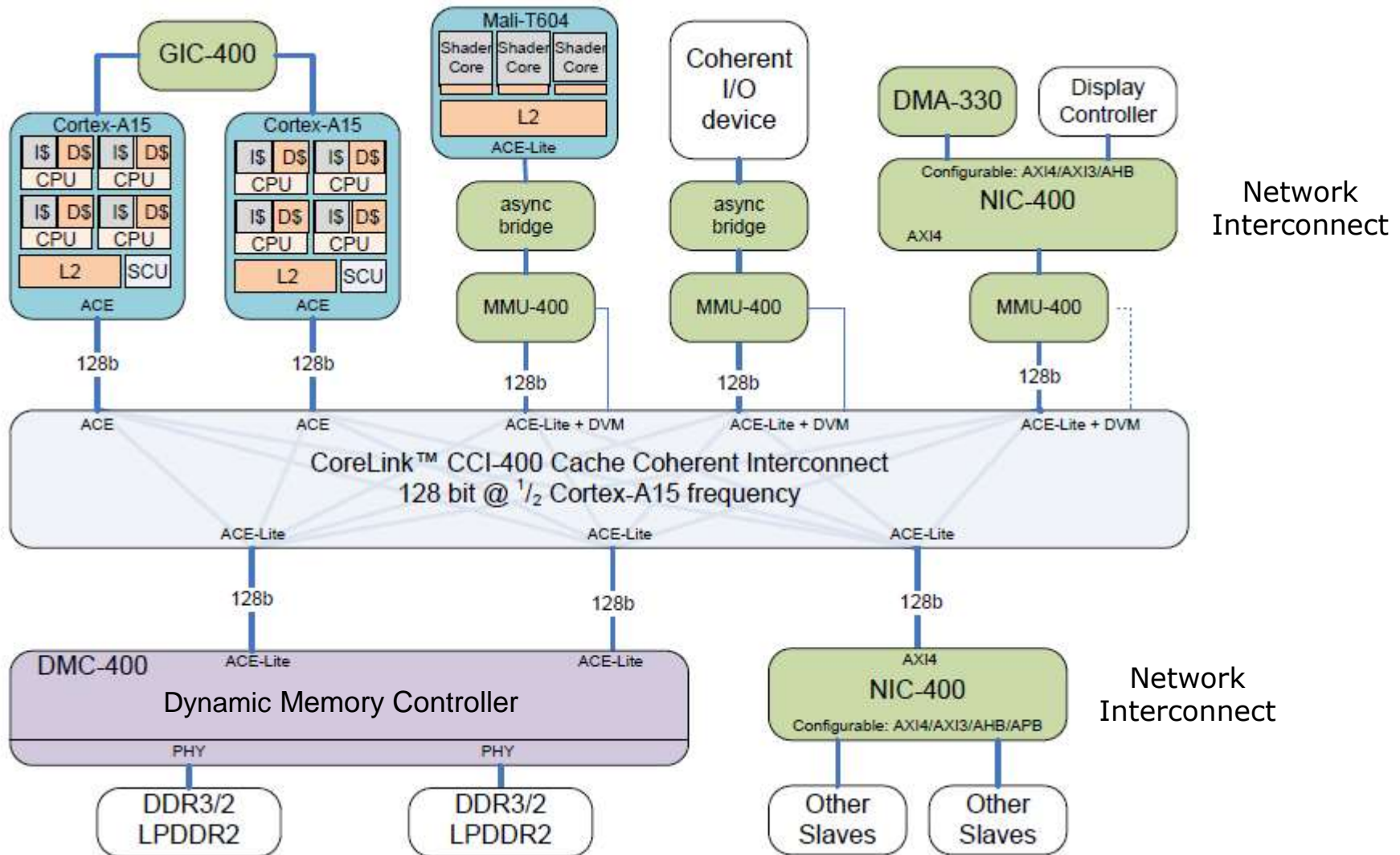
4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (11)

b) Inclusion of an L2 cache



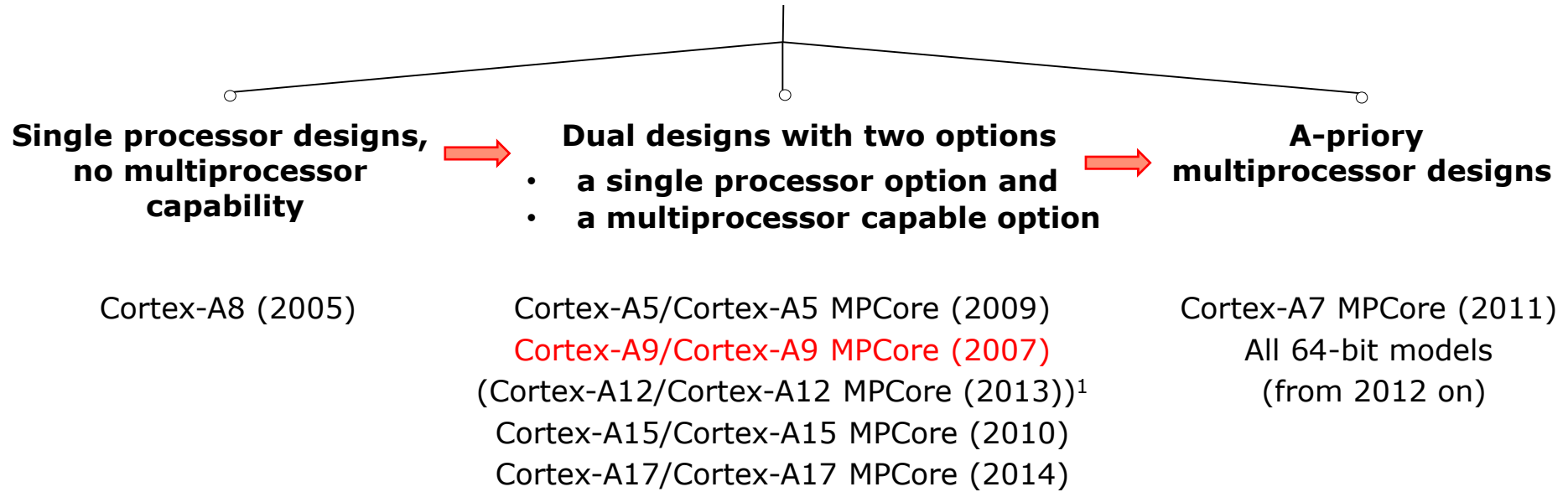
4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (12)

Example for including an L2 cache : Eight-core A15-based high performance cache coherent system [15]



c) Multiprocessor capability

Multiprocessor capability of the Cortex-A series implementing the ARMv7/v8.0 ISA



Here we note that in figures or tables we often omit the MPCore tag for the sake of brevity.

¹The model Cortex-A12 was introduced in 6/2013 but became withdrawn in 10/2014 since its parameters were too close to that of the Cortex-A17, so subsequently we will leave it out in most parts from our discussion.

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (14)

Remarks on the interpretation of the term MPCore by ARM

- ARM introduced the term **MPCore** in connection with the announcement of the **ARM11 MPCore in 2004** despite the fact that the ARM11 MPCore was actually a **multicore processor** including up to 4 cores **rather than a multiprocessor**.
- **Along with the ARM Cortex-A9 MPCore (2007)** ARM re-interpreted this term.

The ARM Cortex-A9 introduced **1-4 cores** and it had two alternatives

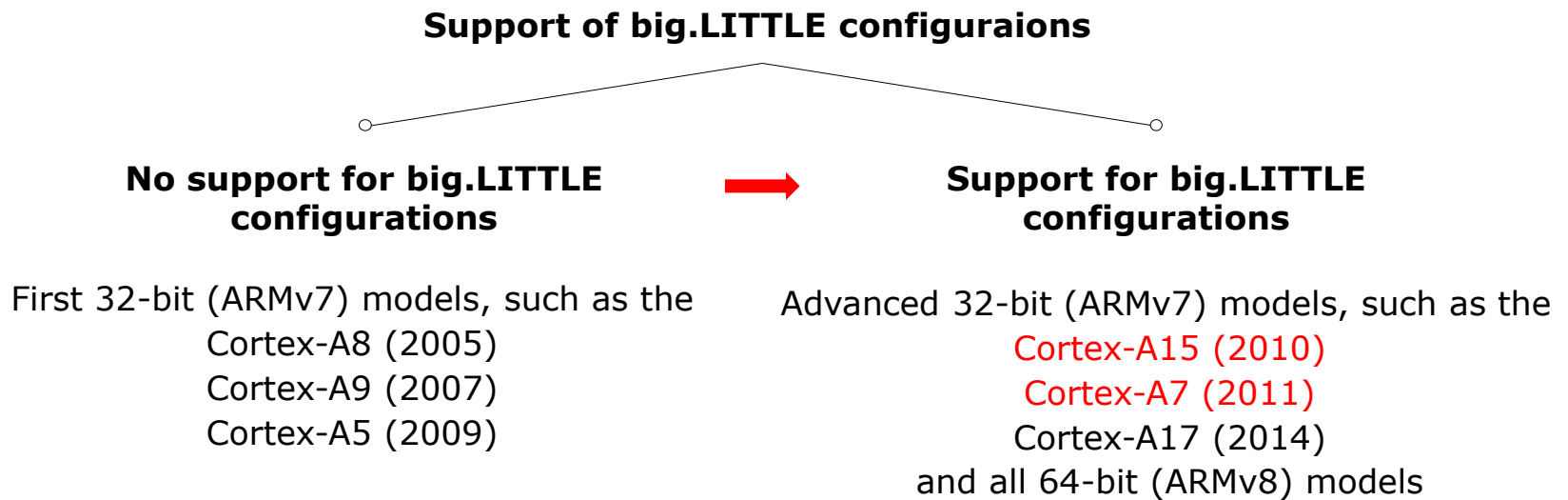
- **ARM Cortex-A9** it does not support multiprocessor configurations and
- **ARM Cortex-A9 MPCore** it supports multiprocessor configurations.

Since then MPCore indicates the **multiprocessor capability** of the processor.

d) Support for big.LITTLE configurations

ARM revealed the **big.LITTLE technology** in a White Paper, in **09/2011**.

Accordingly, ARM began supporting the big.LITTLE technology along with their **advanced 32-bit (ARMv7) designs**, as shown below.



4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (16)

Main features of the Cortex A series implementing the ARMv7 ISA [51]

CPU Core	Architecture	Efficiency	big.LITTLE	Announced	Available in devices	Target
Cortex-A17	ARMv7 (32-bit)	4,0 DMIPS/MHz	Yes (with A7)	2014	2015	Mainstream
Cortex-A15	ARMv7 (32-bit)	4,0 DMIPS/MHz	Yes (with A7)	2010	Q2/2013	High-end
(Cortex-A12)	ARMv7 (32-bit)	3,0 DMIPS/MHz	--	2013	H2/2015	Mainstream)
Cortex-A9	ARMv7 (32-bit)	2,5 DMIPS/MHz	--	2007	2010	Mainstream
Cortex-A8	ARMv7 (32-bit)	2,0 DMIPS/MHz	--	2005	2009	Mainstream
Cortex-A7	ARMv7 (32-bit)	1,9 DMIPS/MHz	Yes (A15/A17)	2011	2012	Low power
Cortex-A5	ARMv7 (32-bit)	1,6 DMIPS/MHz	--	2009	2011	Low power

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (17)

Main features of the Cortex A series implementing the ARMv8.0 ISA [51]

CPU Core	Architecture	Efficiency	big.LITTLE	Announced	Available in devices	Target
Cortex-A73	ARMv8 (64-bit)	7.4-8.5 DMIPS/MHz	Yes (with A53/A35)	2016	2017	High-end
Cortex-A72	ARMv8 (64-bit)	6.3-7.3 DMIPS/MHz	Yes (with A53/A35)	2015	2016	High-end
Cortex-A57	ARMv8 (64-bit)	4,8 DMIPS/MHz	Yes (with A53)	2012	2015	High-end
Cortex-A53	ARMv8 (64-bit)	2,3 DMIPS/MHz	Yes (with A57)	2012	H2/2014	Low power
Cortex-A35	ARMv8 (64-bit)	2,1 DMIPS/MHz	Yes (with A57/A72)	2015	H2/2016	Low power

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (17b)

Main features of the Cortex A series implementing the ARMv8.2 ISA [51]

CPU Core	Architecture	Efficiency	big.LITTLE	Announced	Available in devices	Target
Cortex-A76	ARMv8 (64-bit)	~10.7-12.4 DMIPS/MHz	Yes (with A55)	2018	2018	High-end
Cortex-A75	ARMv8 (64-bit)	~8.2-9.5 DMIPS/MHz	Yes (with A55)	2017	2018	High-end
Cortex-A55	ARMv8 (64-bit)	~3 DMIPS/MHz	Yes (with A75/A76)	2017	2018	Low-end

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (18)

Key features of ARMv7 ISA based microarchitectures (based on [14])

High performance

32-bit

2010	Cortex-A15	3 decode	Out-of-Order issue		15-25 Stages	1-4 cores	3.5-4.0 DMIPS/MHz 2+ GHz 32/28/22 nm
------	-------------------	----------	--------------------	--	--------------	-----------	--

Mainstream

2005	Cortex-A8	2 decode	In-Order		13-23 Stages	1 cores	2 DMIPS/MHz 1 GHz 65/55/45 nm
2007	Cortex-A9	2 decode	Out-of-Order issue		8-11 Stages	1-4 cores	2.5 DMIPS/MHz 2+ GHz 65/45/40/32/28 nm
2013	Cortex-A12	2 decode	Out-of-Order issue		10-12 Stages	1-4 cores	3.0 DMIPS/MHz 2+ GHz 28 nm
2014 (Enhanced A12)	Cortex-A17	2 decode	Out-of-Order issue		10-12 Stages	1-4 cores	3.1-3.3 DMIPS/MHz 2+ GHz 28 nm

Low power

2009 (A9 replacement for low-end devices)	Cortex-A5	1 decode	In-Order		8 Stages	1-4 cores	1.6 DMIPS/MHz 1 GHz 40/28 nm
2011	Cortex-A7	2 decode	Out-of-Order issue		8-10 Stages	1-4 cores	1.9 DMIPS/MHz 1.7 GHz 40/28 nm

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (19)

Key features of ARMv8.0 ISA based microarchitectures (based on [14])

High performance

64-bit

2013	Cortex-A57	3 decode	Out-of-Order issue	 8 Pipeline 15+ Stages	1-4 cores	4.1-4.7 DMIPS/MHz Up to 2.0 GHz 28/20/16/14 nm
2015	Cortex-A72	3 decode	Out-of-Order issue	 8 Pipeline 15+ Stages	1-4 cores	6.3-7.35 DMIPS/MHz Up to 2.5 GHz 28/16 nm
2016	Cortex-A73	2 decode	Out-of-Order issue	 7 Pipeline 11 Stages	1-4 cores	7.4-8.5 DMIPS/MHz Up to 2.8 GHz 10 nm

Low power

64-bit



2013	Cortex-A53	2 decode	In-Order	 5 Pipeline 8 Stages	1-4 cores	2.3 DMIPS/MHz 1.2+ GHz 28/20/16/14/10 nm
2015	Cortex-A35	2 decode	In-Order	 6 Pipeline 8 Stages	1-4 cores	~2.1 DMIPS/MHz Up to 2.0 GHz 28/20/16/14/10 nm

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (19b)

Key features of ARMv8.2 ISA based microarchitectures (based on [14])

High performance

64-bit

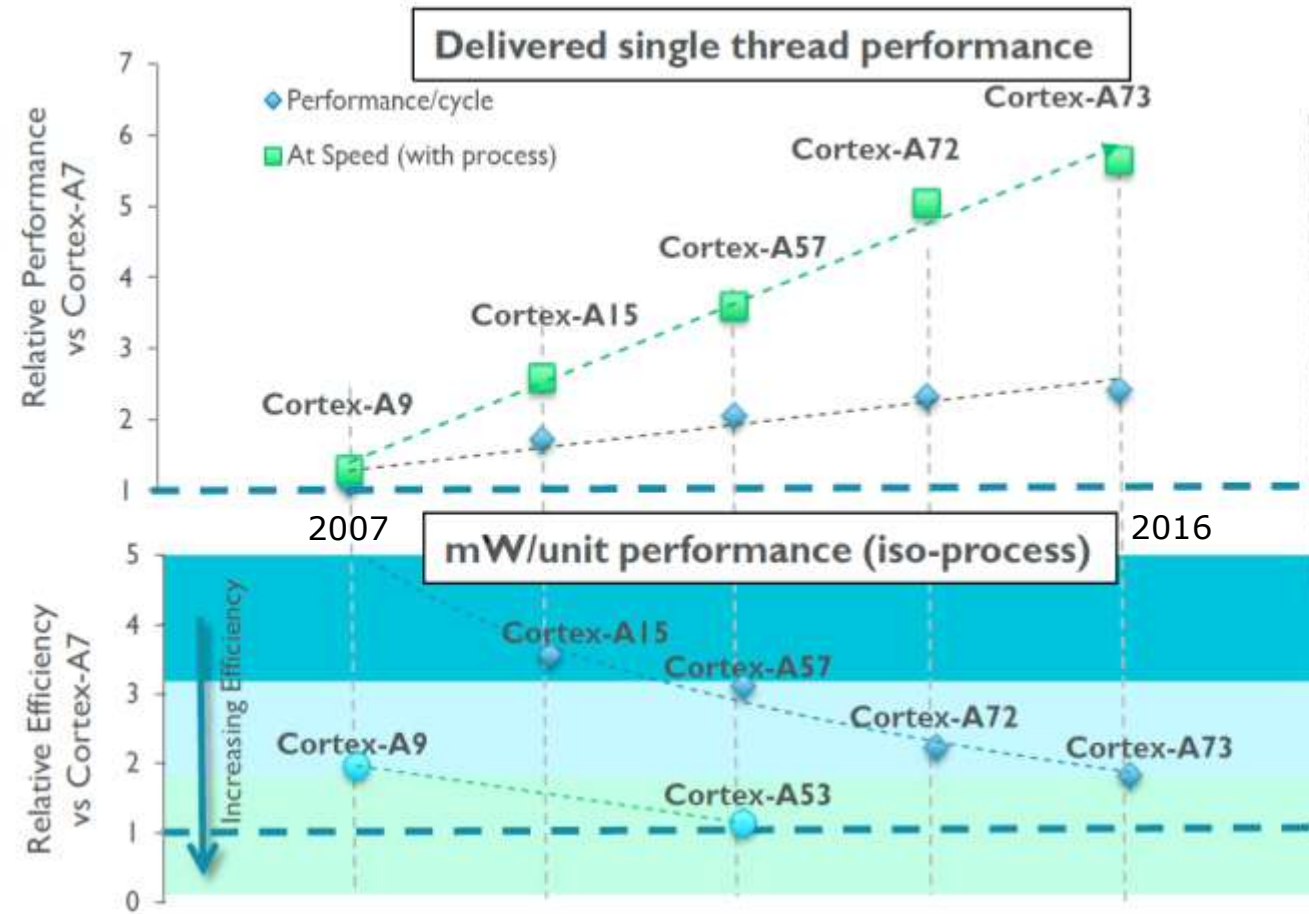
2017	Cortex-A75	3 decode		1-4 cores	≈8.2-9.5 DIMPS/MHz Up to 3 GHz 10 nm
2018	Cortex-A76	4 decode		1-4 cores	≈10.7-12.4 DIMPS/MHz Up to 3 GHz 12/7/5 nm

Low power

2017	Cortex-A55	2 decode		1-4 cores	≈ 3.0 DIMPS/MHz Up to 2.9 GHz 16/14/10/7 nm
------	------------	----------	---	-----------	---

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (20)

Relative performance, efficiency and power efficiency of subsequent models of the Cortex-A series [99]



Accelerated growth in:

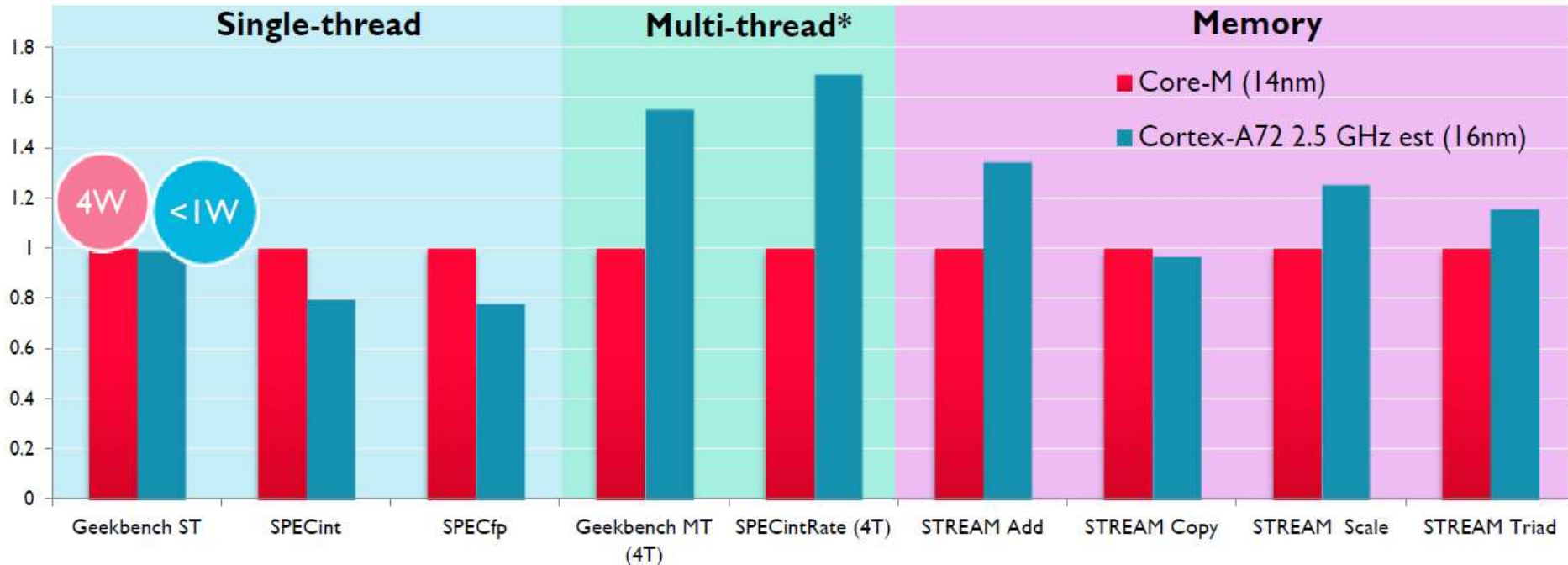
- Performance/Cycle
- Delivered performance

AND

Continuous improvement in power efficiency

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (21)

Performance comparison: ARM's Cortex-A72 vs. Intel's Core-M [72]

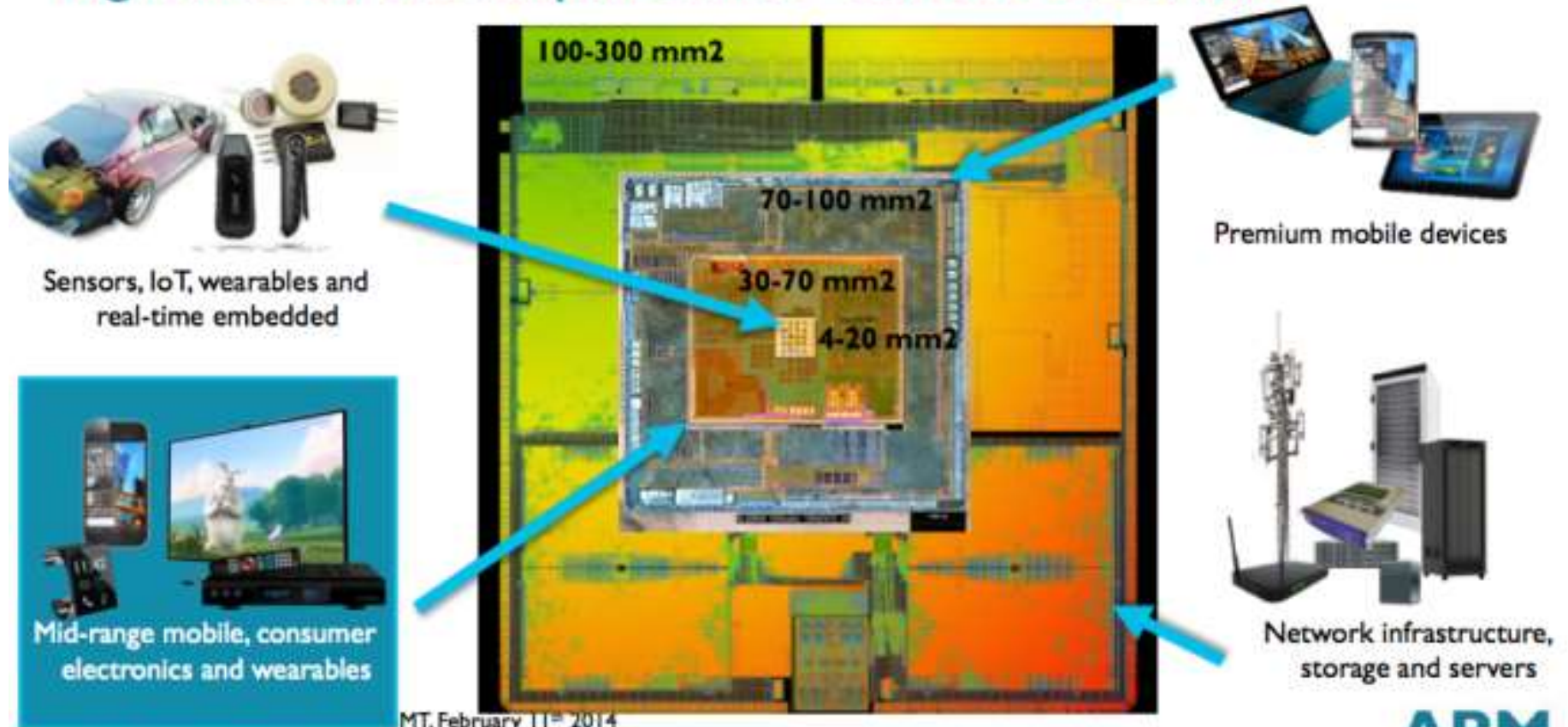


- Intel workloads measured on Dell Venue Pro II. SPEC benchmarks measured using gcc compiler v4.9 with `-o3` flag.
 - Cortex-A72 measured on RTL with realistic memory system with gcc compiler v4.9 - `o3` settings.
 - Multi-threaded workloads use 2C4T Core-M CPU and estimated on 4C Cortex-A72 configuration w/2MB L2 cache.
 - Core-M 5Y10C has maximum rated frequency rating of 2GHz. (Source: ark.intel.com)
- * For multi-threaded workloads, the Core-M will be thermally limited and not able to reach maximum target frequency.

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (22)

Die area requirement of different applications [17]

Right SoC for the Required Task – Sensors to Servers



3 Not to be published without the consent of ARM

ARM

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (23)

Wide configuration options - Example: The 64-bit low power Cortex-A35

- It is a widely configurable application processor, as indicated below.

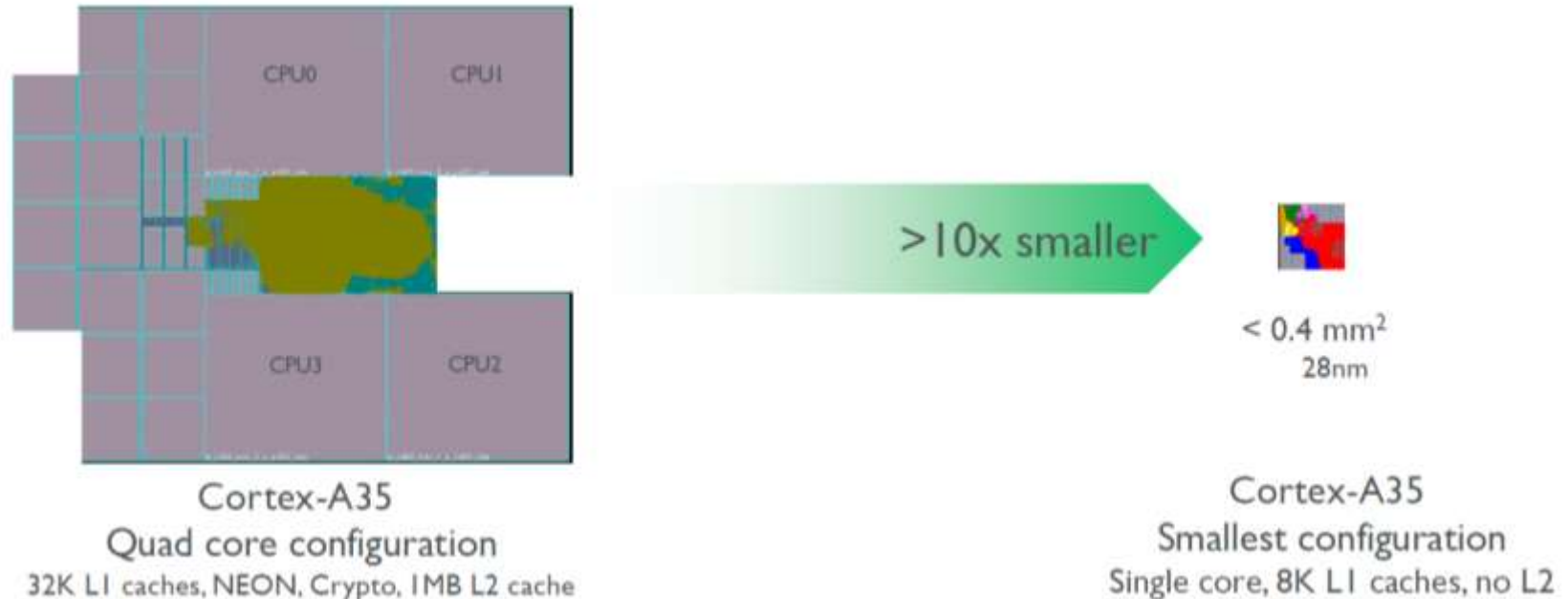
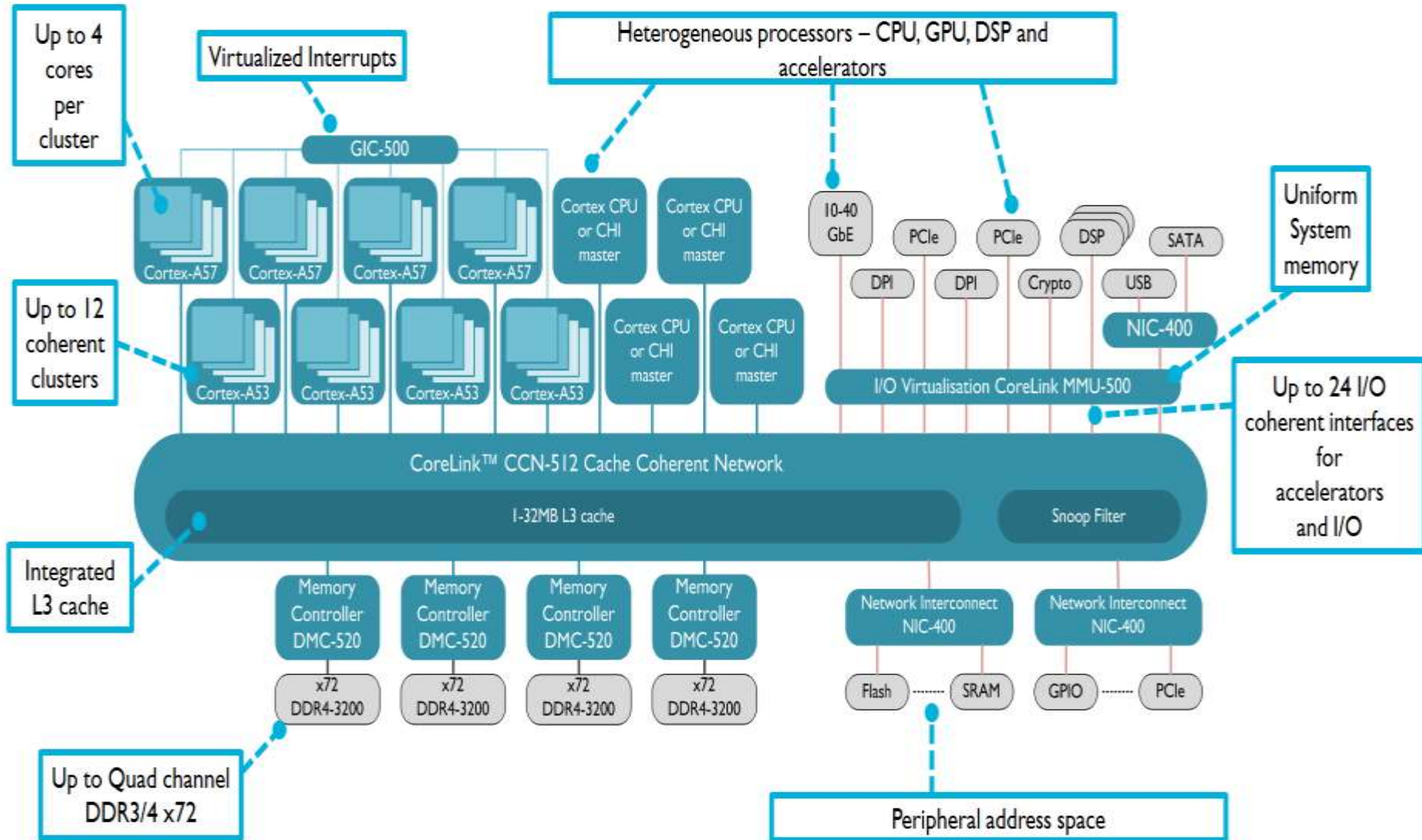


Figure: Configuration options of the Cortex-A35 [77]

- It is configurable for applications ranging from mobiles to deeply embedded.
- Target consumption is below 125 mW.
- On 28 nm technology it achieves 1 GHz by 90 mW, for planned smaller feature sizes of 14/16 nm ARM expects less consumption.

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (24)

Up to 48 core server SoC based on the CoreLink CCN-512 interconnect [72]



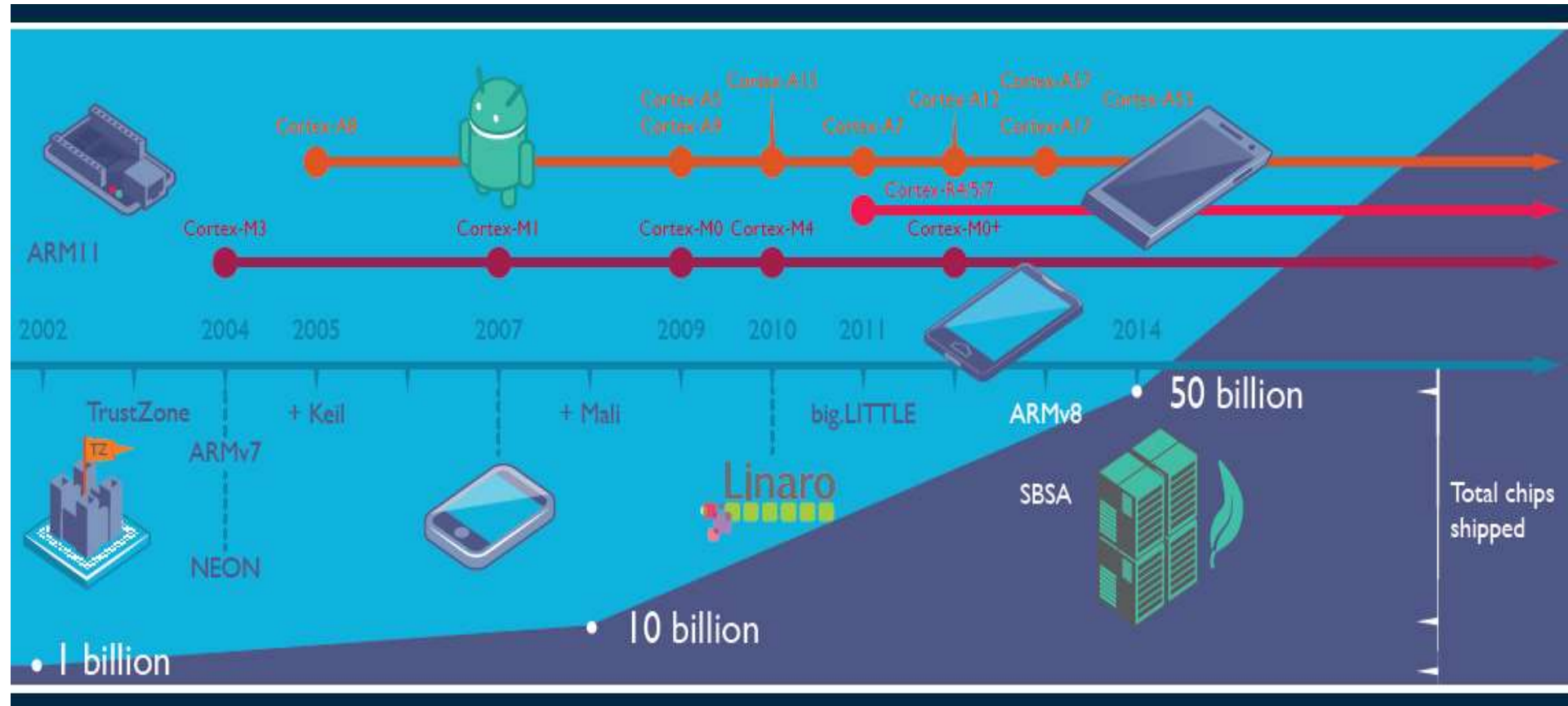
4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (25)

Use of ARMv7-based (32-bit) ARM Cortex-A models in mobiles [18]

	(SOC) System- On-a-Chip	Notable Product(s) Containing	ARM Cortex-A model	No. of Cores
Apple	A4	iPhone 4, iPod Touch (4th Gen), iPad (1st Gen), AppleTV (2nd Gen)	Cortex-A8	1
	A5	iPhone 4S, iPad 2, AppleTV (3rd Gen)	Cortex-A9	2
	A5X	iPad (3rd Gen, Retina Display)	Cortex-A9	2
Samsung	Exynos 3 Single	Samsung Galaxy S, Samsung Galaxy Nexus S,	Cortex-A8	1
	Exynos 4 Dual	Samsung Galaxy SII, Samsung Galaxy Note (International)	Cortex-A9	2
	Exynos 4 Quad	Samsung Galaxy SIII	Cortex-A9	4
	Exynos 5 Dual	Chrombook	Cortex-A15	2
Nvidia	Tegra	Microsoft Zune HD	(ARM11)	1
	Tegra 2	ASUS Eee Pad Transformer, Samsung Galaxy Tab 10.1, Motorola Xoom, Dell Streak 7 & Pro, Sony Tablet S	Cortex-A9	2
	Tegra 3	ASUS Transformer Pad 300, ASUS Nexus 7, Acer Iconia Tab A510 & A700, HTC One X	Cortex-A9	4
Qualcomm	Snapdragon S1	Large number of devices	(ARM11)/A5	1
Texas Instruments	OMAP 3	Barnes and Noble Nook Color	Cortex-A8	1
	OMAP 4	Amazon Kindle Fire, Samsung Galaxy Tab 2, Blackberry Playbook, Samsung Galaxy Nexus, Barnes and Noble Nook Tablet	Cortex-A9	2
	OMAP 5	N/A	Cortex-A15	2

4. Evolution of the Cortex-A series ARMv7/v8.0 ISA-based models (26)

Total number of ARM chips shipped [19]



SBSA: Server Base System Architecture, it is a standardized platform for servers built on 64-bit ARM processors

Remarks on the evolution of the Cortex-A series (taken from [99]) -1

There is a brief but essential description of ARM's Cortex series in [99]. Due to its relevance in the following we will cite parts of it.

"The **Cortex A9** was an incredibly important design for ARM as, in my view, it provided the corner-stone for SoC and device vendors to create some of the designs that powered some of the most successful devices that brought with them a turning-point in smartphone performance and experience. Apple's A5, Samsung's Exynos 4210/4412, and TI OMAP4430/4460 were all SoCs which made the A9 a very successful CPU micro-architecture.

Following the Cortex A9 we saw the introduction of the **Cortex A15**. The core was a substantial jump in terms of performance as it provided the single largest IPC improvement in ARM's Cortex A-profile of application processors. While the A15 represented a large performance boost, it came at significant cost in terms of power efficiency and overall power usage. It took some time for the Cortex A15 to establish itself in the mobile space as the first designs such as the Exynos 5250 and 5410 failed to impress due to bad power efficiency due to various issues.

It's at this point where ARM introduced **big.LITTLE** with the argument that one can have the best of both worlds, a high-power performant core together with a low-power high-efficiency core. It was not until late 2014 and 2015 did we finally see some acceptable implementations of A15 big.LITTLE solutions such as the Kirin 920 or Exynos 5422."

Remarks on the evolution of the Cortex-A series (taken from [99]) -2

"The **Cortex A57** succeeded the Cortex A15 and was ARM's first "big" core to employ ARMv8 64-bit ISA. Accompanied by the high-efficiency Cortex A53 cores this represented an important shift not unlike the x86-64 introduction in the desktop PC space well over a decade before. The cores came at a moment where the industry was still at shock of Apple's introduction of the A7 SoC and Cyclone CPU micro-architecture, beating ARM in terms delivering the first 64-bit ARMv8 silicon. Suddenly everybody in the industry was playing catch-up in trying to bring their own 64-bit products as it was seen as an absolutely required feature-check to remain competitive.

This pressured shift to 64-bit was in my view a crippling blow to many 2015's SoCs as it forced vendors into employing **sub-optimal Cortex A57 and A53 designs**. HiSilicon and MediaTek saw an actual regression in performance as flagship SoCs such as the Kirin 930 and Helio X10 had to make due with only A53 cores for performance as they decided against employing A57 cores due to power consumption concerns. The Kirin 930 or the X10 were in effect slower chipsets than their predecessors. Only Samsung was fairly successful in releasing reasonable designs such as the Exynos 5433 and Exynos 7420 – yet these had respectively regressed or barely improved in terms of power efficiency when compared to mature Cortex A15 implementations such as the Exynos 5430. Then of course we had sort of a lost generation of devices due to Qualcomm's unsuccessful Snapdragon 810 and 808 SoCs, a topic we'll eventually revisit in our deep dive of the Snapdragon 820 and Exynos 8890."

Remarks on the evolution of the Cortex-A series (taken from [99]) -3

"Some readers will notice I left out the **Cortex A12 and A17** – and I did that on purpose in trying to get to my point. The Cortex A12 was unveiled in July 2013 and presented as a successor to the Cortex A9. The core had a relatively short lifetime as it was quickly replaced within 6 months with the Cortex A17 in February 2014 which improved performance and also made the core big.LITTLE compatible with the Cortex A7. The Cortex A17 saw limited adoption in the mobile space. In fact, among the few SoCs such as Rockchip's RK3288 and some little known chips such as HiSilicon's Hi3536 multimedia SoC, it was only MediaTek's MT6595 that saw moderate success in design wins such as Meizu's MX4.

5. Cortex-A models based on the ARMv8.0 ISA

- 5.1 Overview of the Cortex-A models based on the ARMv8.0 ISA
- 5.2 High performance Cortex-A models based on the ARMv8.0 ISA
- 5.3 Low-power Cortex- A models based on the ARMv8.0 ISA

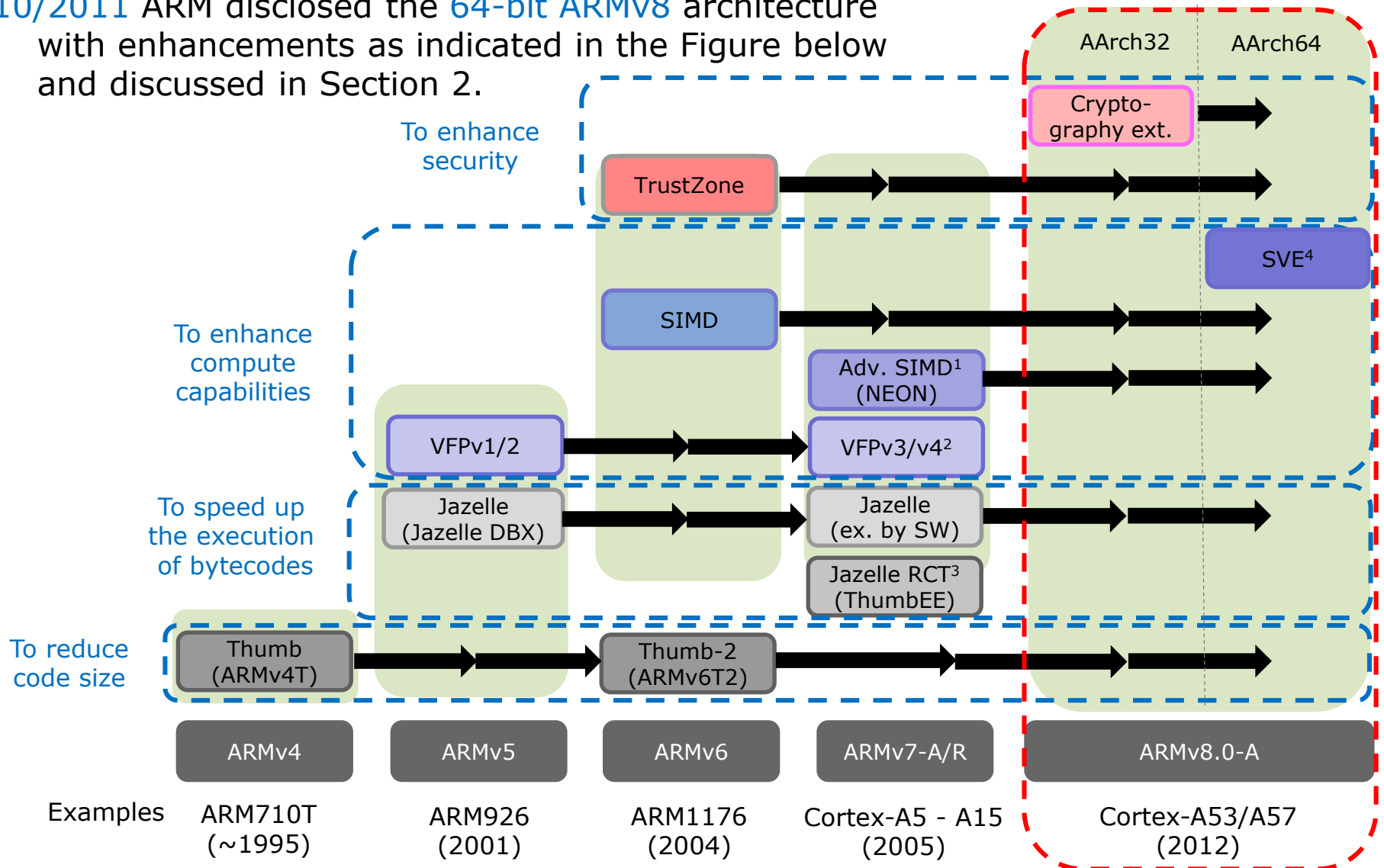
(This Section will not be discussed)

5.1 Overview of the Cortex-A models based on the ARMv8.0 ISA

5.1 Overview of the Cortex-a models based on the ARMv8.0 ISA (1)

5.1 Overview of Cortex-A models based on the ARMv8.0 ISA -1 (After [64])

10/2011 ARM disclosed the 64-bit ARMv8 architecture with enhancements as indicated in the Figure below and discussed in Section 2.

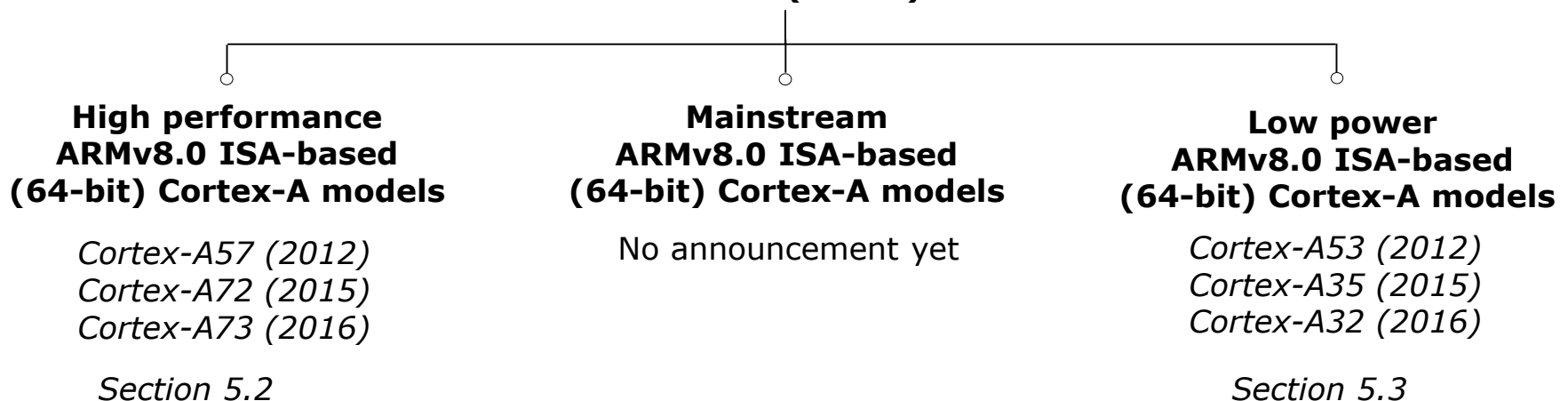


Remarks: See on the slide 2.1 Overview (5).

5.1 Overview of the Cortex-a models based on the ARMv8.0 ISA -2

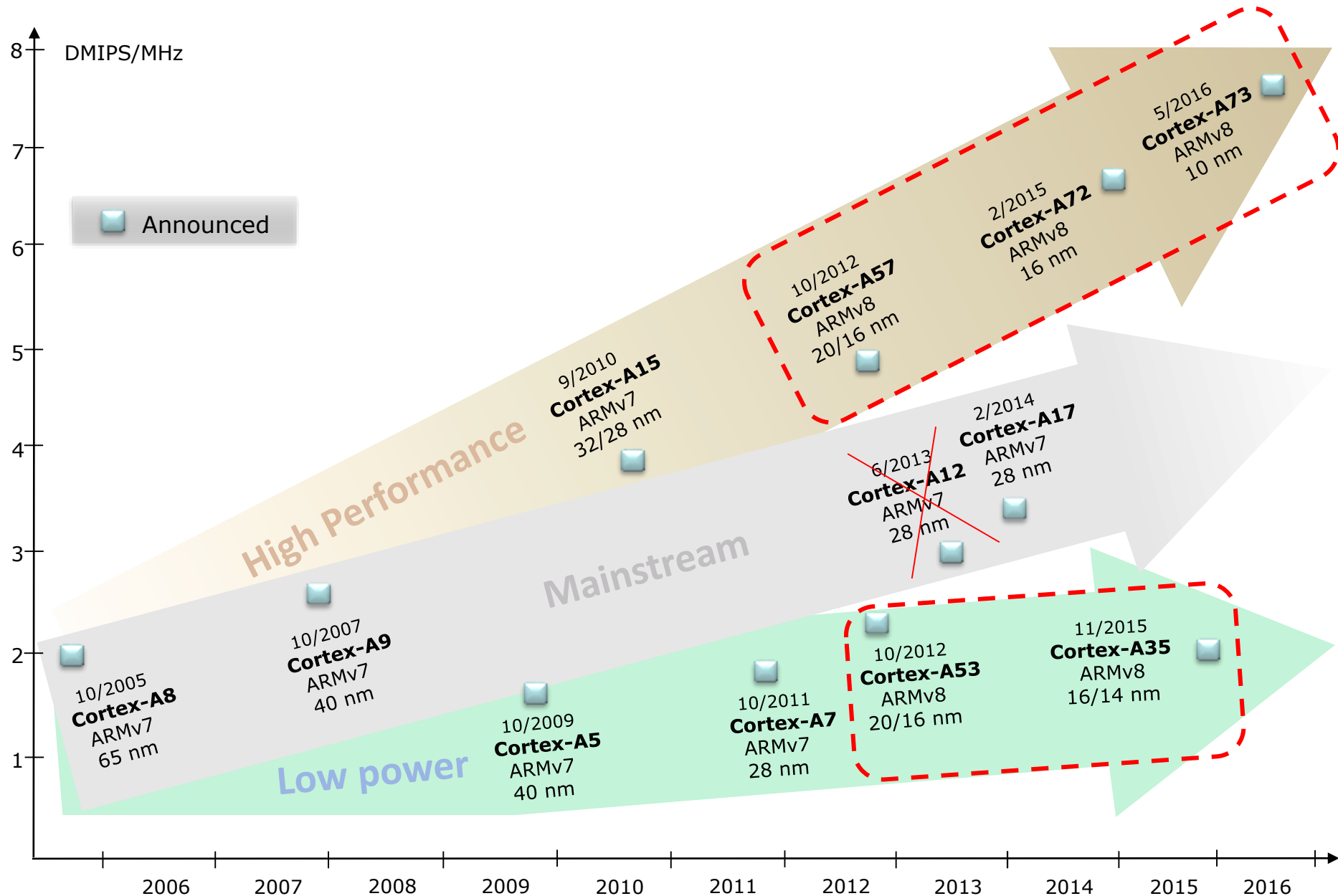
- **10/2012**: ARM announced the 64-bit **high performance Cortex-A-57** and the **low power Cortex-A-53** processors, as first implementations of the ARMv8 architecture, with immediate availability.
- **02/2015** ARM extended their 64-bit Cortex-A series by the **high performance Cortex-A72** model, and in **11/2015** by the **low power Cortex-A35**, as shown below.
- **05/2016** ARM extended the 64-bit Cortex-A series by the **high performance Cortex-A73** model.

ARMv8.0 ISA-based (64-bit) Cortex A models



5.1 Overview of the Cortex-a models based on the ARMv8.0 ISA (3)

5.1 Overview of the Cortex-A models based on the ARMv8.0 ISA -3



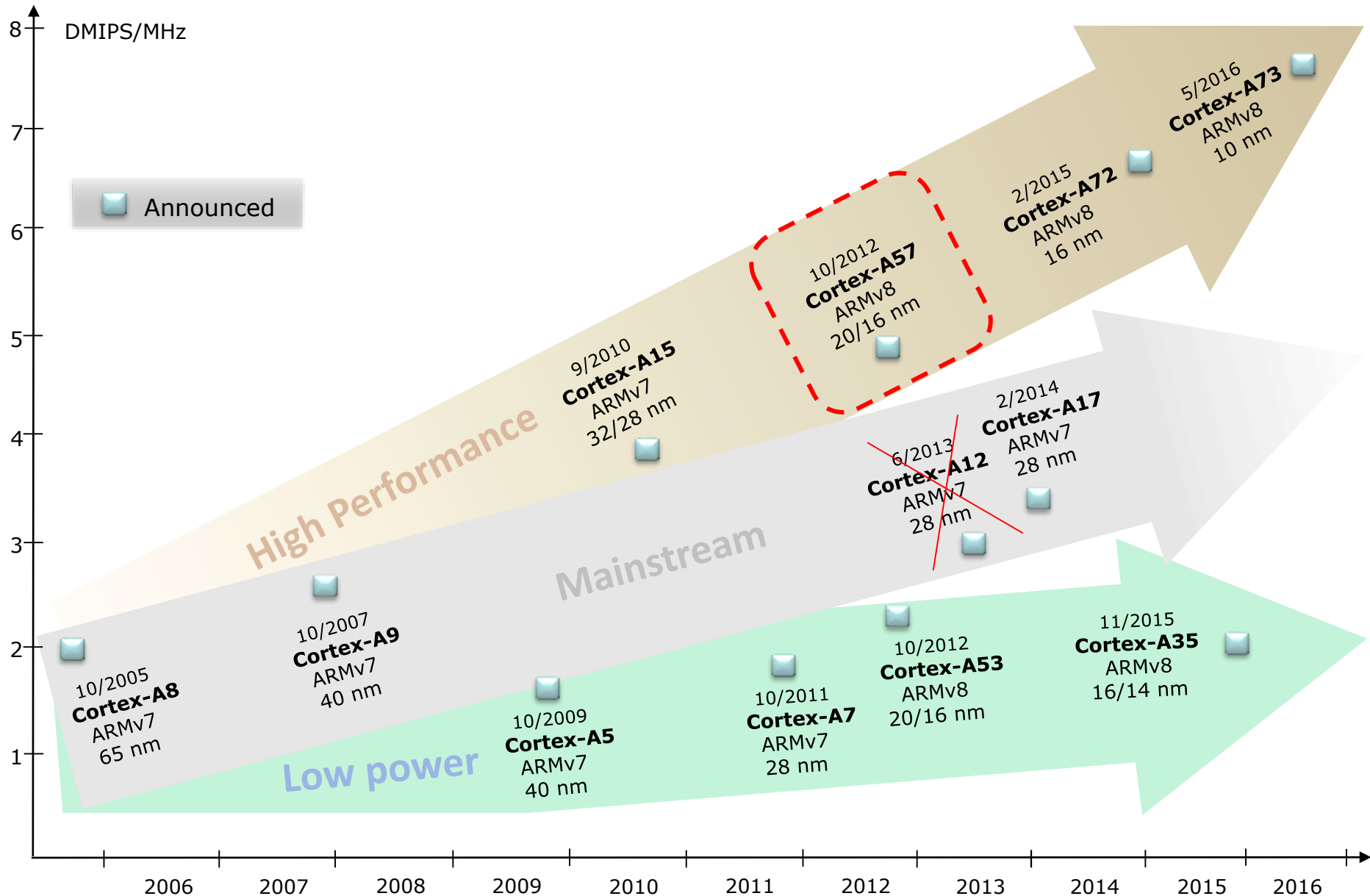
5.2 High performance Cortex-A models based on the ARMv8.0 ISA

- 5.2.1 The high performance Cortex-A57
- 5.2.2 The high performance Cortex-A72
- 5.2.3 The high performance Cortex-A73

5.2.1 The high performance Cortex-A57

5.2.1 The high performance Cortex-A57 (1)

5.2.1 The high performance Cortex-A57 -1 (based on [12])



The high performance Cortex-A57 -2

- **10/2012:** Announced along with the **low power Cortex-A53** processor, with immediate availability for licensing.
- **2014: First mobile devices** with Cortex-A57 models are emerged.
- They are 64-bit successors to the high performance 32-bit Cortex A15.
- The A57 and A53 models can be **used either as stand alone processors or as components of a big-LITTLE configuration**, similar to the 32-bit Cortex-A15/A7 combination.
- **Interoperability** with the **ARM Mali GPU family** is provided.
- Target process technology: **16 or 20 nm**.

Key features of the microarchitecture of the Cortex-A57

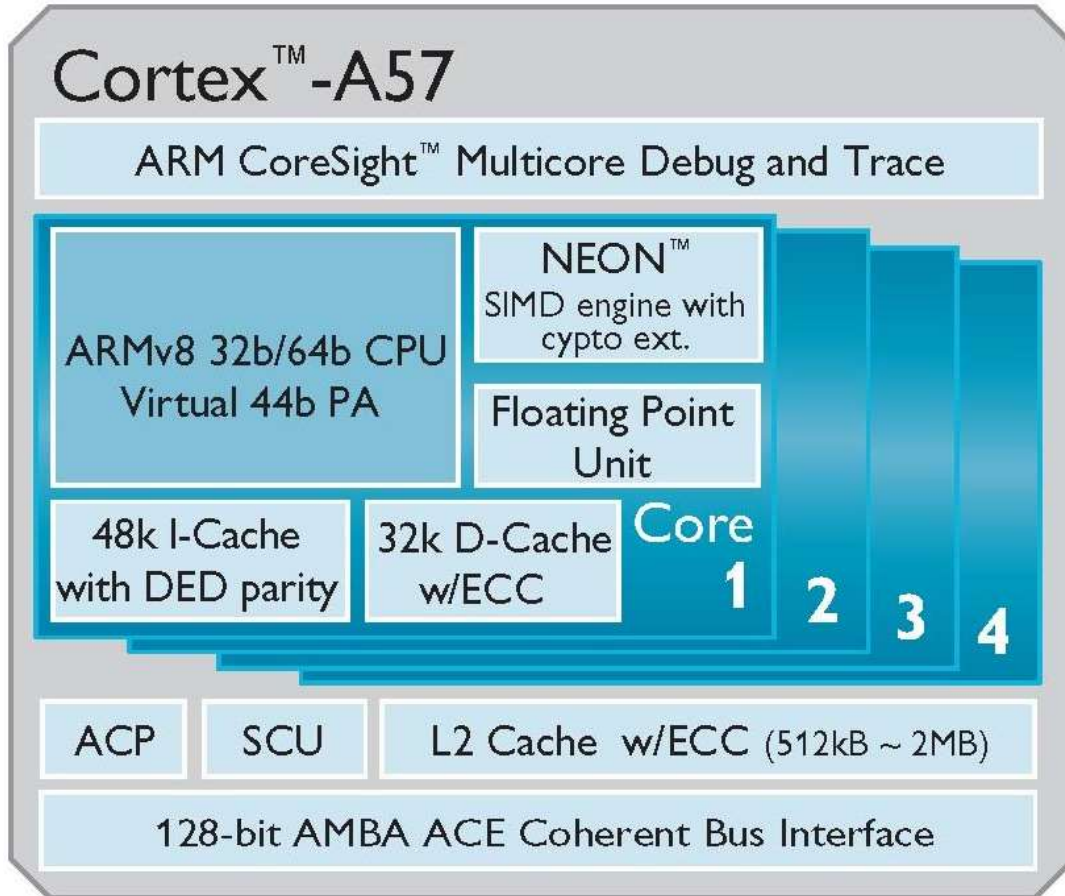
- Full compatible with the preceding ARMv7 32-bit ISA.
- Integrated L2 cache of 512 kB to 2 MB.
- Up to 4 Cortex-A57 CPUs that execute the ARMv8 64-bit ISA.
- 128-bit AMBA ACE coherent interface for connecting multiple (up to 4) Cortex-A57 processors.
- Processor wide cache coherence supported by a Snoop Control Unit (SCU).

Remarks

AMBA: Advanced Microcontroller Bus Architecture
(On-chip bus standard for SoC) designs

5.2.1 The high performance Cortex-A57 (4)

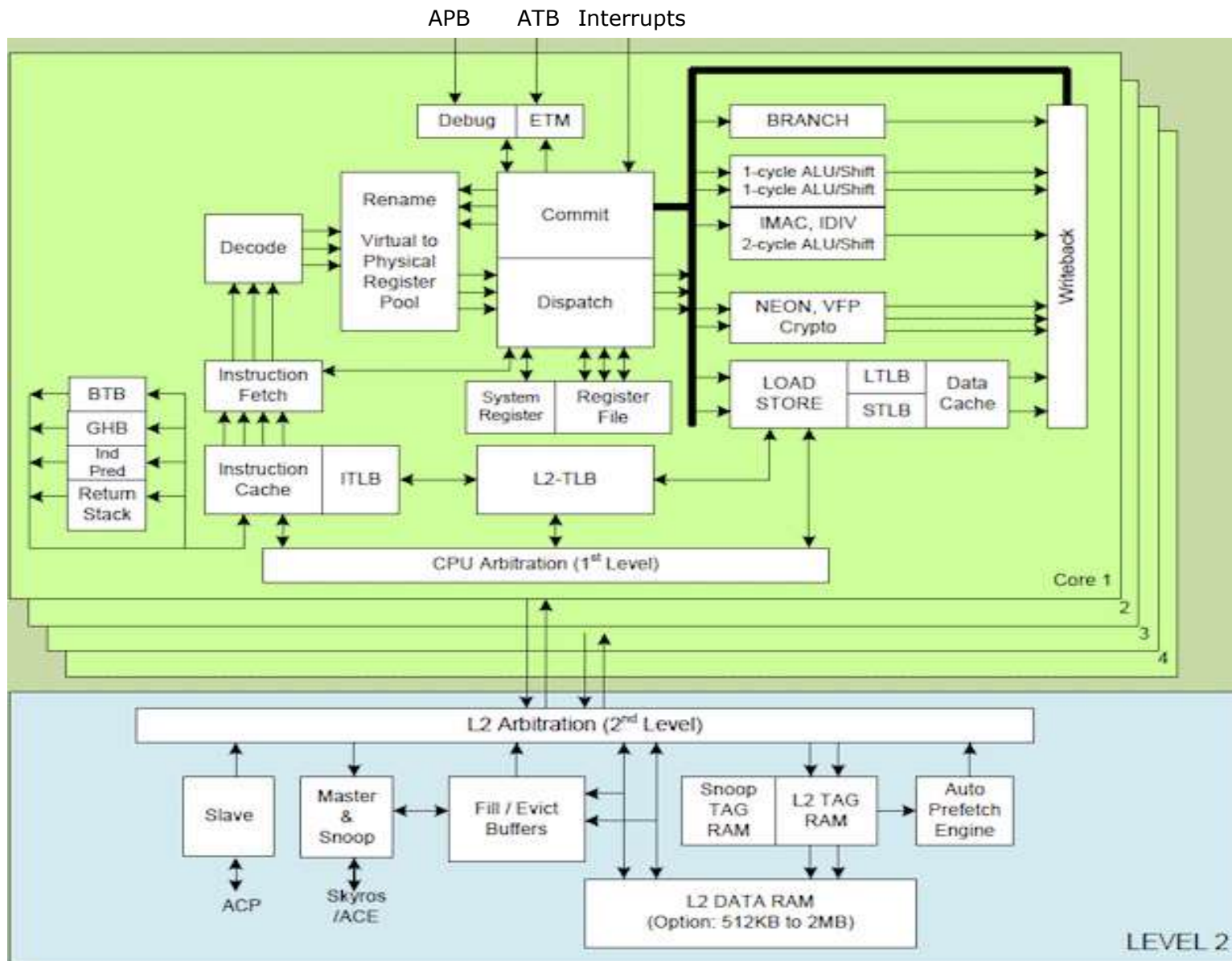
High level block diagram of the Cortex-A57 [53]



- PA: Physical Address
- DED: Double Error Detection
- ECC: Error Correcting Code
- ACP: Accelerator Coherence Port
(to connect non-cached coherent data sources)
- SCU: Snoop Control Unit
- AMBA: Advanced Microcontroller Bus Architecture
(On-chip bus standard for SoC designs)
- ACE: AXI Coherency Extensions
(Used in big.LITTLE systems for smartphones, tablets, etc.)

5.2.1 The high performance Cortex-A57 (5)

Main functional blocks of the high performance Cortex-A57 CPU [74]



Remarks: Non trivial abbreviations in the above Figure

ATB: AMBA Trace Bus

APB: Advanced Peripheral Bus

ACP: Accelerator Coherence Port

(to connect non-cached coherent data sources)

ACE: AXI Coherency Extensions

(Used in big.LITTLE systems for smartphones, tablets, etc.)

AXI: Advanced eXtensible Interface

(The most widespread AMBA interface).

Key features of the microarchitecture of the Cortex-A57 core -1

- Each core fetches **four instructions** per cycle from the Icache,
- decodes and renames **three microinstructions** per cycle,
- dispatches **three microinstructions** per cycle to the issue queues,
- whereas the issue queues **issue up to eight microinstructions** per cycle to the eight available execution units.

The available execution units are

- a branch unit
- dual single cycle integer units
- a multi-cycle integer MAC/DIV/CRC unit
- dual Advanced SIMD (NEON)/FP. Crypto units and
- dual load/store units.

Key features of the microarchitecture of the Cortex-A57 core -2

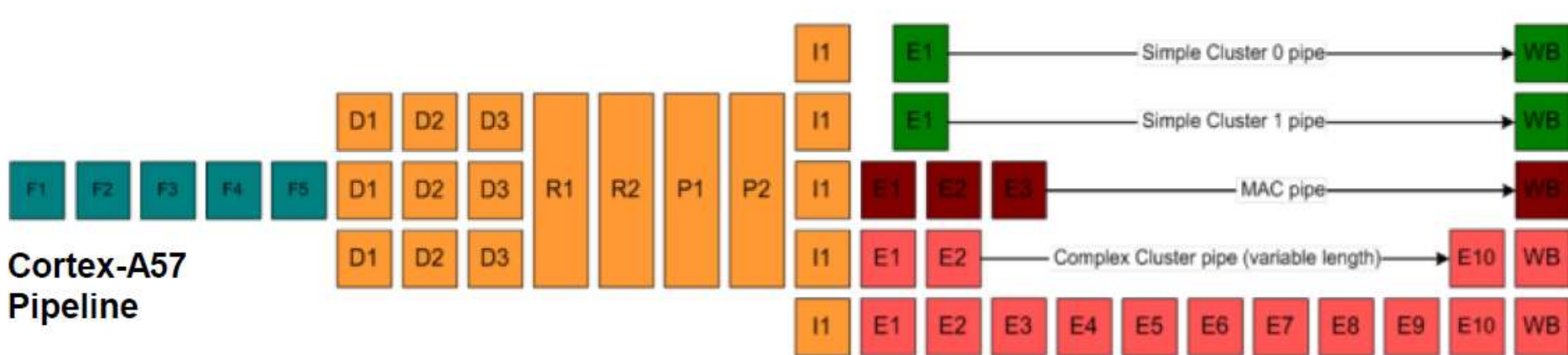
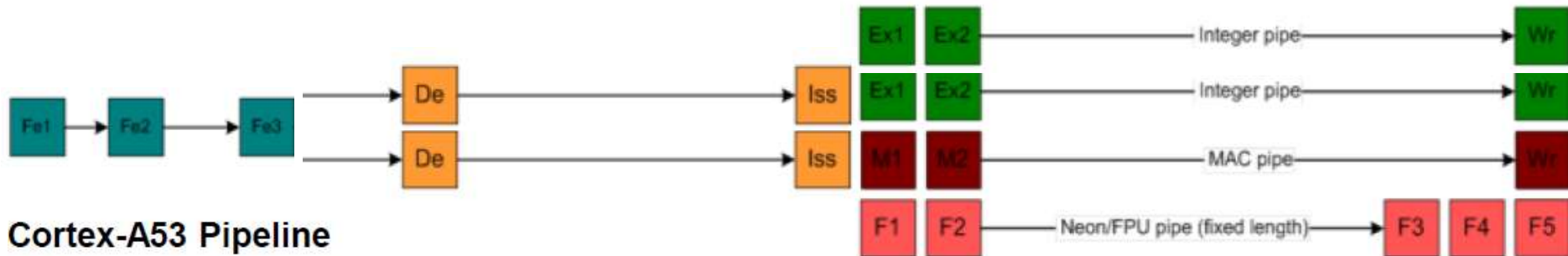
- Core efficiency: 4.8 DMIPS/MHz
- Expected core frequency up to 2.5 GHz in a 16 nm process implementation.

Pipeline of the Cortex-A57 core -1

- The Cortex-A57 core has a 3-wide in-order front end and an 8 issue-wide out-of-order back end pipeline with 15 stages for integer processing and additional pipeline stages for NEON and FP processing, as indicated in the next two Figures.

5.2.1 The high performance Cortex-A57 (10)

Contrasting the Cortex-A53 and Cortex-A57 arithmetic pipelines [Based on 54]

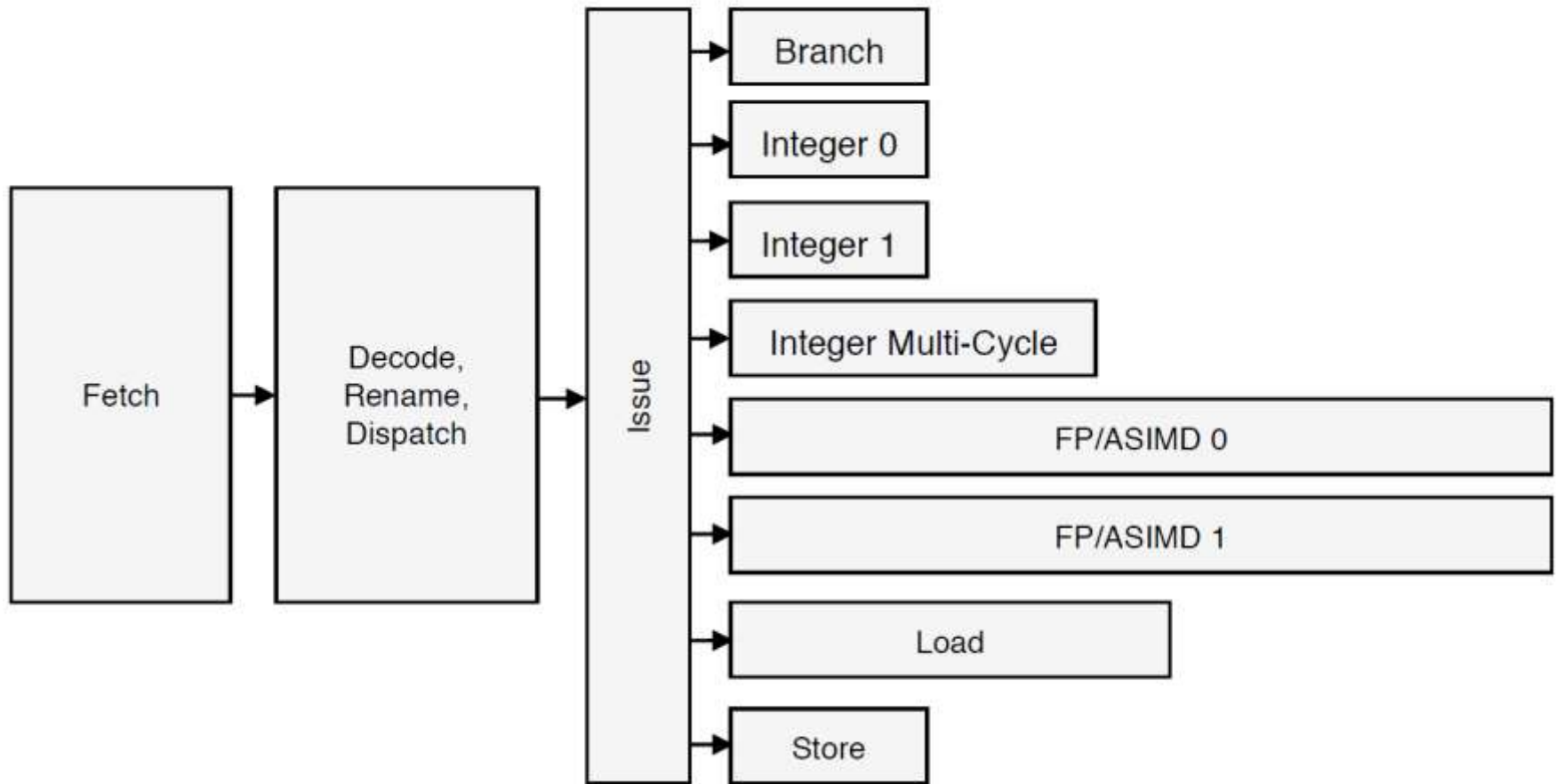


D: Decode
R: Rename
P: Dispatch
I: Issue
E: Execute
WB: Write Back

Notes: a) Branch and Load/Store pipelines not shown
(1x Load/Store pipeline for the Cortex A-53 and
2x Load/Store and 1x Branch pipeline for the Cortex-A-57)
b) For the A57 only one Complex Cluster pipeline shown
from the two available (these are actually ASIMD pipes)

5.2.1 The high performance Cortex-A57 (10b)

Pipeline structure of a Cortex-A57 core, as shown in the Cortex-A57 Software Optimization Guide [122]



Operation of ASIMD (advanced SIMD) pipelines [123]

- **ASIMD multiply-accumulate pipelines** support **late-forwarding** of accumulate operands from similar μ ops in order to speed up the execution of a typical sequence of FP multiply-accumulate instructions, as detailed next.
- Actually, in order to issue an FMAD operation not all three input operands need to be ready.

The FMAD operation can already be started when the multiply operands are available, the accumulate operand can be added later after the multiply operation has already been calculated.

- This kind of execution of FMA operations is called **late-forwarding**.
- ASIMD is **beneficial**, when a **chain of FMAD operations** needs to be executed such that **one FMA is using the result of the previous FMA** operation as its accumulate operand.

FMAD operations last 9 cycles, thus **without ASIMD 9 cycles need to be waited before the next FMAD can be issued on the pipe.**

But with late forwarding **the next FMAD can be issued already in the 4th cycle**, as by the time the FMAD needs the accumulate operand the previous FMAD would already be finished.

- All in all FSIMD allows the interleaved execution of a chain of FMAD operations significantly faster than without this capability.

5.2.1 The high performance Cortex-A57 (10d)

Execute latencies of FMADD instructions with and without ASIMD capability [122]

Instruction Group	AArch64 Instructions	Exec Latency	Execution Throughput	Utilized Pipelines	Notes
FP multiply, no FZ	FMUL, FNMUL	6	2	F0/F1	2
FP multiply accumulate, FZ	FMADD, FMSUB, FNMADD, FNMSUB	9 (4)	2	F0/F1	3
FP multiply accumulate, no FZ	FMADD, FMSUB, FNMADD, FNMSUB	10 (4)	2	F0/F1	3

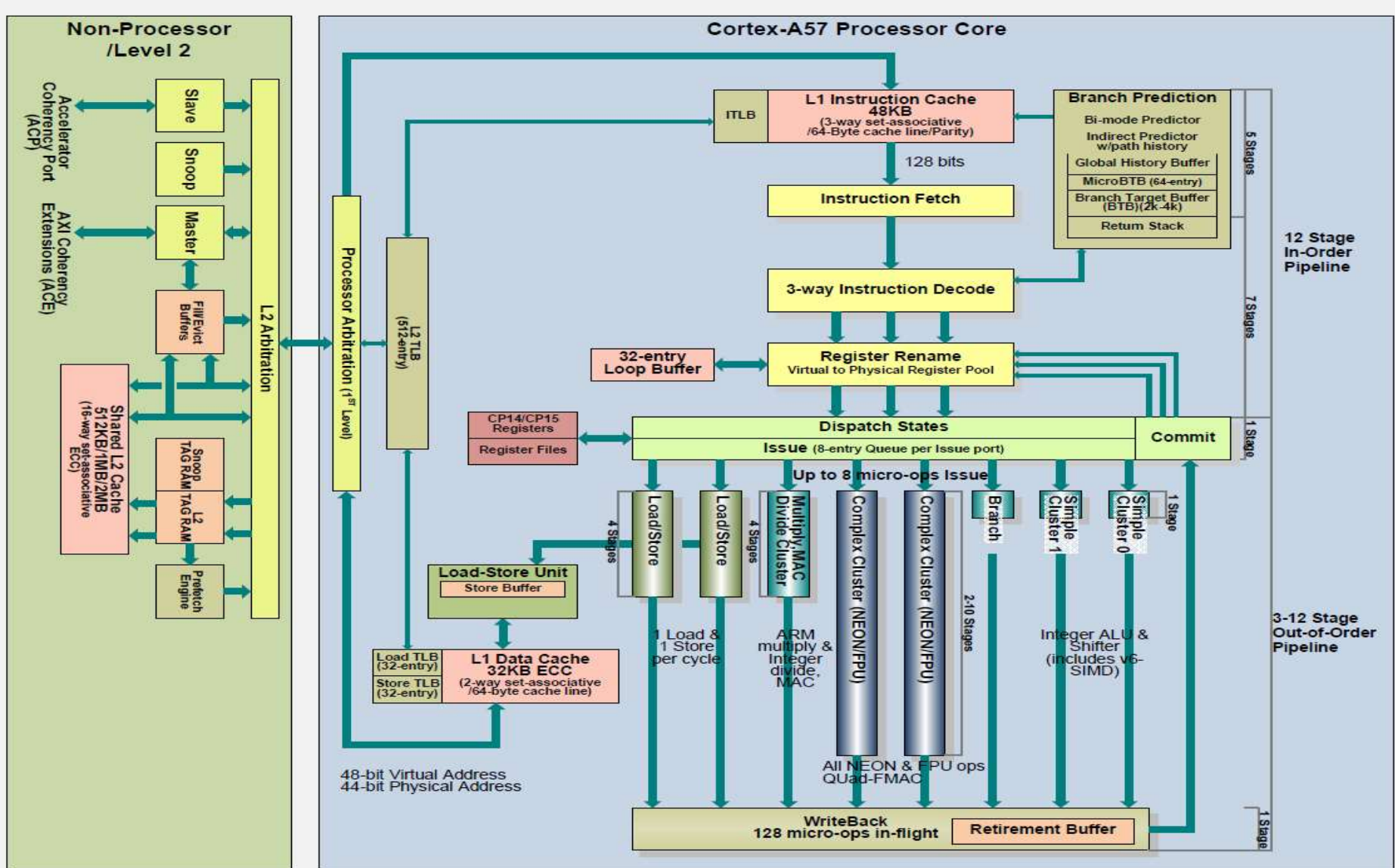
NOTE 3 – FP multiply-accumulate pipelines support late-forwarding of accumulate operands from similar μ ops, allowing a typical sequence of multiply-accumulate μ ops to issue one every N cycles (accumulate latency N shown in parentheses).

Implementation of ASIMD pipelines in subsequent ARM processors

- The **ASIMD** capability of the A57 processors was officially **disclosed** in the Software Optimization Guide of this processor only **in 01/2015, many years after its introduction** (10/2012).
- According to the available documentation, the **high-performance models implementing the ARM8 ISA** (Cortex-A57/72/73 etc.) **provide this feature whereas the low-performance models, like the Cortex-A53 etc. not**).

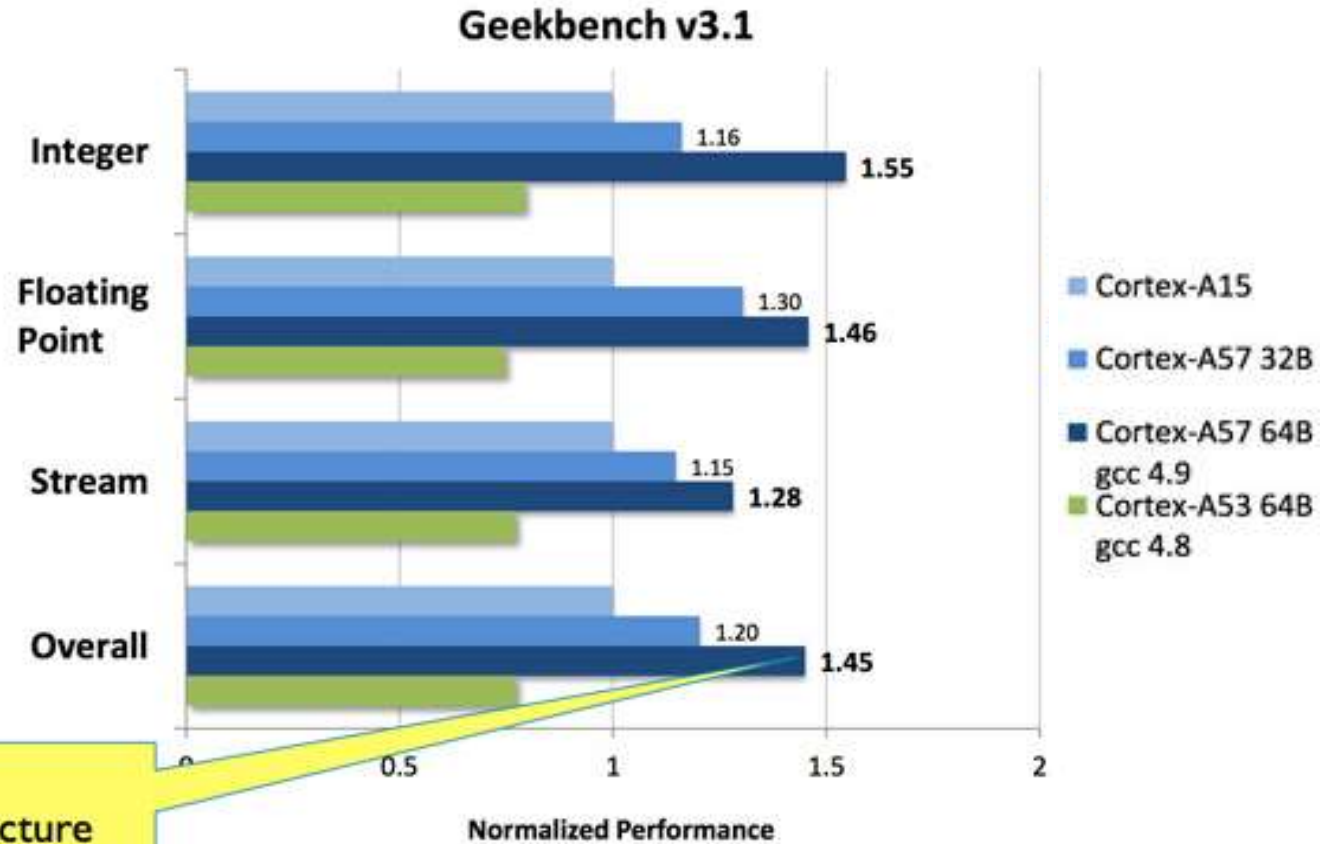
5.2.1 The high performance Cortex-A57 (11)

Actual pipeline structure of the Cortex-A57 core (alternative view) [71]



5.2.1 The high performance Cortex-A57 (12)

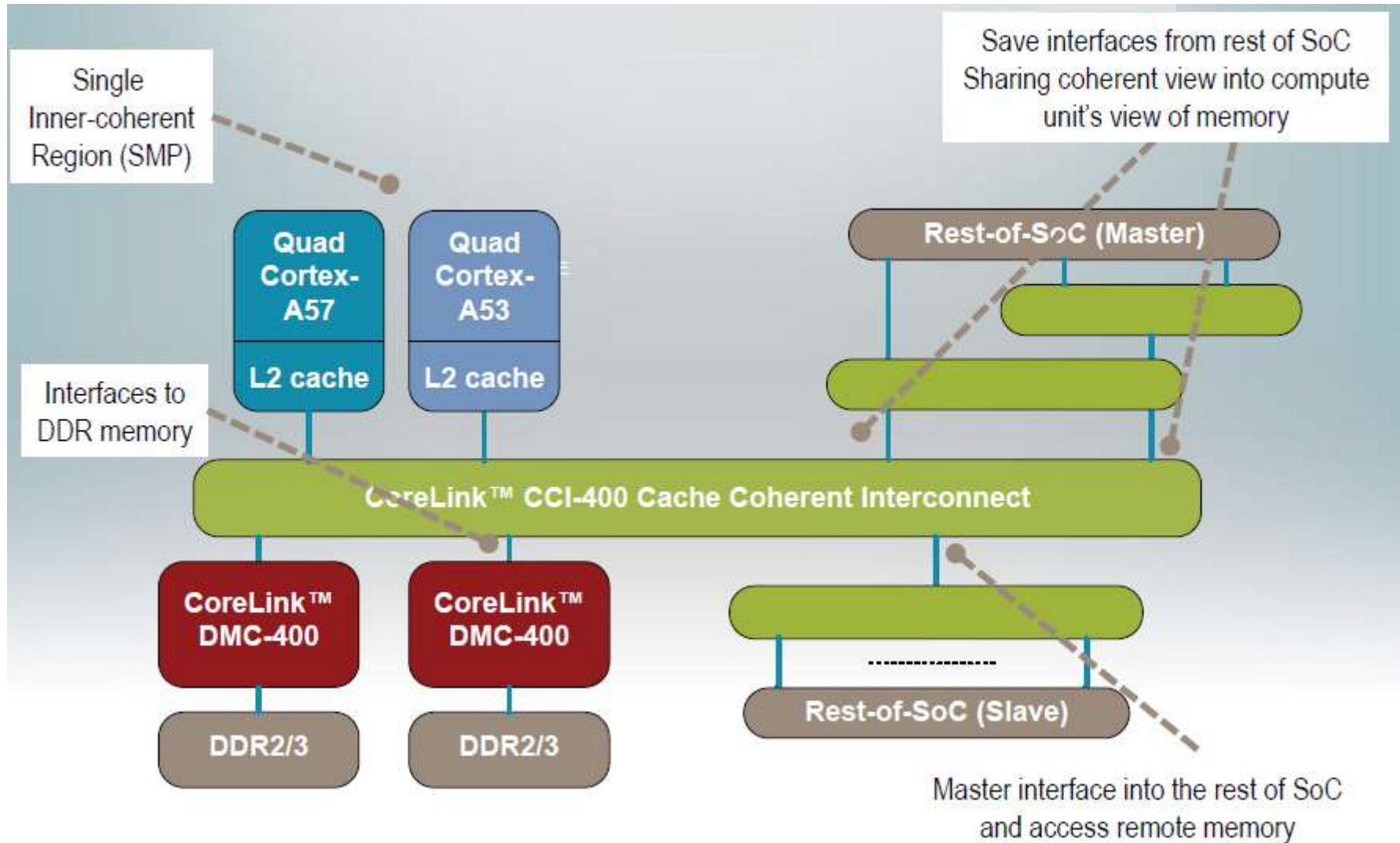
Cortex-A57/A53 performance - compared to the Cortex-A15 [55]



45% increase through incremental microarchitecture improvements

5.2.1 The high performance Cortex-A57 (13)

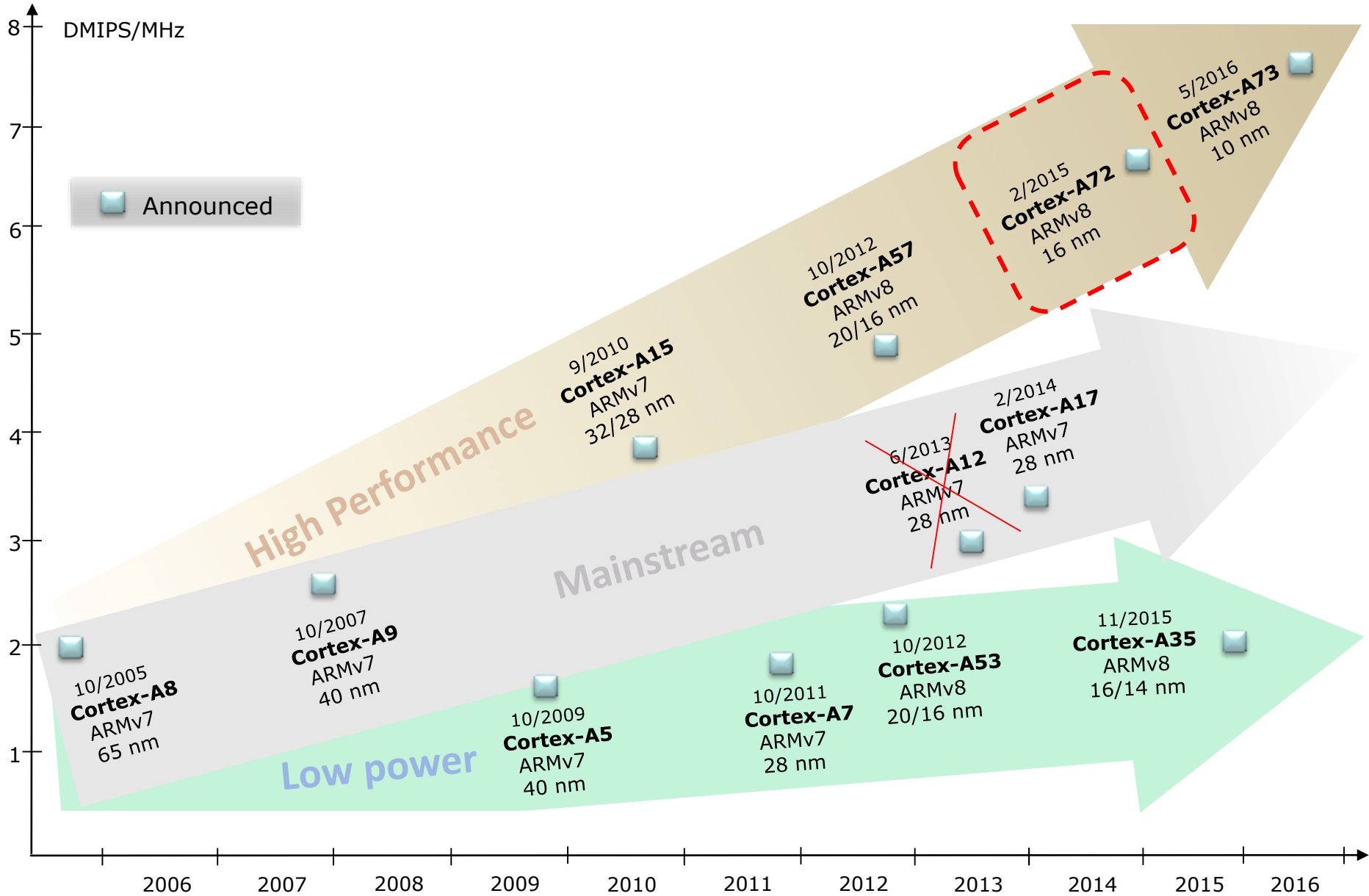
Example: A Cortex-A57/Cortex-A53 big.LITTLE system [52]



5.2.2 The high performance Cortex-A72

5.2.2 The high performance Cortex-A72 (1)

5.2.2 The high performance Cortex-A72 (1) (based on [12])



The high performance Cortex-A72 -2 [91]

- **02/2015: Announced** along with the CoreLink CCI-500 cache Cache Coherent Interconnect and the Mali T880 GPU with immediate availability for licensing.
- **Q1/2016: First premium level mobile devices** with Cortex-A72.
- The Cortex-A72 is **based on the** high performance 64-bit **Cortex-A57**. Nevertheless, every logical block of the Cortex-A57 was **optimized for power efficiency**.

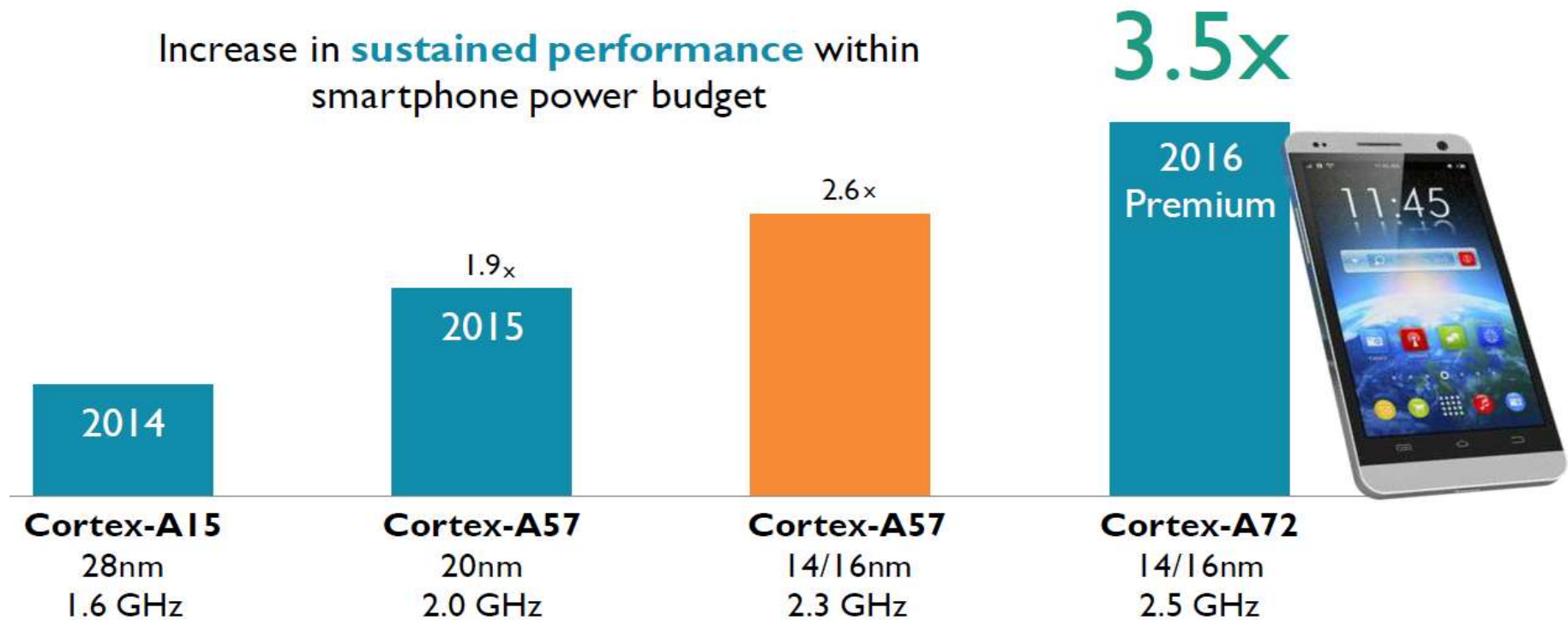
It was designed in ARM's Austin, Texas design center.

- The Cortex-A72 **can be used either as stand alone processors or as part of a big-LITTLE configuration along with the Cortex-A53 processors.**
Interoperability with the ARM Mali GPU family.
- Target process technology: **16 nm FinFET technology.**

5.2.2 The high performance Cortex-A72 (3)

Sustained relative performance of high performance Cortex-a models within a given power budget [91]

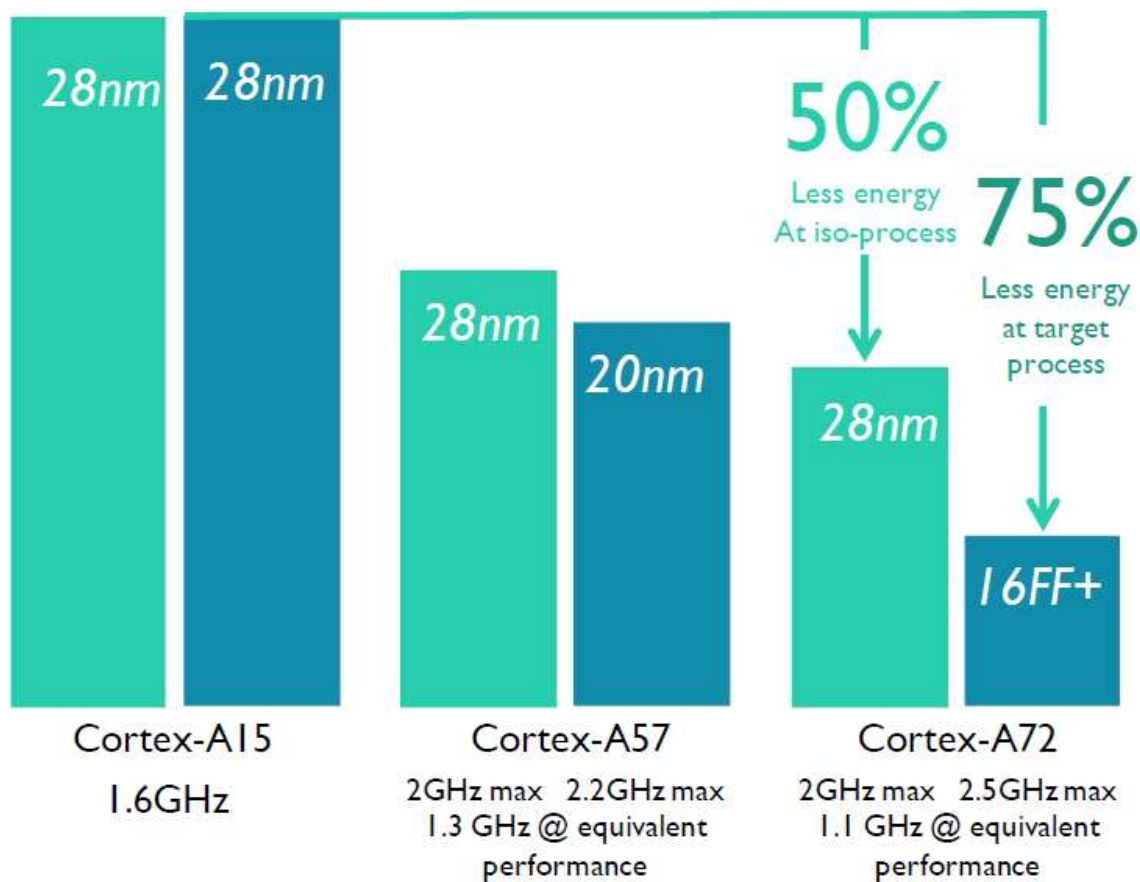
Increase in **sustained performance** within smartphone power budget



5.2.2 The high performance Cortex-A72 (4)

Energy consumption of high performance Cortex-A models for the same workload [91]

Energy consumed for same workloads



Combined with Cortex-A53:

40-60%

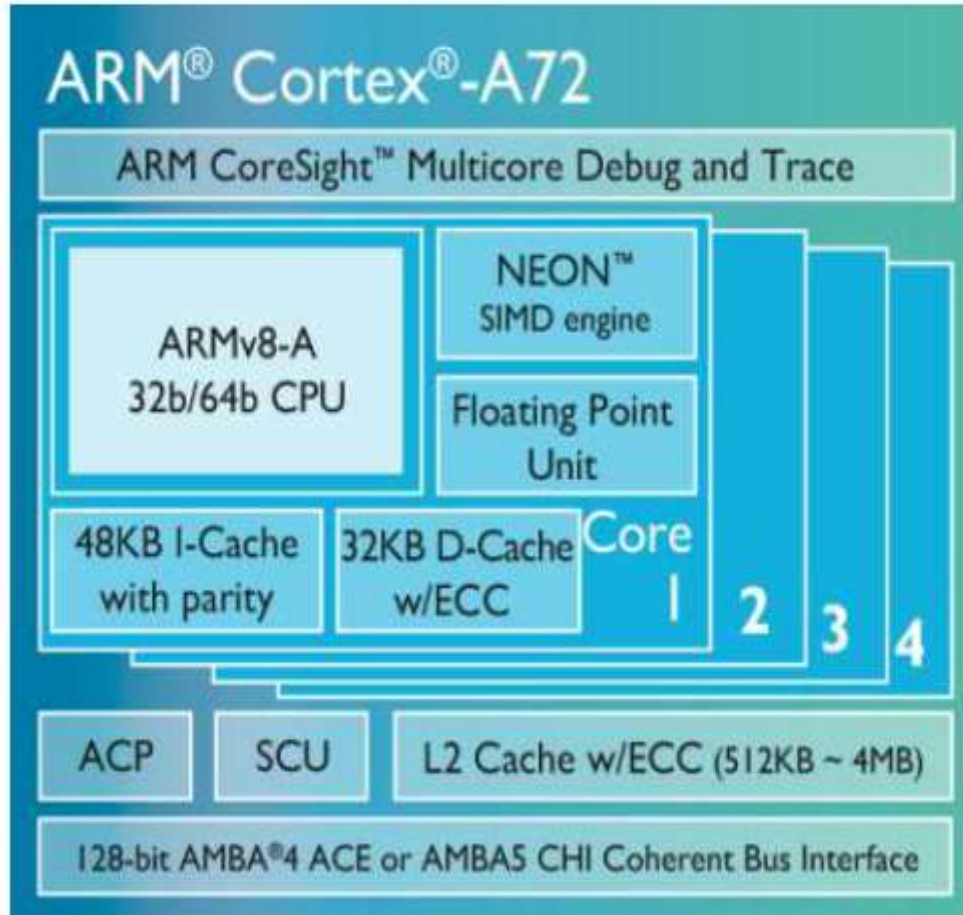
further reductions on average across multiple workloads



ARM[®] big.LITTLE™
Processor Technology

5.2.2 The high performance Cortex-A72 (5)

High level block diagram of the Cortex-A72 [91]



ECC: **E**rror **C**orrecting **C**ode

ACP: **A**ccelerator **C**oherence **P**ort
(to connect non-cached coherent data sources)

SCU: **S**noop **C**ontrol **U**nit
(Provides coherence within the the processor)

AMBA: **A**dvanced **M**icrocontroller **B**us **A**rchitecture
(On-chip bus standard for SoC designs)

ACE: **A**XI **C**oherency **E**xtensions
AMBA4 cache coherent interface
(Used in big.LITTLE systems for smartphones, tablets, etc.)

CHI: **C**oherent **H**ub Interface
(AMBA5 Cache coherent interface used in servers)

5.2.2 The high performance Cortex-A72 (6)

Key features of the microarchitecture of the Cortex-A72 [91]

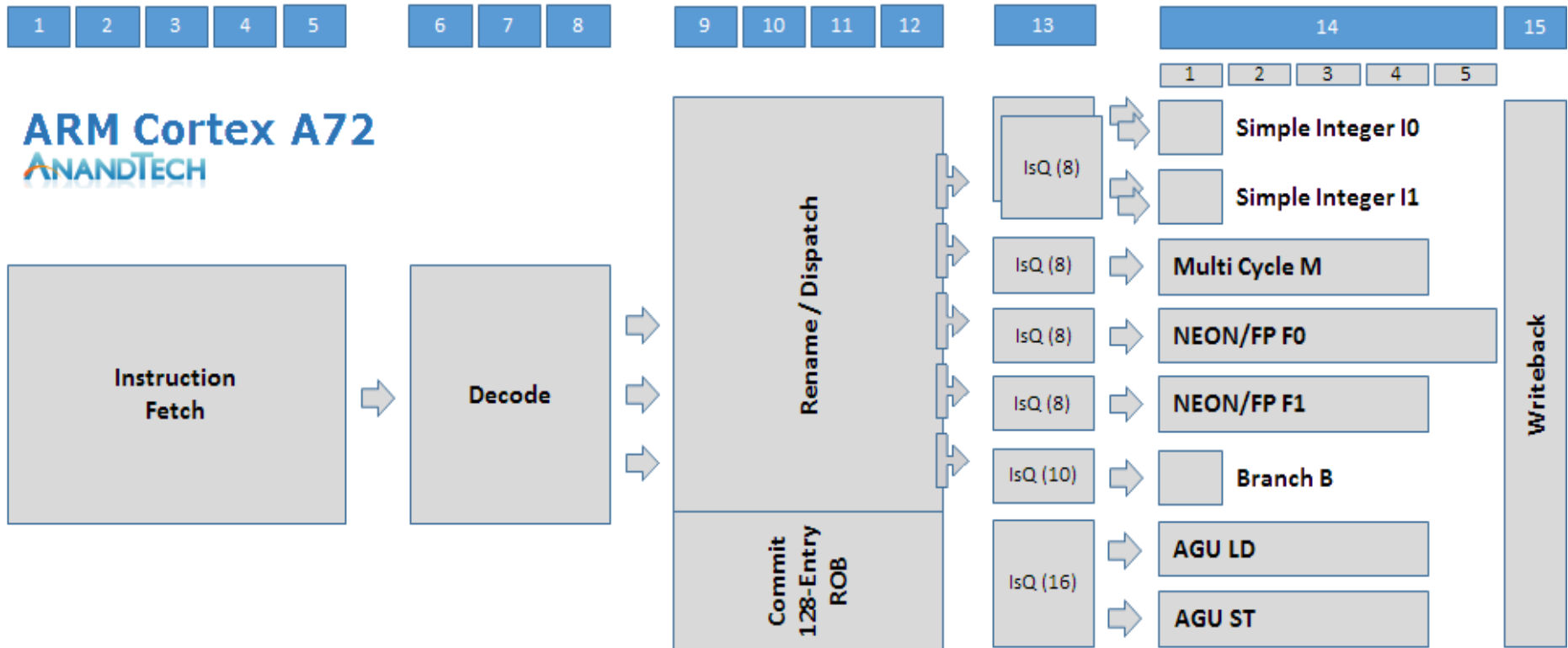
- It has a three-wide out-of-order front-end and a five-wide out-of-order back-end, as seen in the next Figure.
- Integrated L2 cache of 512 kB to 4 MB.
- Up to 4 Cortex-A73 CPUs executing the ARMv8 64-bit ISA.
- 128-bit coherent AMBA4 ACE or AMBA5 CHI interface for connecting multiple (up to 4) Cortex-A57 processors and further system components.
- Processor wide cache coherence supported by a Snoop Control Unit (SCU).
- Significant improvements in power efficiency through optimizing every block of the Cortex-A57.

Remarks

AMBA: Advanced Microcontroller Bus Architecture
(On-chip bus standard for SoC) designs

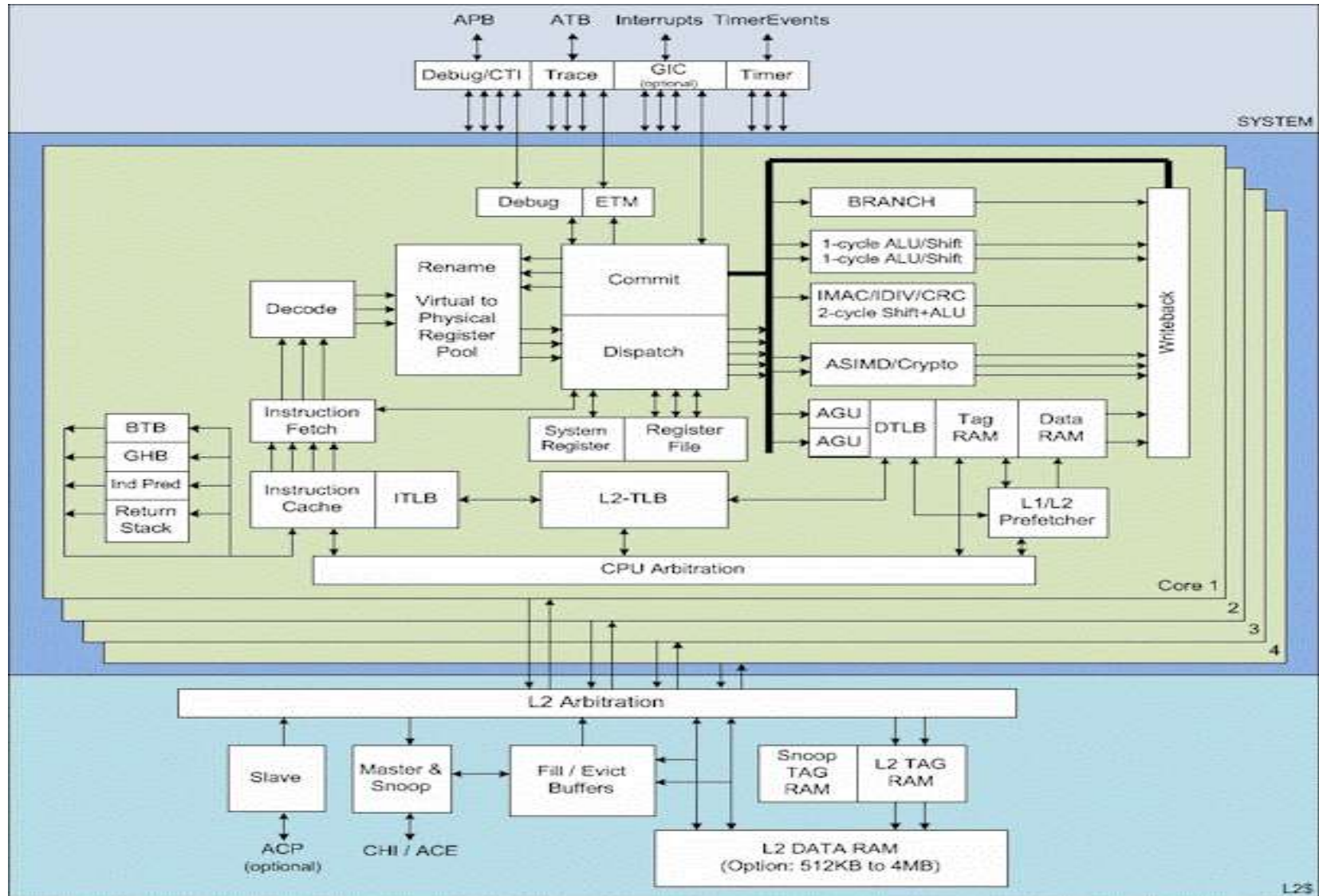
5.2.2 The high performance Cortex-A72 (7)

Pipeline structure of the high-end Cortex-A72 [92]



5.2.2 The high performance Cortex-A72 (8)

More detailed block diagram of the high performance Cortex-A72 [95]



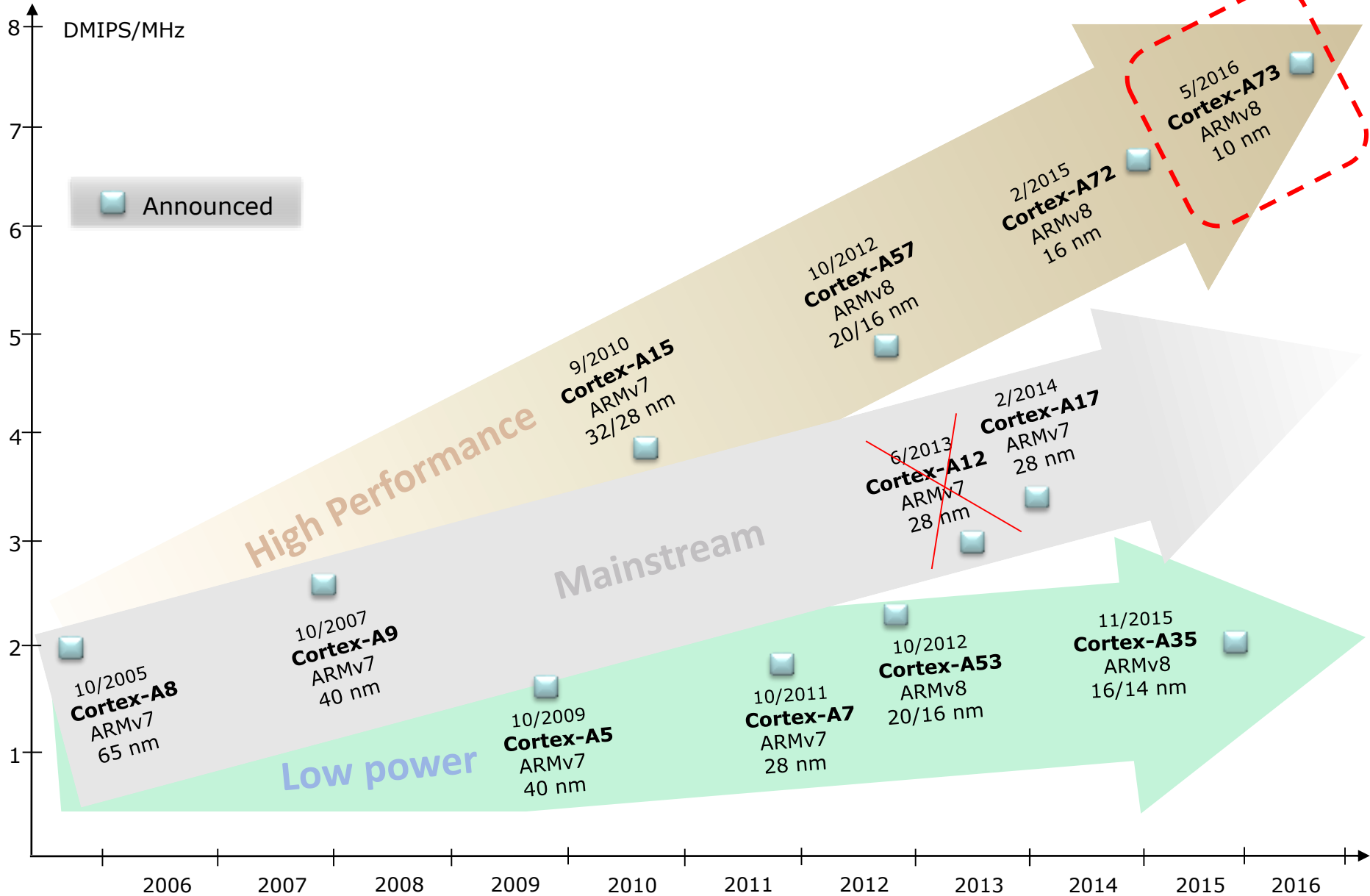
Remark

A detailed description of the microarchitecture of the Cortex-A72 can be found in [91].

5.2.3 The high performance Cortex-A73

5.2.3 The high performance Cortex-A73 (1)

5.2.3 The high performance Cortex-A73 (1) (based on [12])

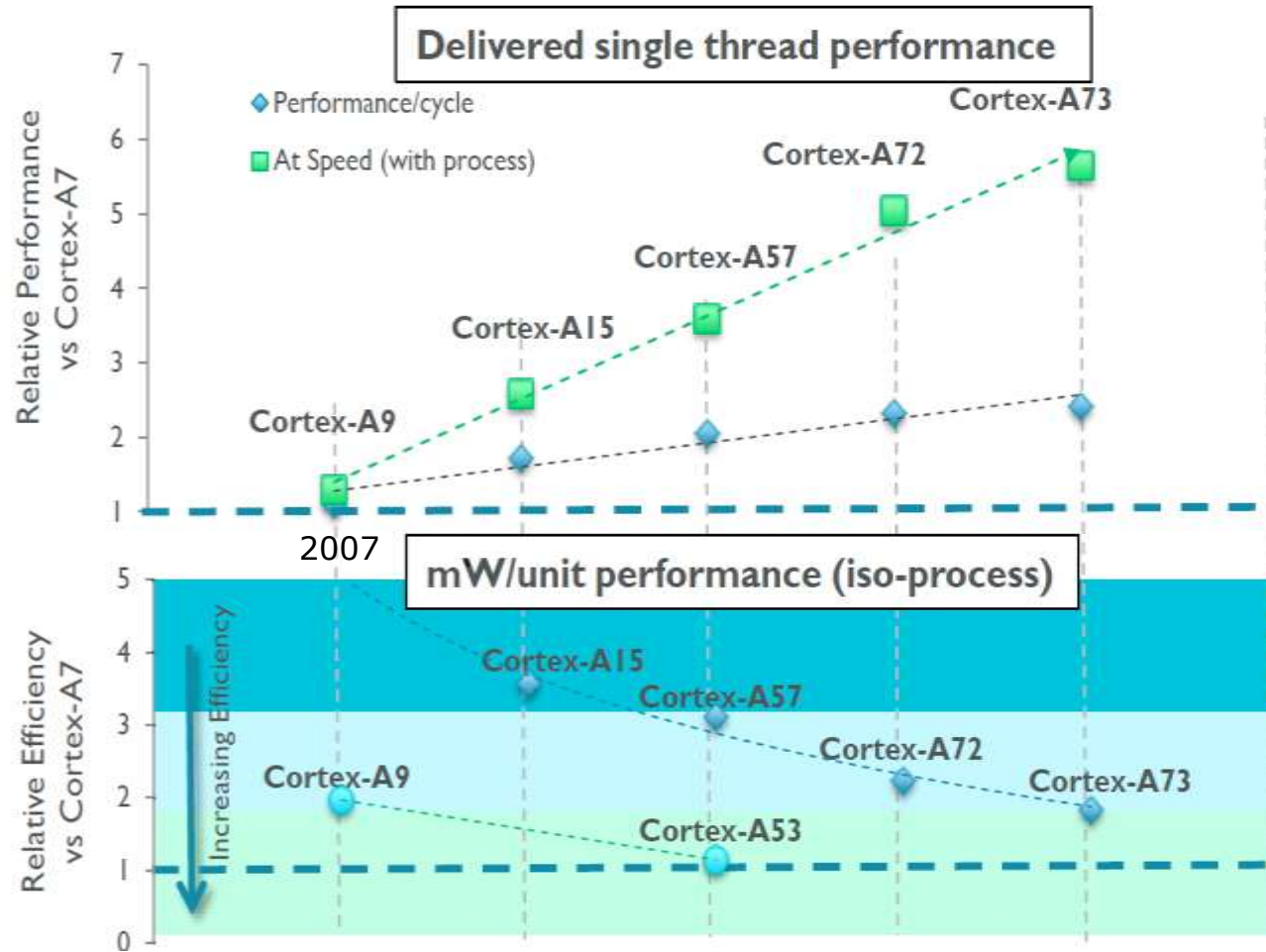


The high performance Cortex-A73 -2 [92]

- **05/2016: Announced** along with the G71 graphics processor with immediate availability for licensing.
- **First premium level mobile devices** with Cortex-A73 in 2017.
- The Cortex-A73 is **successor to the** high performance 64-bit **Cortex-A72**.
But whereas the Cortex-A72 was designed in Austin, Texas, the Cortex-A73 is a designed from scratch by a French team in Sophia-Antipolis design park.
The design started in 2013 **inspired by** the mainstream **Cortex-A17**.
- The Cortex-A73 **can be used either as stand alone processors or as part of a big-LITTLE configuration along sith the Cortex-a53/A35 processors.**
Interoperability with the ARM Mali GPU family is provided.
- Target process technology: **10 nm FinFET technology.**

5.2.3 The high performance Cortex-A73 (3)

Relative performance, efficiency and power efficiency of subsequent models of the Cortex-A line [92]



Accelerated growth in:

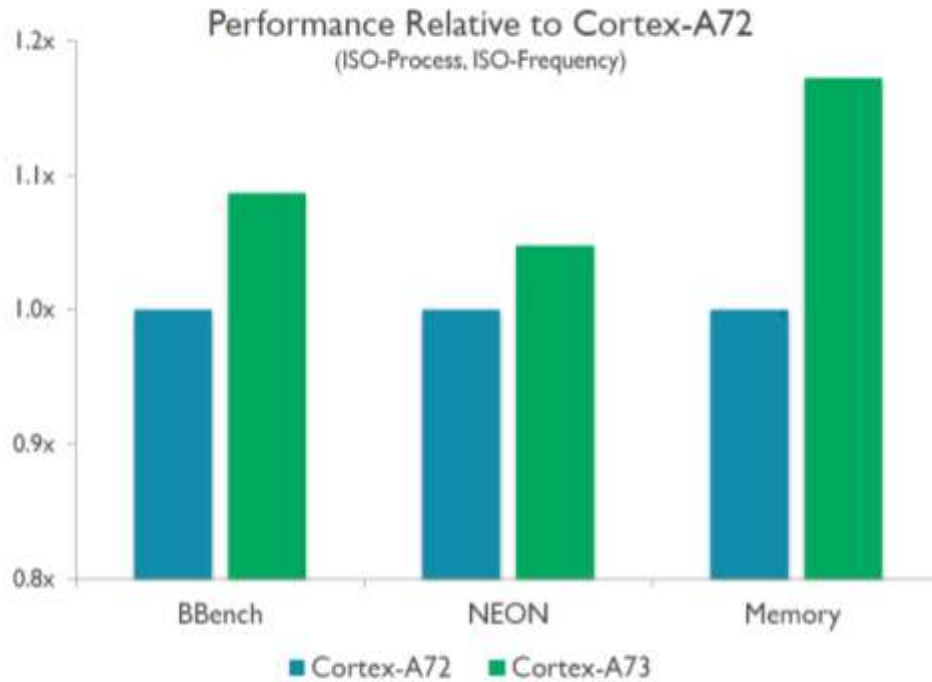
- Performance/Cycle
- Delivered performance

AND

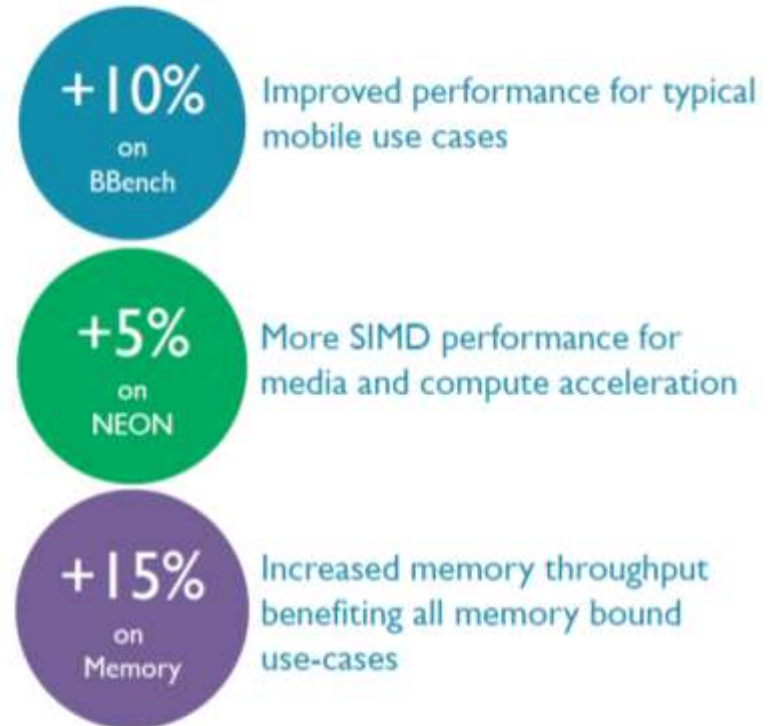
Continuous improvement
in power efficiency

5.2.3 The high performance Cortex-A73 (4)

Cortex-A73 performance improvement over the Cortex-A72 [92]



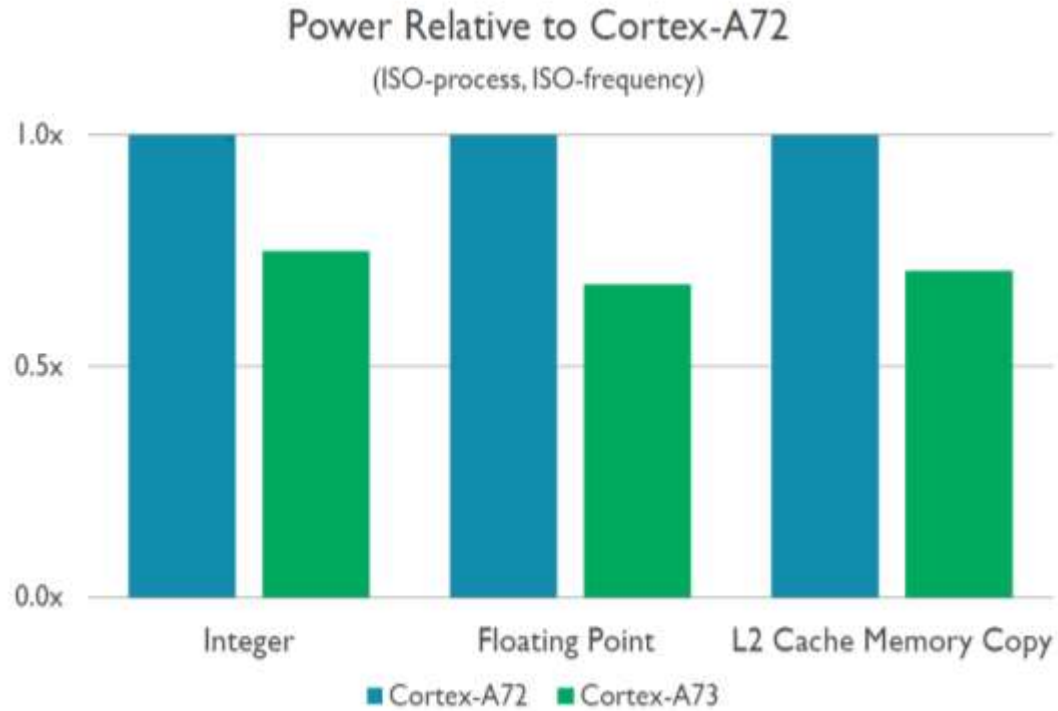
Improvement over Cortex-A72



Memory : JMC Stream Copy 64-bit, NEON : Video FFMEG Codec
Bbench : Bbench v1.1

5.2.3 The high performance Cortex-A73 (5)

Cortex-A73 power reduction over the Cortex-A72 [92]



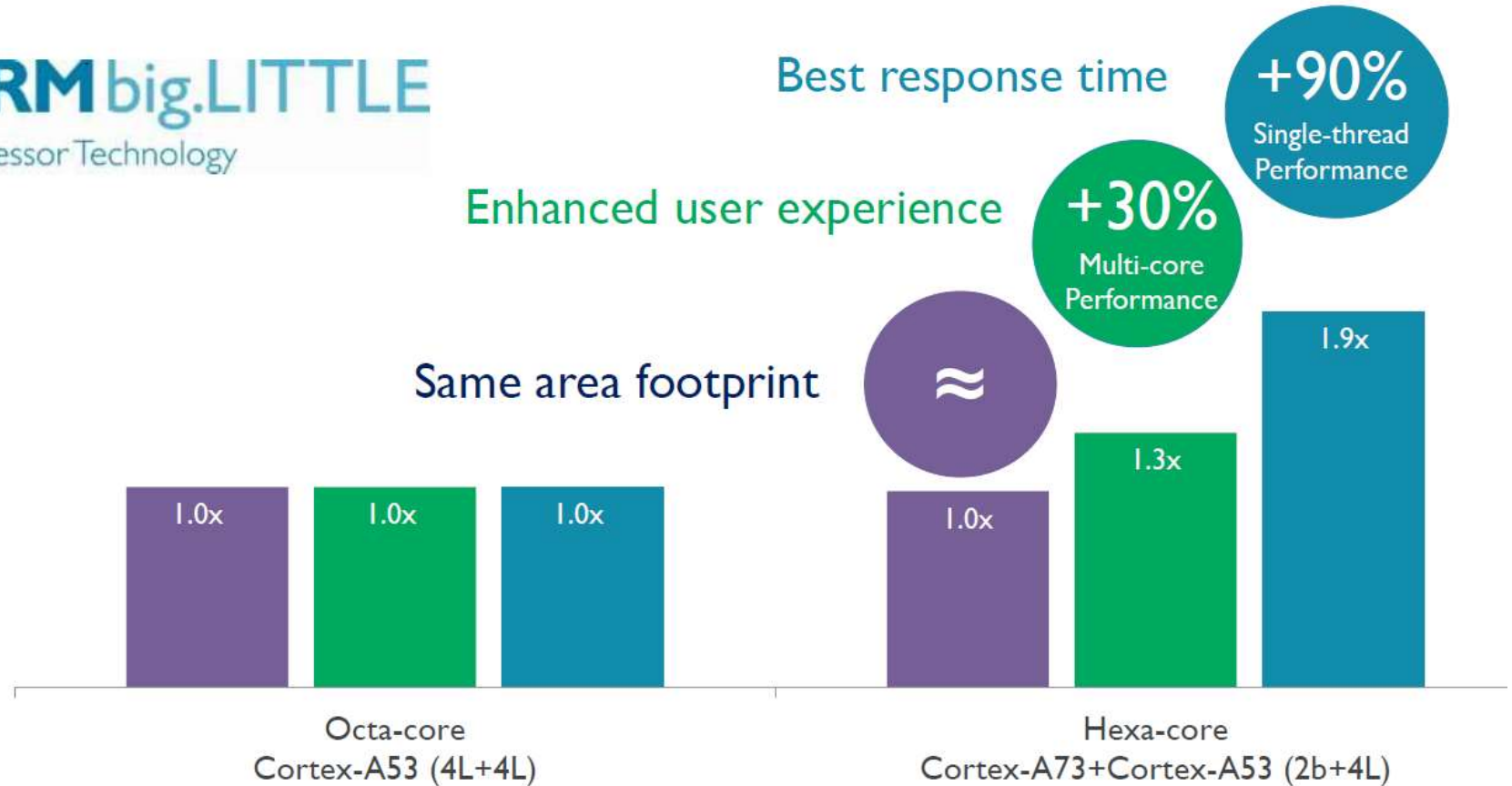
>20%
Power Saved

Giving additional thermal headroom for the rest of the SOC

5.2.3 The high performance Cortex-A73 (6)

Cortex-A73 performance improvement in a big-LITTLE configuration [92]

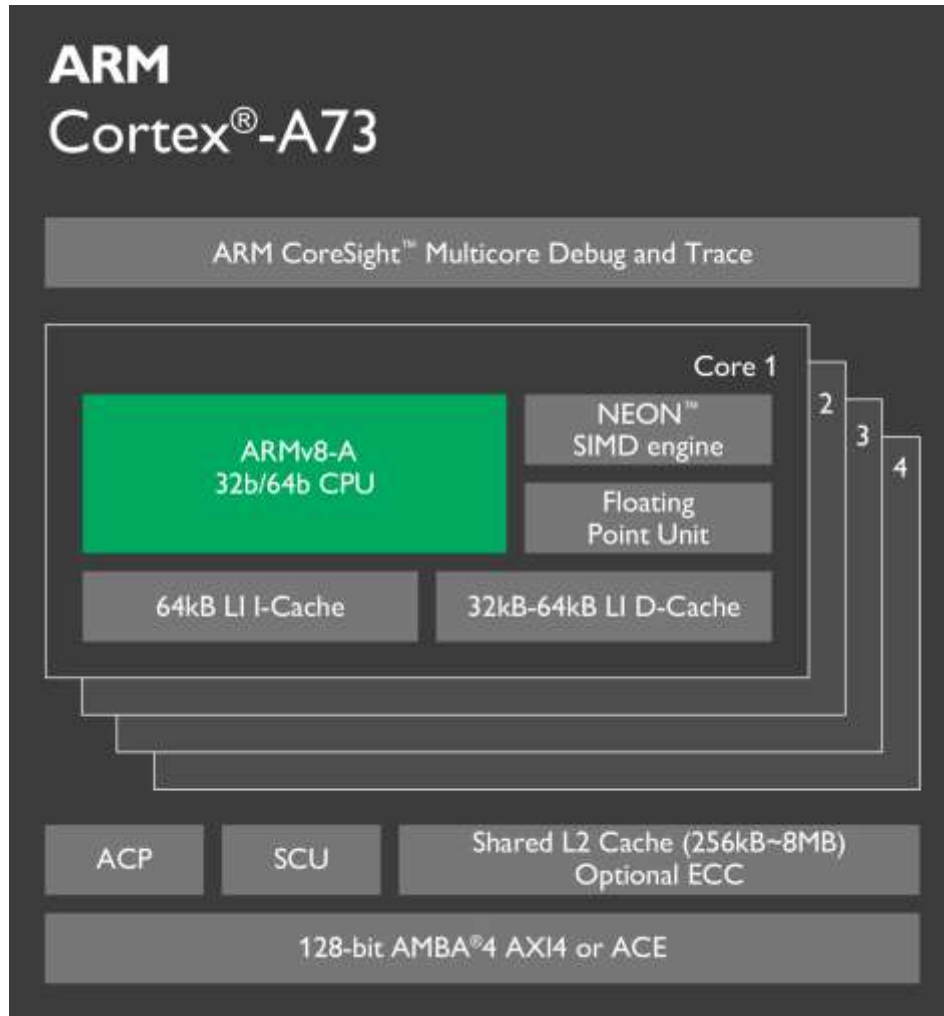
ARM big.LITTLE
Processor Technology



More performance, same footprint

5.2.3 The high performance Cortex-A73 (7)

High level block diagram of the Cortex-A73 [93]



ECC: **E**rror **C**orrecting **C**ode

ACP: **A**ccelerator **C**oherence **P**ort
(to connect non-cached coherent data sources)

SCU: **S**noop **C**ontrol **U**nit
(Provides coherence within the the processor)

AMBA: **A**dvanced **M**icrocontroller **B**us **A**rchitecture
(On-chip bus standard for SoC designs)

AXI4: **A**dvanced **e**Xtensible **I**nterface
(The most widespread AMBA4 non cache-coherent interface)

ACE: **A**XI **C**oherency **E**xtensions
AMBA4 cache coherent interface
(Used in big.LITTLE systems for smartphones, tablets, etc.).

Key features of the microarchitecture of the Cortex-A73 [92]

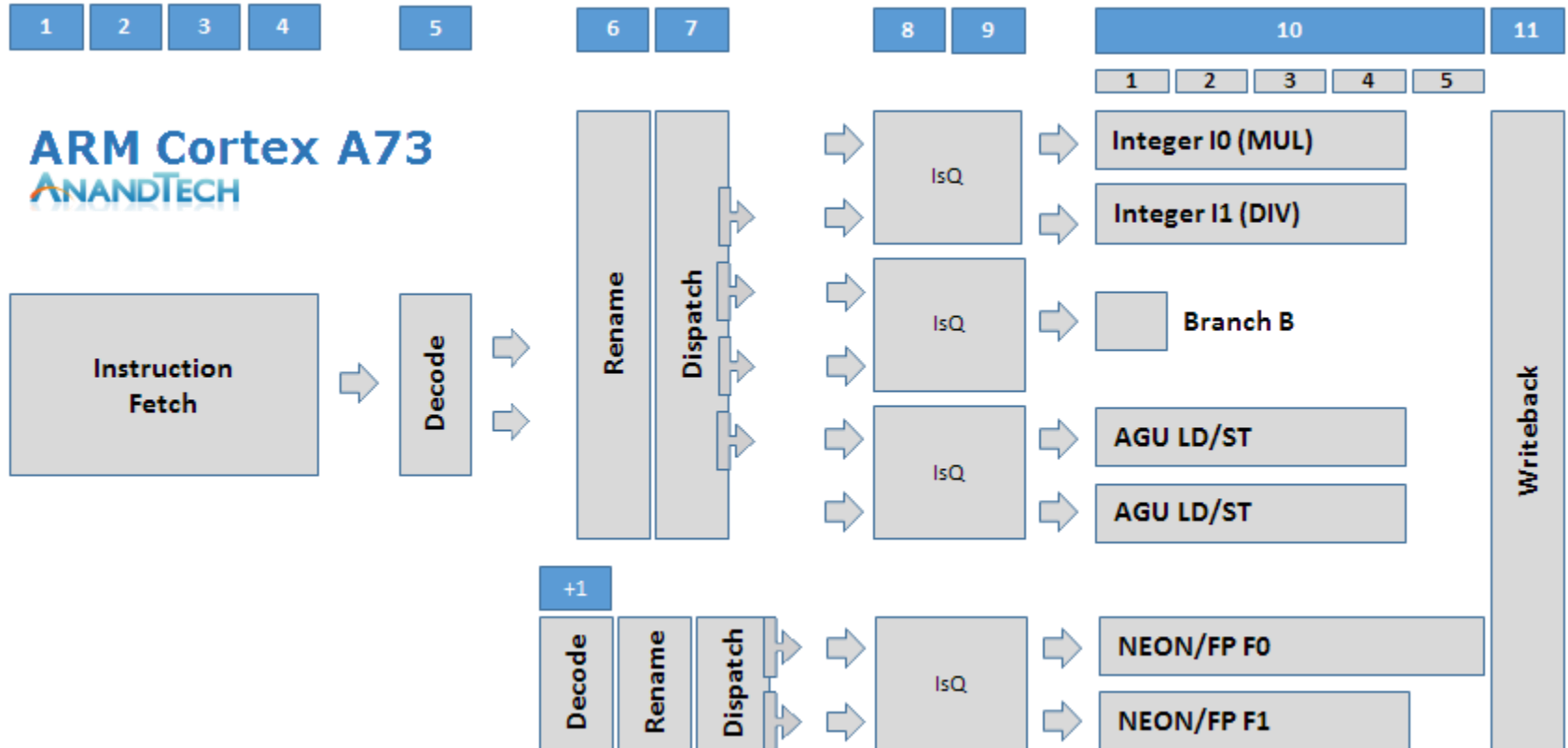
- It has a two-wide out-of-order front-end and a four-wide out-of-order back-end. This is in contrast to the Cortex-A72 that has a three-wide in-order front-end and a five-wide out-of-order back-end (see subsequent Figures).
- Integrated L2 cache of 256 kB to 8 MB.
- Up to 4 Cortex-A73 CPUs executing the ARMv8 64-bit ISA.
- 128-bit AMBA AXI4 non-coherent or ACE coherent interface for connecting multiple (up to 4) Cortex-A57 processors and further system components.
- Processor wide cache coherence supported by a Snoop Control Unit (SCU).

Remarks

AMBA: Advanced Microcontroller Bus Architecture
(On-chip bus standard for SoC) designs

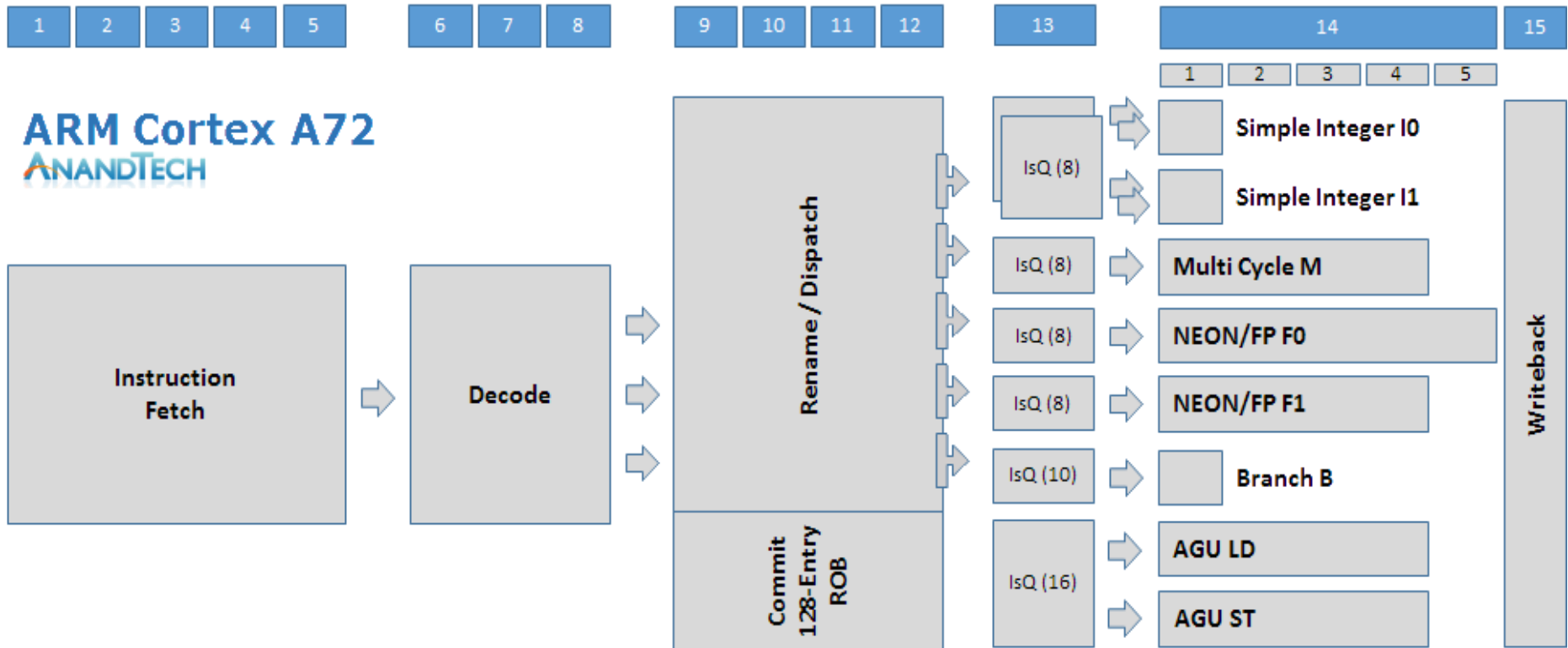
5.2.3 The high performance Cortex-A73 (9)

Pipeline structure of the high-end Cortex-A73 [92]



5.2.3 The high performance Cortex-A73 (10)

By contrast: Pipeline structure of the high-end Cortex-A72 [92]

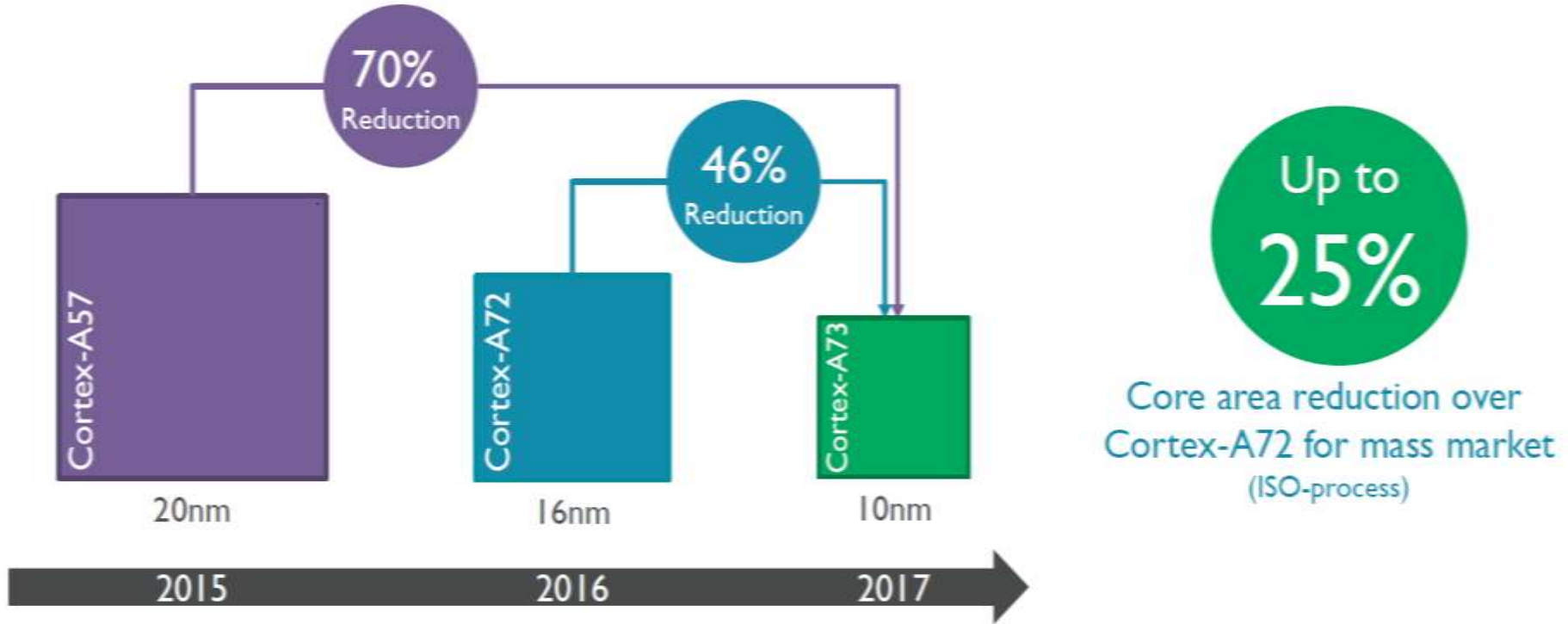


Key differences in the pipeline structure of the Cortex-A73 and -A72 [92]

- The Cortex-A73 has a **shorter pipeline** than the -A72 (11+ stages vs. 14+ stages).
- The Cortex-A73 has a **smaller front-end** than the -A72 (a dual-issue out-of-order front-end vs. a three issue out-of-order front-end).
- The Cortex-A73 has also a **smaller back-end** than the -A72 (four-wide out-of order back-end with an issue rate of 7 vs. a five-wide out-of order back-end with an issue-rate of 10).
- Still the Cortex-A73 has a **higher efficiency** (7.4-8.5 DMIPS/MHz vs. 6.3-7.35 DMIPS/MHz), achieved through a superior design.

5.2.3 The high performance Cortex-A73 (12)

Die size reduction in ARM's high-performance 64-bit processor line [92]



Cortex-A73 reduces footprint allowing additional silicon area for other IP

5.2.3 The high performance Cortex-A73 (13)

Range of scalability of the Cortex-A73 processor [92]

Premium smartphone configuration

Performance optimized

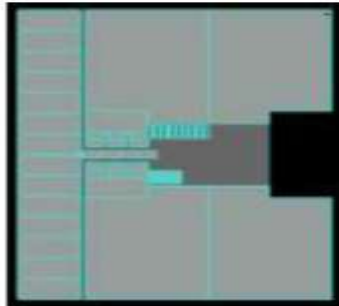
I0FF

Quad core, ~2.8 GHz

64K/64K L1, 2M L2

High Performance libraries

~5 mm²



Mass market consumer configuration

Area optimized

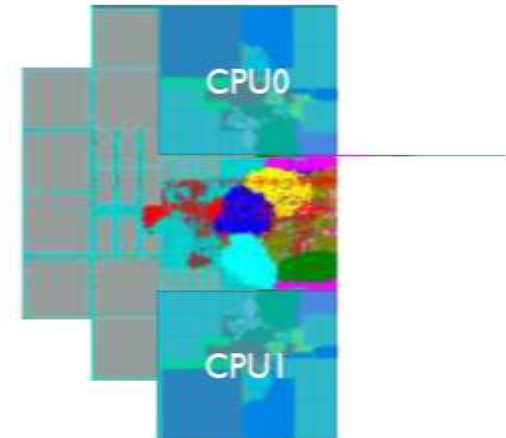
28HPC

Dual core, ~2GHz

32K/64K L1, 1M L2

High Density Libraries

~6 mm²



ARM

Remark

A detailed description of the microarchitecture of the cortex-A73 can be found in [92].

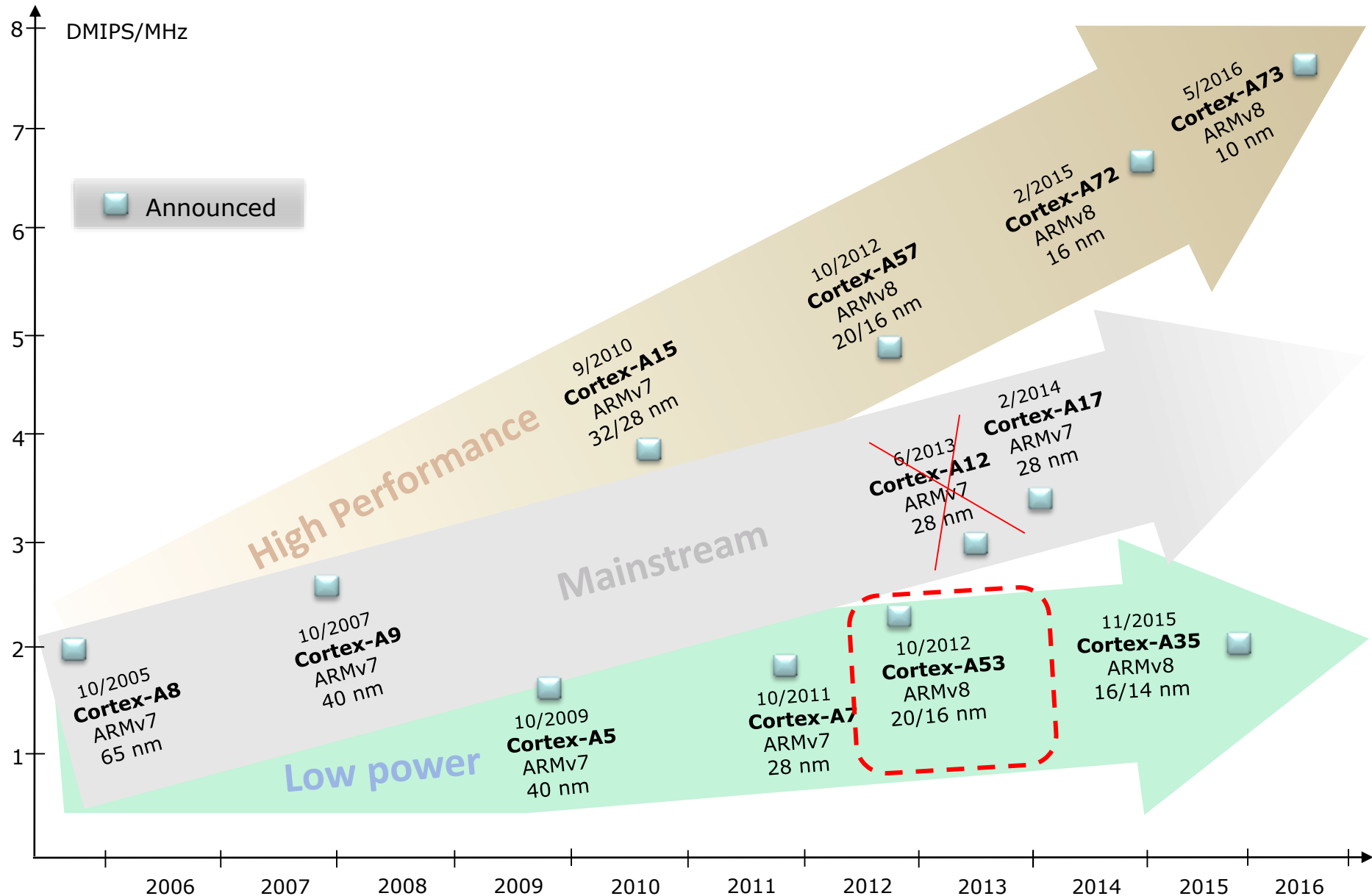
5. 3 The low power Cortex-A models based on the ARMv8.0 ISA

- 5.3.1 The low power Cortex-A53
- 5.3.2 The low power Cortex-A35

5.3.1 The low power Cortex-A53

5.3.1 The low power Cortex-A53 (1)

5.3.1 The low power Cortex-A53 -1 (based on [12])

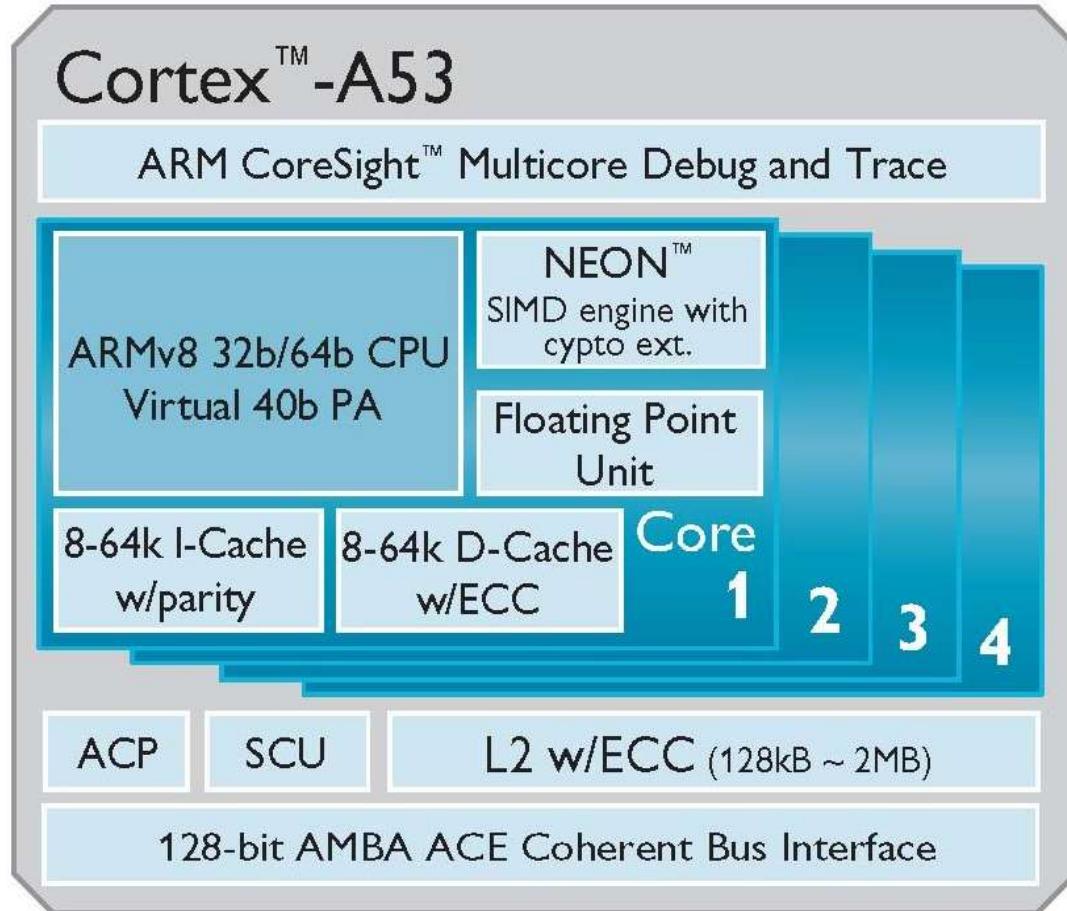


The low power Cortex-A53 -2 [75]

- **10/2012**: Announced along with the performance oriented Cortex-A57 processor, with immediate availability for licensing.
- **2014**: First mobile devices with the Cortex-A53.
- It is the **64-bit successor to the Cortex-A7**.
- The A57 and A53 models can be used either as stand alone processors or as components in a big-LITTLE configuration, similar to the 32-bit Cortex-A15 and A7 models.
- Target process technology: **16 or 20 nm**.

5.3.1 The low power Cortex-A53 (3)

High level block diagram of the Cortex-A53 [53]



PA: **P**hysical **A**ddress

ECC: **E**rror **C**orrecting **C**ode

ACP: **A**ccelerator **C**oherence **P**ort
(to connect non-cached coherent data sources)

SCU: **S**noop **C**ontrol **U**nit

AMBA: **A**dvanced **M**icrocontroller **B**us **A**rchitecture
(On-chip bus standard for SoC designs)

ACE: **A**XI **C**oherency **E**xtensions
AMBA4 cache coherent interface
(Used in big.LITTLE systems for smartphones, tablets, etc.)

5.3.1 The low power Cortex-A53 (4)

Key features of the microarchitecture of the Cortex-A53 core -1

- Fully compatible with the preceding ARMv7 32-bit ISA.
- Integrated L2 cache of 128 kB to 2 MB.
- Up to 4 Cortex-A53 CPU cores that execute the ARMv8 64-bit ISA.
- 128-bit AMBA coherent interface for connecting multiple (up to 4) Cortex-A53 processors.
- Processor wide cache coherence supported by a Snoop Control Unit (SCU).

Remarks

AMBA: Advanced Microcontroller Bus Architecture
(On-chip bus standard for SoC) designs

5.3.1 The low power Cortex-A53 (5)

Key features of the microarchitecture of the Cortex-A53 core -2

- The **Cortex-A53 core** has a **dual-issue in-order pipeline** actually with 5 individual pipelines constituting the back end, as indicated in the next Figure.

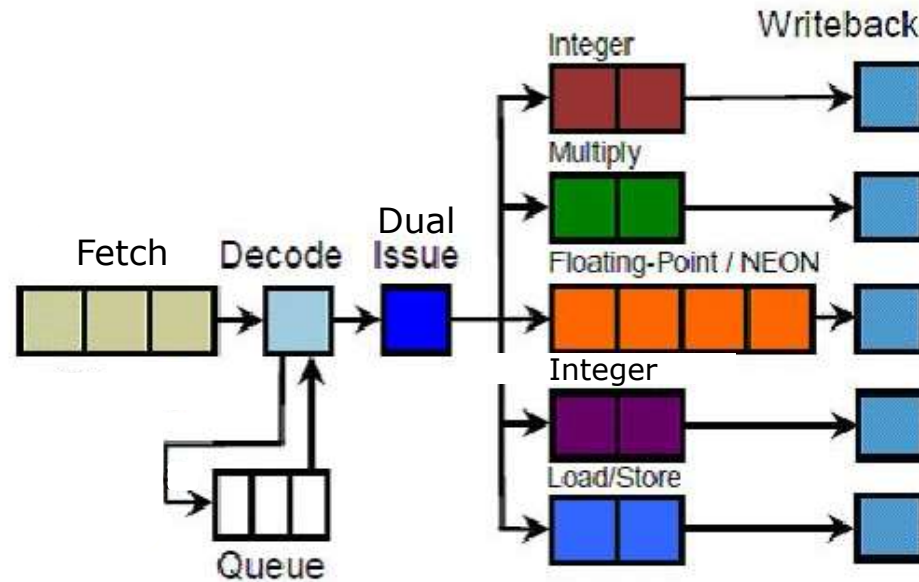


Figure: Pipeline stages of the Cortex-A53 [76]

- The pipeline for integer processing has 8 pipeline stages, NEON and FP processing has two additional pipeline stages, as seen in the Figure above.

5.3.1 The low power Cortex-A53 (6)

Contrasting the Cortex-A7 and Cortex-A53 microarchitectures [75]

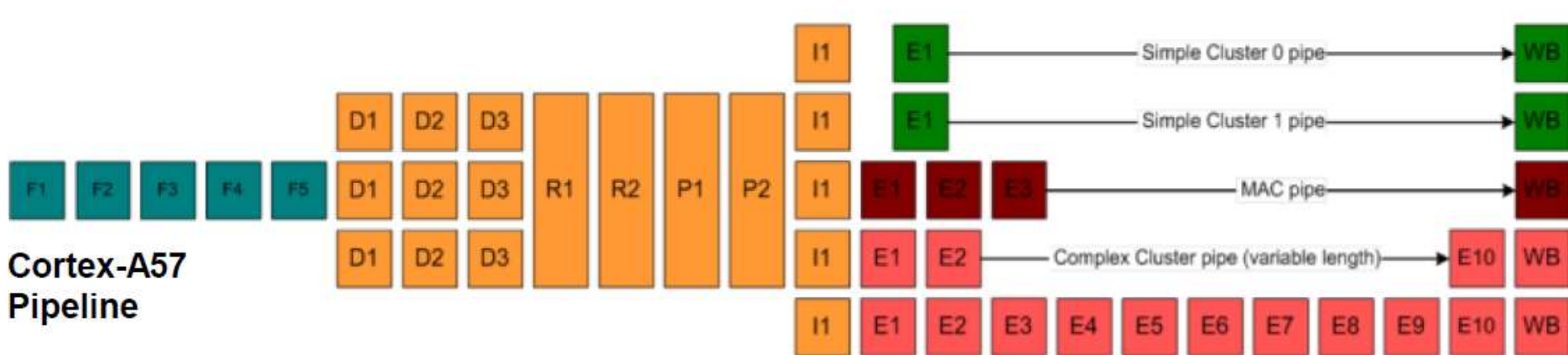
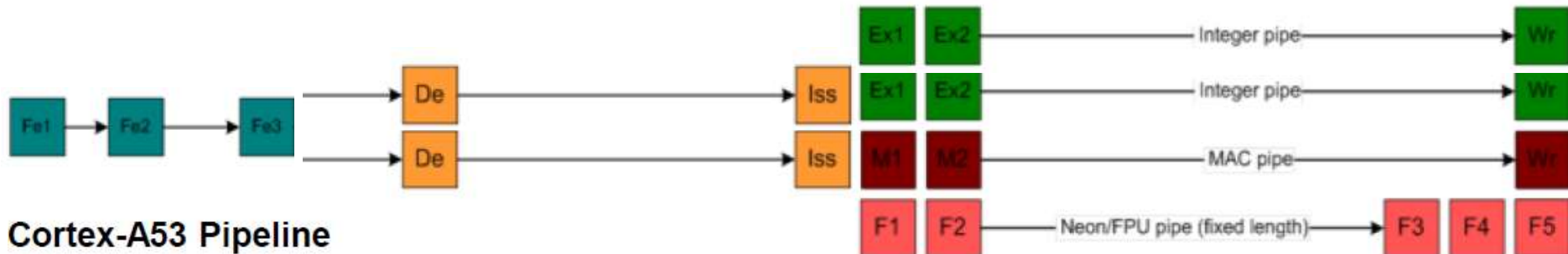
ARM CPU Core Comparison		
	Cortex-A7	Cortex-A53
ARM ISA	ARMv7 (32-bit)	ARMv8 (32/64-bit)
Issue Width	Partially 2 micro-ops	2 micro-ops
Pipeline Length	8	8
Integer Add units	2	2
Integer Mul unit	1	1
Load/Store Units	1	1
Branch Unit	1	1
FP/NEON ALU	1x64-bit	1x64-bit
L1 Cache	8KB-64KB I\$ + 8KB-64KB D\$	8KB-64KB I\$ + 8KB-64KB D\$
L2 Cache	128KB - 1MB (Optional)	128KB - 2MB (Optional)

Remarks [75]

- We note that the **Cortex-A7** has actually a partially dual-issue capability, meaning that the second issue slot can only issue branch and integer operations.
- In the **Cortex-A53** the second issue slot can also issue load-store and FP/NEON operations.
- In addition, the branch prediction capabilities of the A53 were also significantly improved by including conditional and indirect jump predictors.

5.3.1 The low power Cortex-A53 (8)

Contrasting the Cortex-A53 and Cortex-A57 arithmetic pipelines
 [Based on 54]



D: Decode
 R: Rename
 P: Dispatch
 I: Issue
 E: Execute
 WB: Write Back

Note: Branch and Load/Store pipelines not shown
 (1x Load/Store pipeline for the Cortex A-53 and
 2x Load/Store and 1x Branch pipeline for the Cortex-A-57)

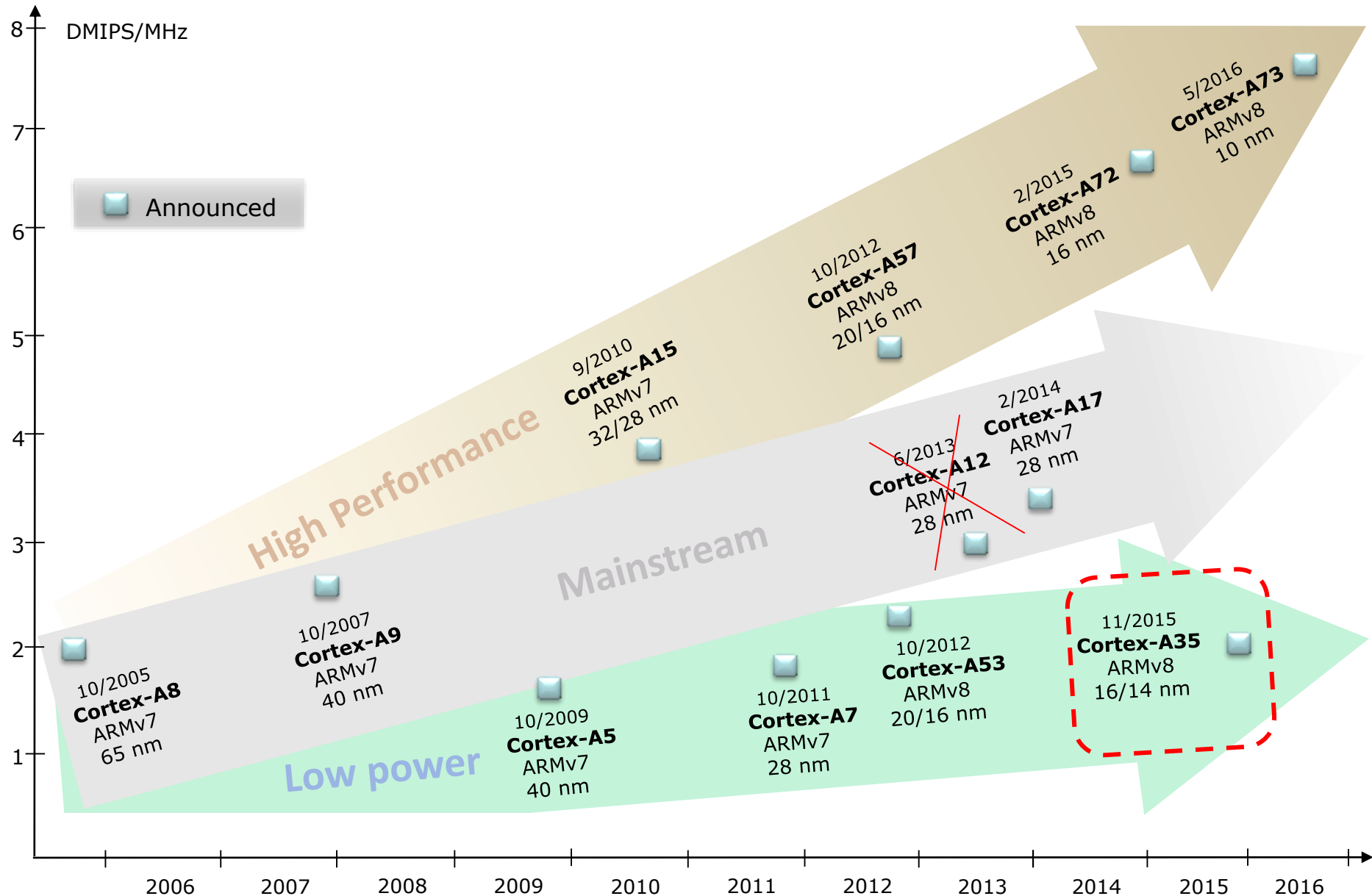
Key features of the microarchitecture of the Cortex-A53 core -3

- Core efficiency: 2.3 DMIPS/MHz
- Expected core frequency up to 1.5 GHz.

5.3.2 The low power Cortex-A35

5.3.2 The low power Cortex-A35 (1)

5.3.2 The low power Cortex-A35 -1 (based on [12])

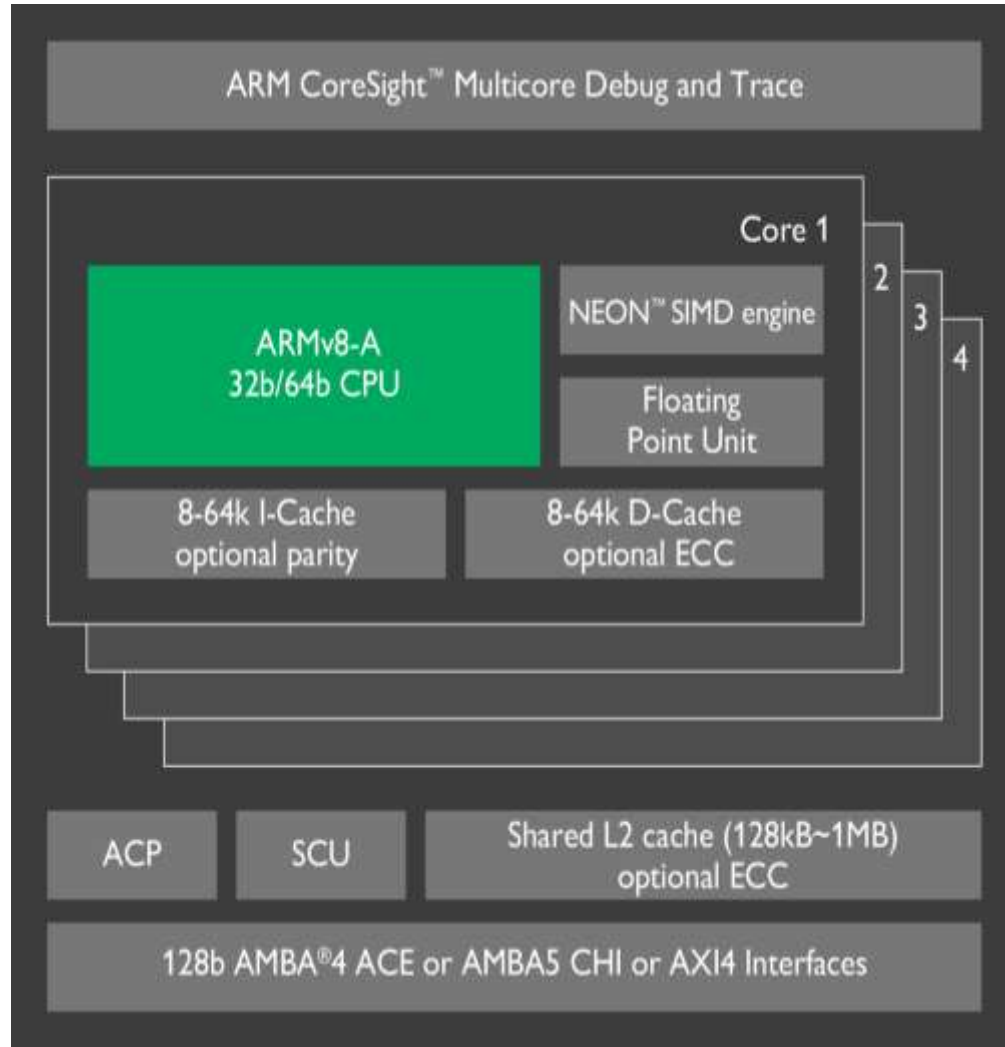


The low power Cortex-A35 -2 [90]

- **11/2015**: Announced with immediate availability for licensing.
- **End of 2016**: First mobile devices with the Cortex-A35 expected.
- It is the **successor to the Cortex-A53**.
- It is the **smallest, most efficient** model of the Cortex series.
- The A35 models can be **used either as stand alone processors or as components in a big-LITTLE configuration**.
- Target process technology: **16/14 nm**.

5.3.2 The low power Cortex-A35 (3)

High level block diagram of the Cortex-A35 [94]



ECC: **E**rror **C**orrecting **C**ode

ACP: **A**ccelerator **C**oherence **P**ort
(to connect non-cached coherent data sources)

SCU: **S**noop **C**ontrol **U**nit
(Provides coherence within the processor)

AMBA: **A**dvanced **M**icrocontroller **B**us **A**rchitecture
(On-chip bus standard for SoC designs)

ACE: **A**XI **C**oherency **E**xtensions
AMBA4 cache coherent interface
(Used in big.LITTLE systems for smartphones, tablets, etc.)

CHI: **C**oherent **H**ub Interface
(AMBA5 Cache coherent interface used in servers)

AXI4: **A**dvanced **e**Xtensible **I**nterface
(The most widespread AMBA4 non cache-coherent interface).

Key features of the microarchitecture of the Cortex-A35 core -1 [90]

- Fully compatible with the ARMv8 64-bit ISA.
- In-order 8-stage pipeline with limited dual issue capability
- Integrated L2 cache of 128 kB to 1 MB.
- Up to 4 Cortex-A35 CPU cores.
- Processor wide cache coherence supported by a Snoop Control Unit (SCU).

Optionally three 128-bit AMBA interfaces

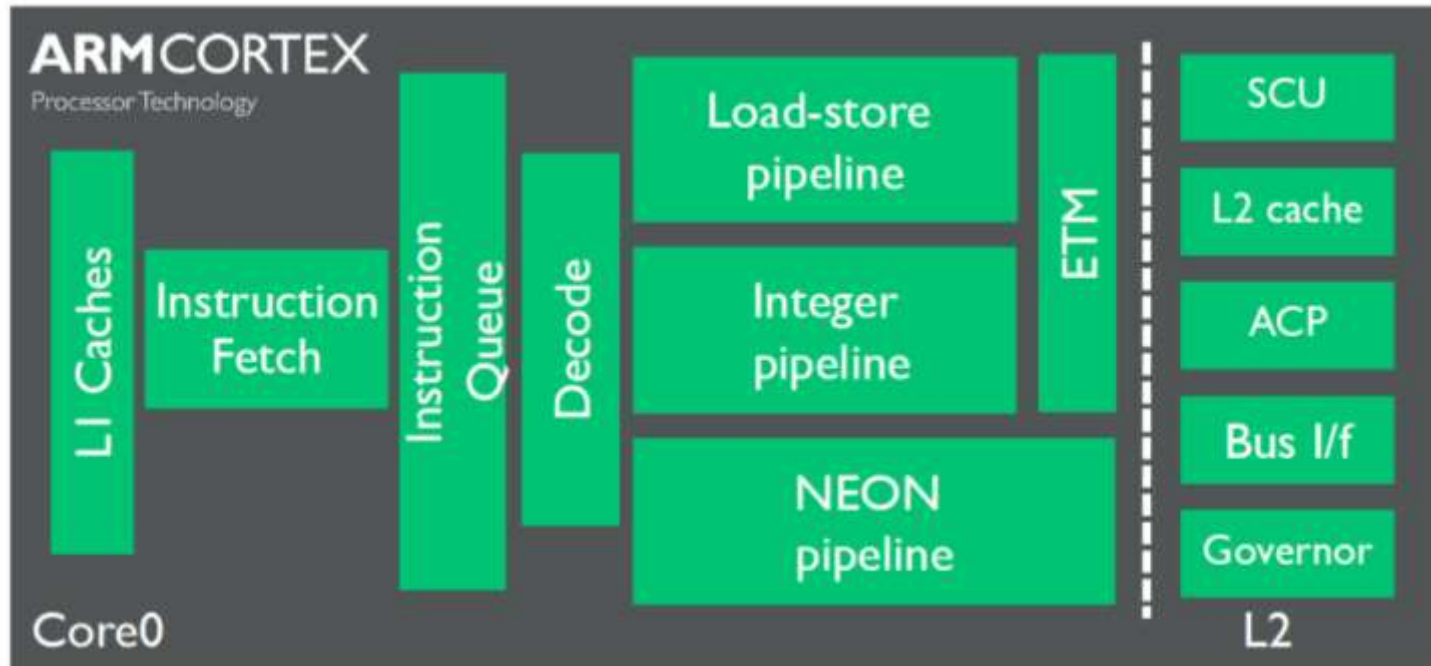
- AXI4: **A**dvanced **eX**tensible **I**nterface
(The most widespread AMBA4 non cache-coherent interface).
- ACE: **A**XI **C**oherency **E**xtensions
AMBA4 cache coherent interface used in big.LITTLE systems for smartphones, tablets, etc.)
- CHI: **C**oherent **H**ub Interface
(AMBA5 Cache coherent interface used in servers)

Remark

AMBA: Advanced Microcontroller Bus Architecture
(On-chip bus standard for SoC) designs

5.3.2 The low power Cortex-A35 (5)

The in-order 8-stage pipeline of the Cortex-A35 providing limited dual issue capability Cortex-A35 [90]

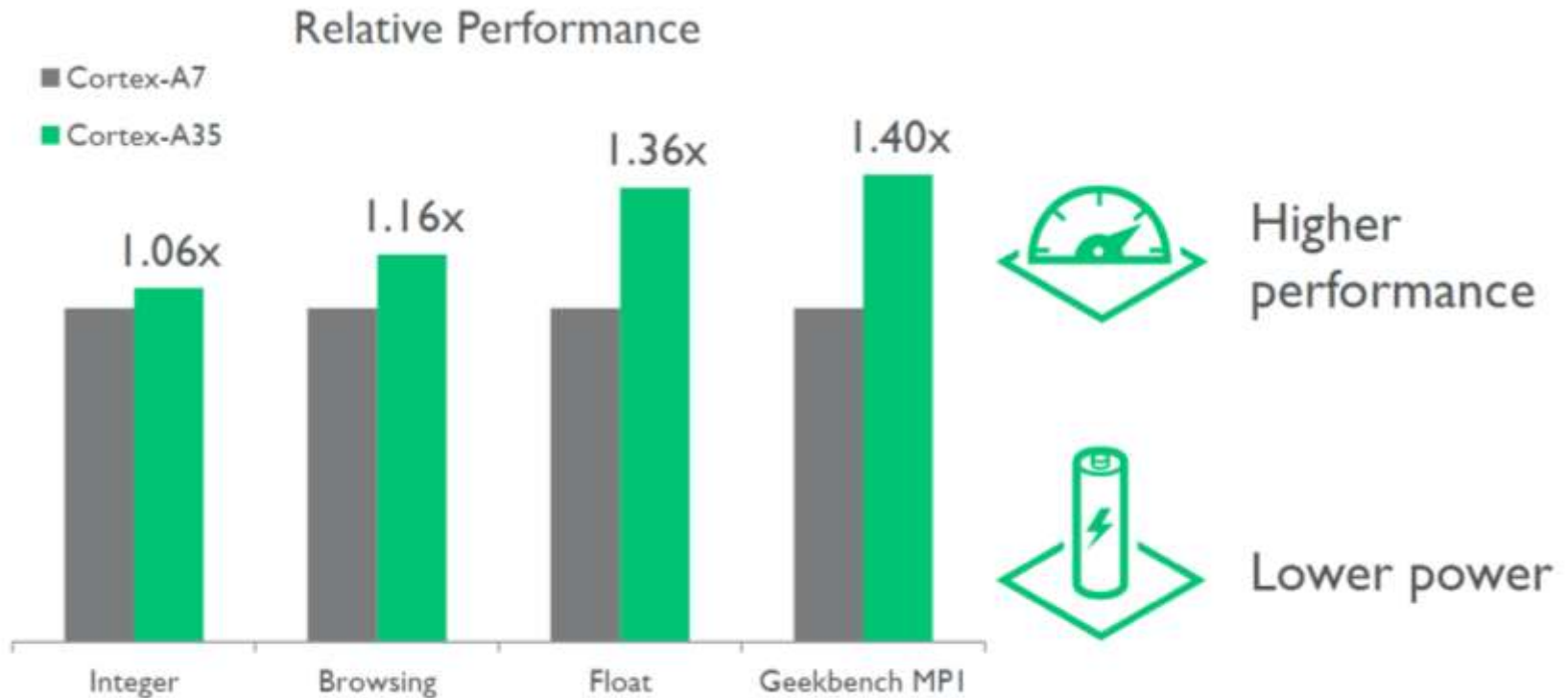


ETM: Embedded Trace Macrocell, a low-level debugging technology in ARM microprocessors.

Governor: Hardware unit for setting the processor into the standby state.

5.3.2 The low power Cortex-A35 (6)

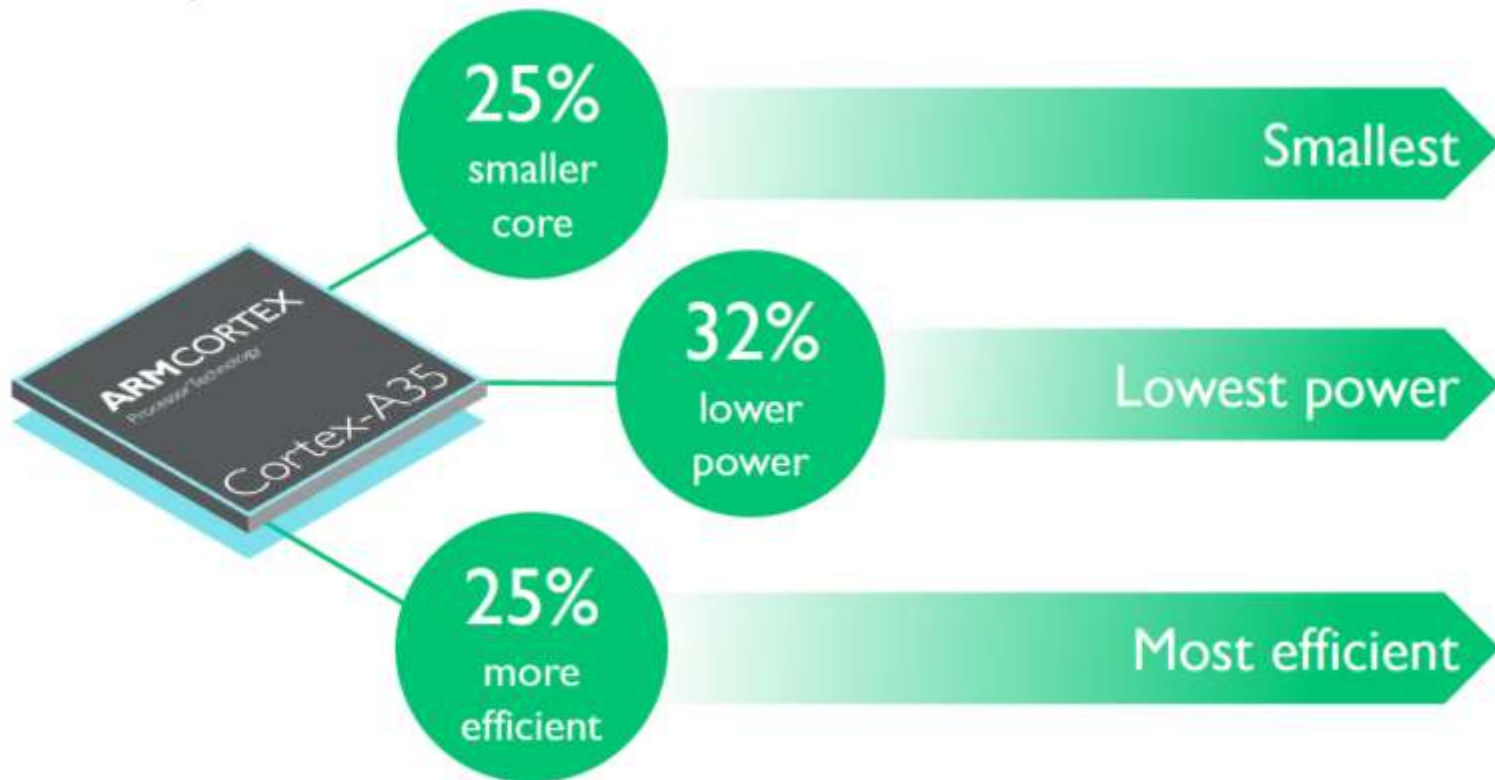
Relative performance of the Cortex-A35 vs. the Cortex-A7 assuming the same process technology (28 nm) [90]



Comparisons assume same process technology and implementation for both processors

5.3.2 The low power Cortex-A35 (7)

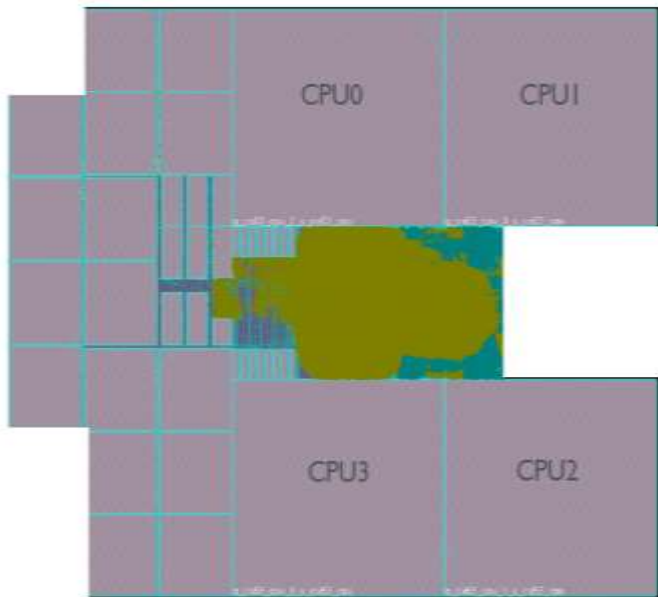
Relative power consumption of the Cortex-A35 vs. the Cortex-53 assuming the same clock frequency and process technology (28 nm) [90]



5.3.2 The low power Cortex-A35 (8)

The range of configurability of the Cortex-A35 processor [90]

Configuration options of the Cortex-A35 are ranging from mobile to deeply embedded.



Cortex-A35

Quad core configuration

32K LI caches, NEON, Crypto, 1MB L2 cache



>10x smaller



< 0.4 mm²
28nm

Cortex-A35

Smallest configuration

Single core, 8K LI caches, no L2

6. Cortex-A models based on the ARMv8.2 ISA

- 6.1 Overview
- 6.2 DynamIQ core clusters
- 6.3 The high performance Cortex-A75
- 6.4 The low power Cortex-A55

6.1 Overview

6.1 Overview

a) Main extensions of the ARMv8 ISA-1 [97]

Revision	Released	Main enhancements	Cortex-A processors
ARMv8-A	10/2011		A32/A35/ A53/A57/ A72/A73
ARMv8.1-A	12/2014	<ul style="list-style-type: none"> Incremental benefits over v8.0 relating to the instruction set, exception model and memory translation 	
ARMv8.2-A	03/2017	<ul style="list-style-type: none"> Half-precision FP Optional increase of the address space from 48 to 52 bits RAS extensions Statistical profiling to be able to analyze large working sets 	A55/A75 (Provides also ARMv8.1 support)

6.1 Overview (2)

a) Main extensions of the ARMv8 ISA-2 [97]

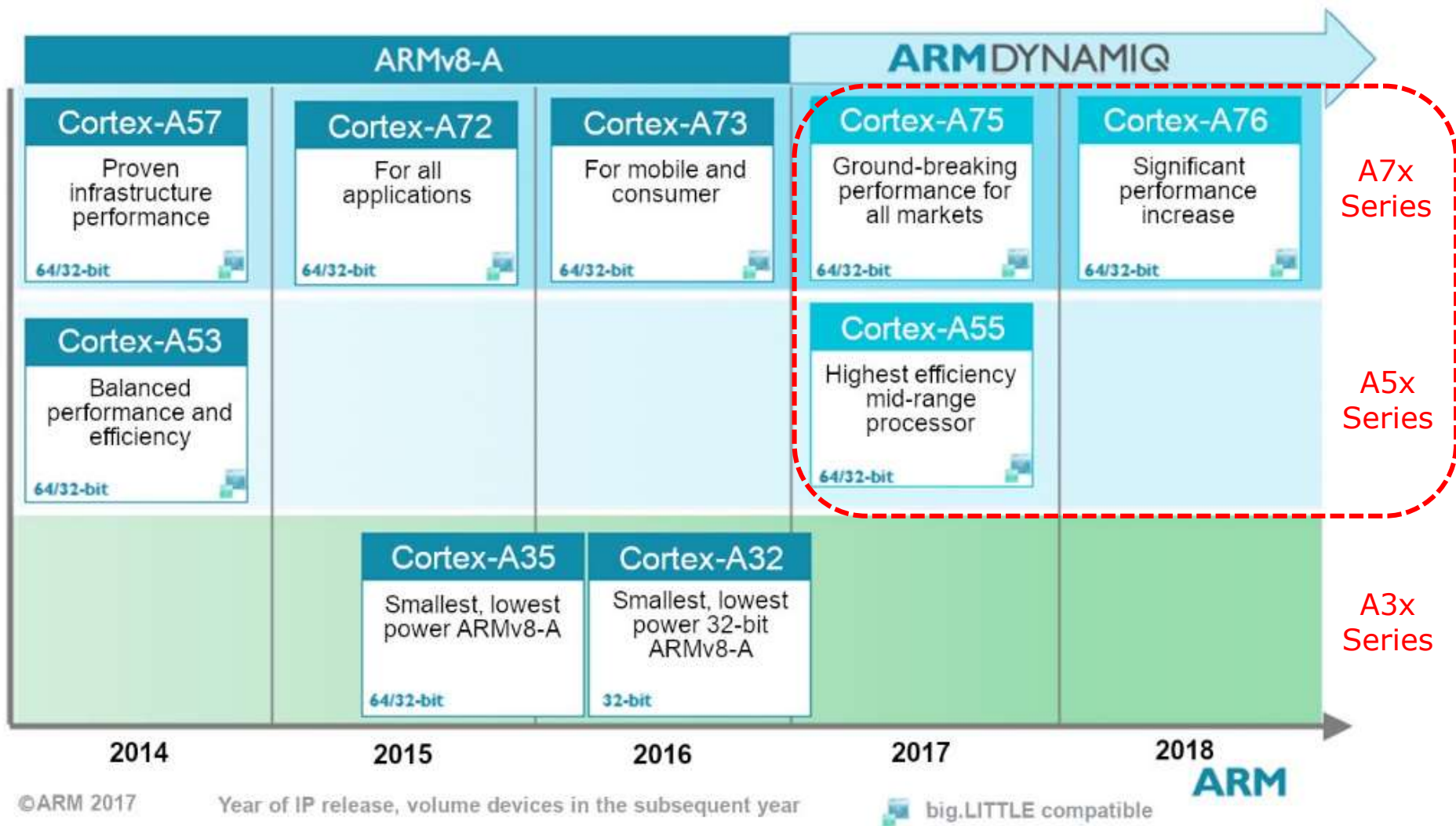
Revision	Released	Main enhancements	Cortex-A processors
ARMv8.3-A	12/2017	<ul style="list-style-type: none">• Pointer authentication• Improvements to the exception model for nested virtualization• Small-scale enhancements to the ISA	TBA
ARMv8.4-A	10/2018	<ul style="list-style-type: none">• Crypto extensions• Memory partitioning and monitoring capabilities• New Secure EL2 state	TBA
ARMv8.5-A	n.a.	<ul style="list-style-type: none">• Vulnerability Detection• Memory Tagging• Branch Target Indicators• Further small-scale enhancements	TBD

b) Main extensions of the microarchitecture of processors supporting the ARMv8.2 ISA

The main extension is the **DynamIQ core cluster**, described in Section 6.2.

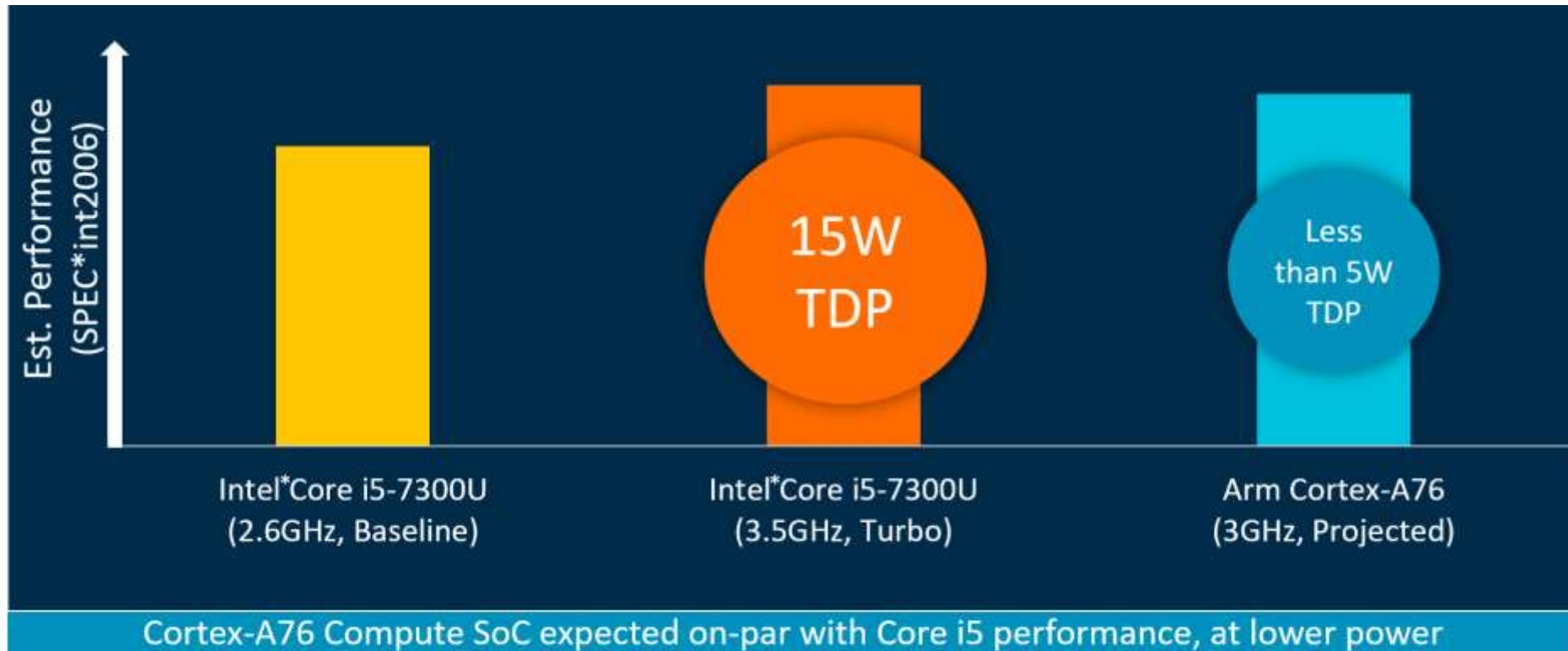
6.1 Overview (4)

ARMv8.2 ISA based processors - Overview



6.3 ARM's intention to enter the PC market

Based on SPECint 2006 benchmark results and related estimations ARM expects Cortex-A76-based processors to provide the same performance than Intel's 7. gen. (Kaby Lake based) i5-7300U processors, see Figure.



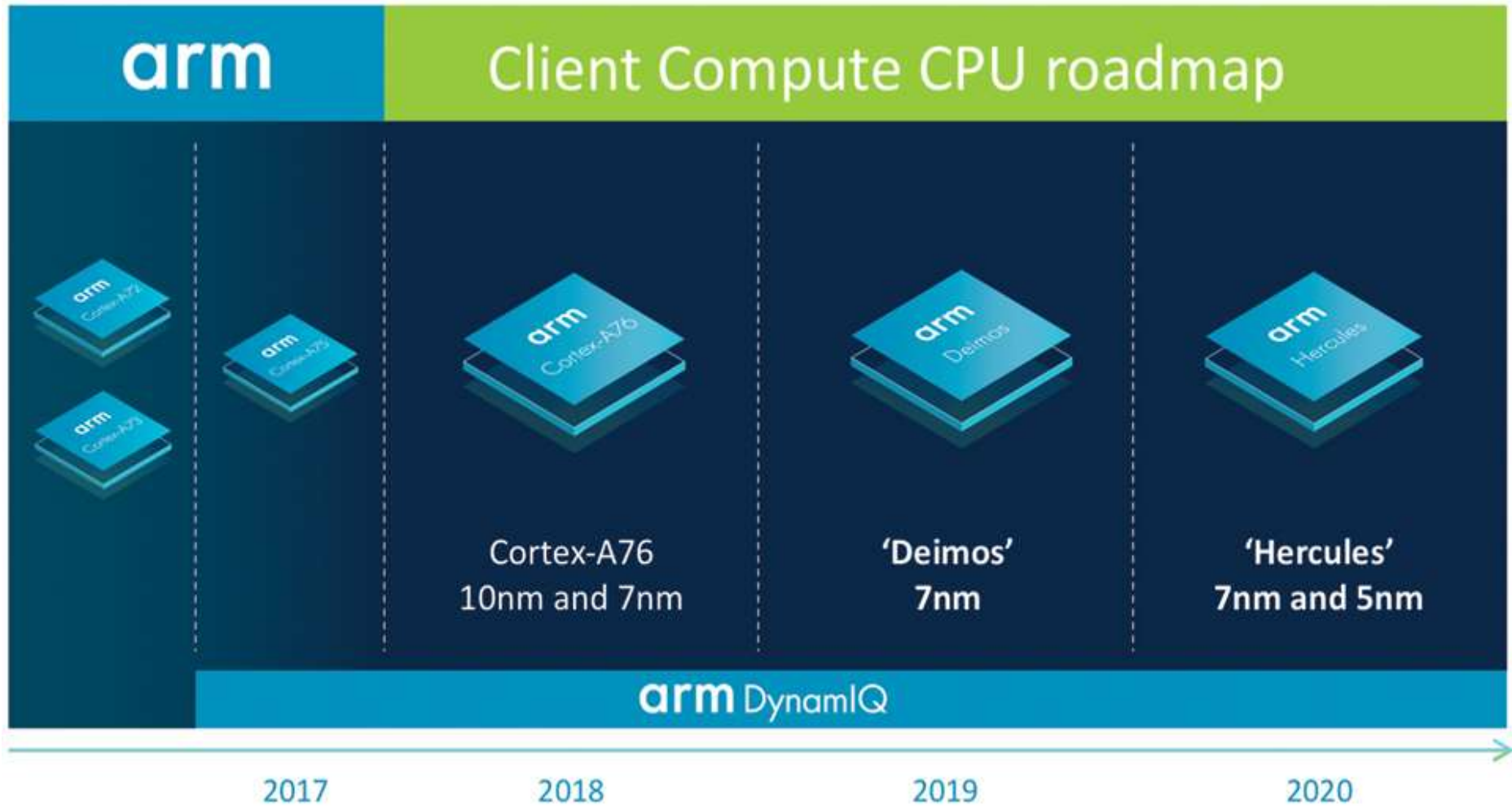
Arm power estimated as full SoC TDP for Compute platform. No indication of power for smartphone SoCs. Results median score of 10 runs at room temperature, screensaver disabled, using Ubuntu 18.04 in dual-boot configuration.

Measured estimates on SPECint*_base2006 (SPECspeed* Integer component of SPEC CPU* 2006) on Intel Core i5-7300U, and Arm Cortex-A76, 3GHz, 7nm, 4MB L3, 100ns LMBench load-to-use latency, using GCC 7.3 toolchain. Results are measured estimates using specific computer systems, software, components, operations, and functions and changes to any of these factors will cause the results to vary.

Figure: Performance comparison: ARM's Cortex-A76 vs. Intel Core i5-7300U [99]

6.1 Overview (6)

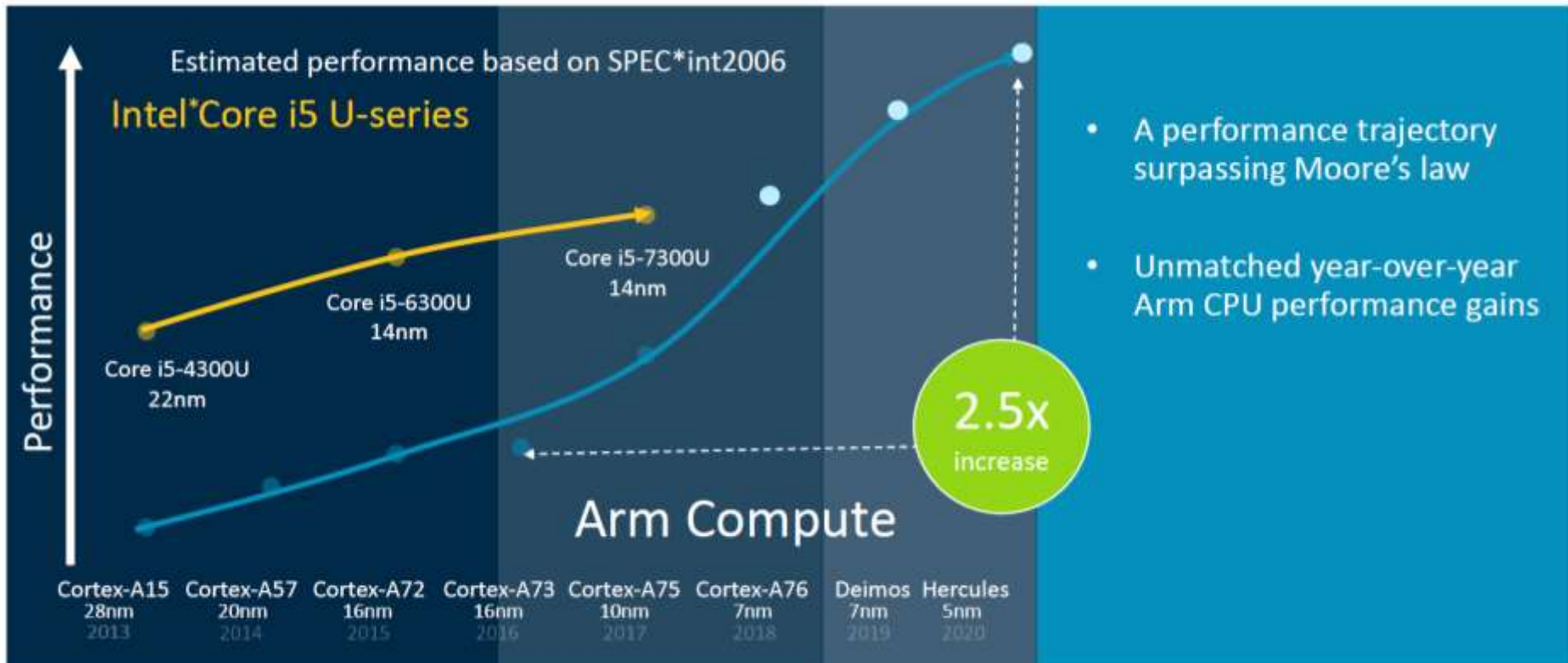
ARM's Client Compute CPU roadmap 2019-2020 [99]



6.1 Overview (7)

ARM's projected performance gain for their planned processors

ARM expects **faster performance gains** for their planned models than awaited for Intel's comparable processors, as seen in the Figure below.



Measured estimates on SPECint*_base2006 (SPECspeed* Integer component of SPEC CPU* 2006) on Intel Core i5-7300U, Core i5-6300U, Core i5-4300U, Arm single-core performance estimated for compute platform. Results are measured estimates using specific computer systems, software, components, operations, and functions and changes to any of these factors will cause the results to vary.

Figure: ARM's projected year-over-year performance gain over Intel's processors [100]

6.1 Overview (8)

Geekbench 4 scores of Intel's 15 W i7-xxxxU processors [116]

Model	CPU cores	Launched	GB 4 SC max	GB 4 MC max	Family	Techn.
I7-4500U	2C	6/2013	3854	6854	Haswell	22 nm
I7-4600U	2C	9/2013	4202	7592	Haswell	22 nm
i7-5600U	2C	1/2015	4217	8018	Broadwell	14 nm
I7-6600U	2C	9/2015	4775	9010	Skylake	14 nm
I7-7560U	2C	8/2016	5000	10182	Kaby Lake	14 nm
I7-7660U	2C	1/2017	5128	10389	Kaby Lake	14 nm
I7-8565U	4C	8/2018	5576	17813	Coffee Lake	14 nm

6.1 Overview (9)

Geekbench 4 scores of Apple's A series processors [114]

Model	CPU cores	Launched	GB 4 SC max	GB 4 MC max	Techn.
A8	2C	9/2014	1663	2855	20 nm
A8X	3C	10/2014	1798	4214	20 nm
A9	2C	9/2015	2524	4391	14/16 nm
A9X	2C	11/2015	3057	5114	16 nm
A10	2+2C	9/2016	3480	5928	16 nm
A10X	3+3C	6/2017	3915	9339	10 nm
A11	2+4C	9/2017	4224	10185	10 nm
A12	2+4C	9/2018	4797	11260	7 nm
A12X	4+4C	10/2018	5006	17925	7 nm

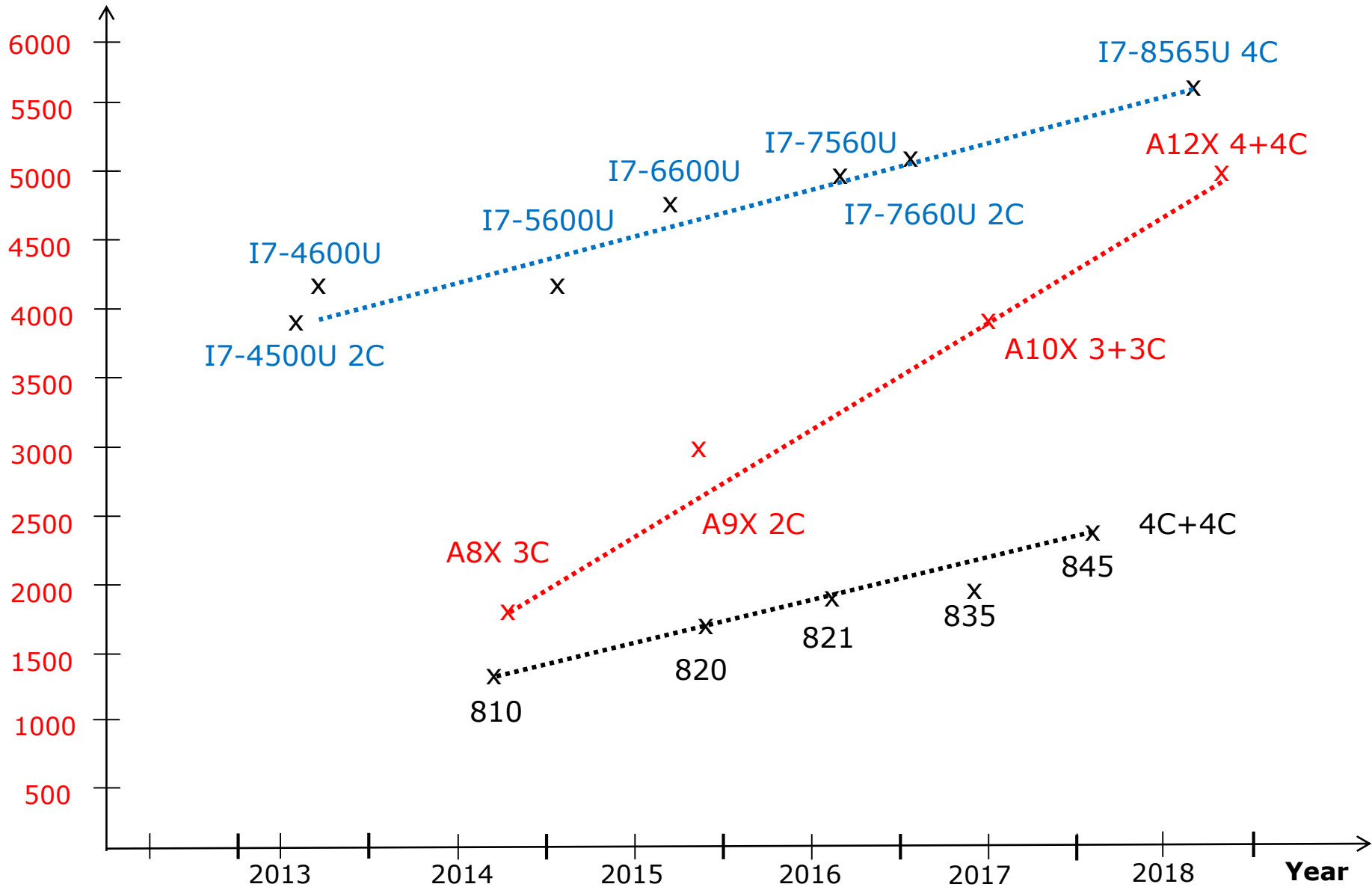
6.1 Overview (10)

Geekbench 4 scores of Qualcomm's Snapdragon processors [115]

Model	CPU cores	Launched	GB 4 SC max	GB 4 MC max	Techn.
808	2+4C	Q3/2014	1152	2813	20 nm
810	4+4C	Q3/2014	1351	3446	20 nm
820	4+4C	Q4/2015	1702	3955	14 nm
821	4+4C	Q3/2016	1880	4430	14 nm
835	4+4C	Q2/2017	1947	6624	10 nm
845	4+4C	Q1/2018	2415	8689	10 nm
850	4+4C	Q3/2018			10 nm
855	1+3+4C	Q1/2019			7 nm

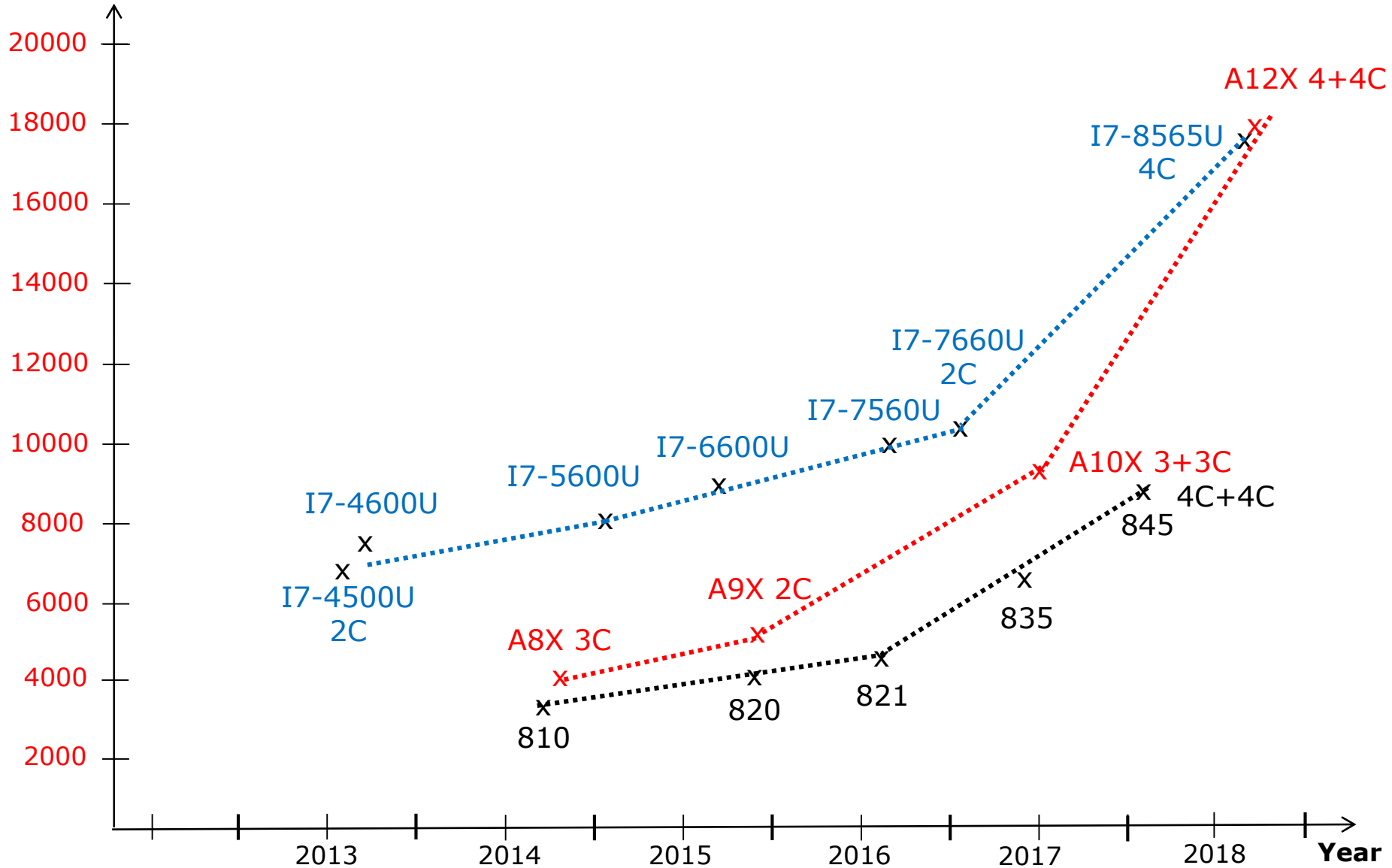
6.1 Overview (11)

Geekbench 4 SC scores of Intel's, Apple's and Qualcomm's processors



6.1 Overview (12)

Geekbench 4 MC scores of Intel's, Apple's and Qualcomm's processors



6.2 DynamIQ core clusters

DynamIQ core clusters -2

- ARM started working on the **DynamIQ** core cluster technology in 2013 [98].
- **DynamIQ** was announced in 03/2017.
- It is an **evolution of the big.LITTLE technology**, as indicated in the next slide.

6.2 DynamIQ core clusters -1

- They are the **next step** in the evolution of the core cluster technology.

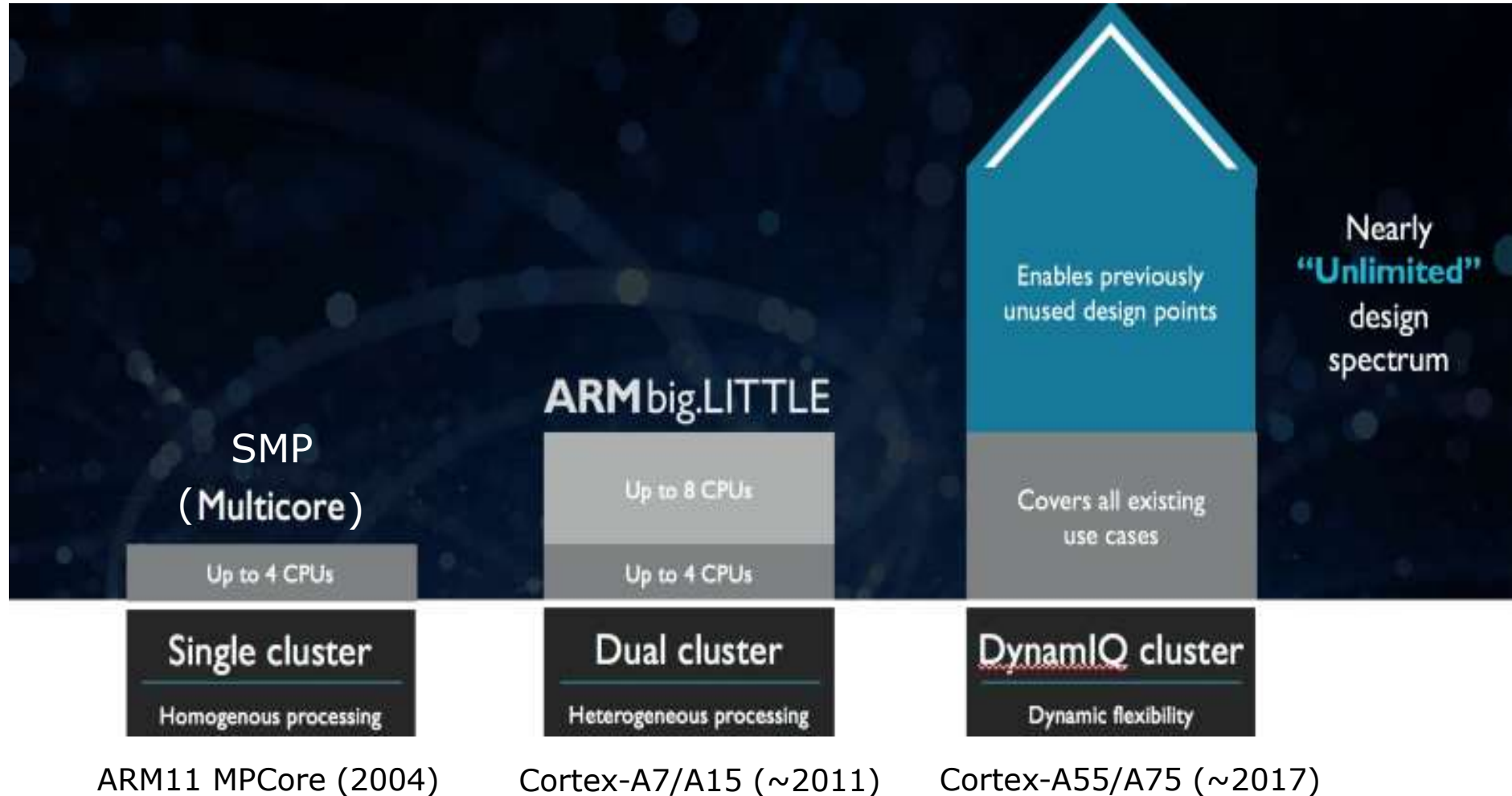
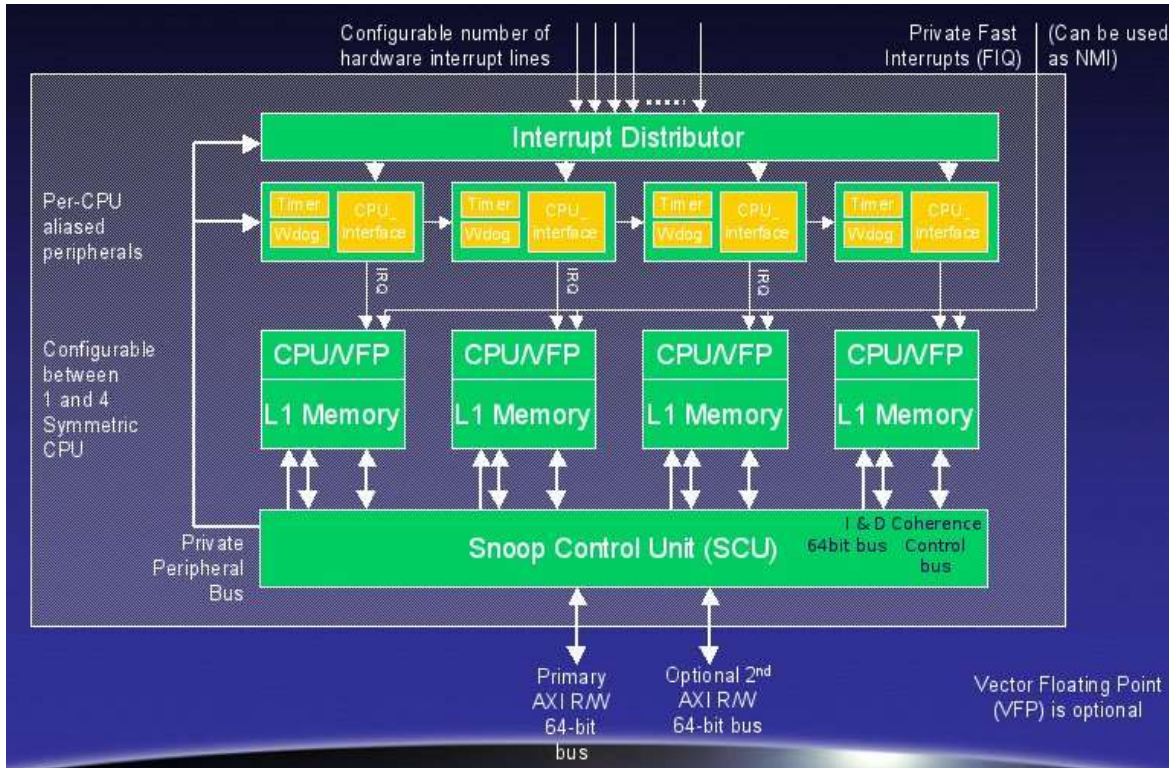


Figure: Evolution steps of the core cluster technology [114]

6.2 DynamIQ core clusters (3)

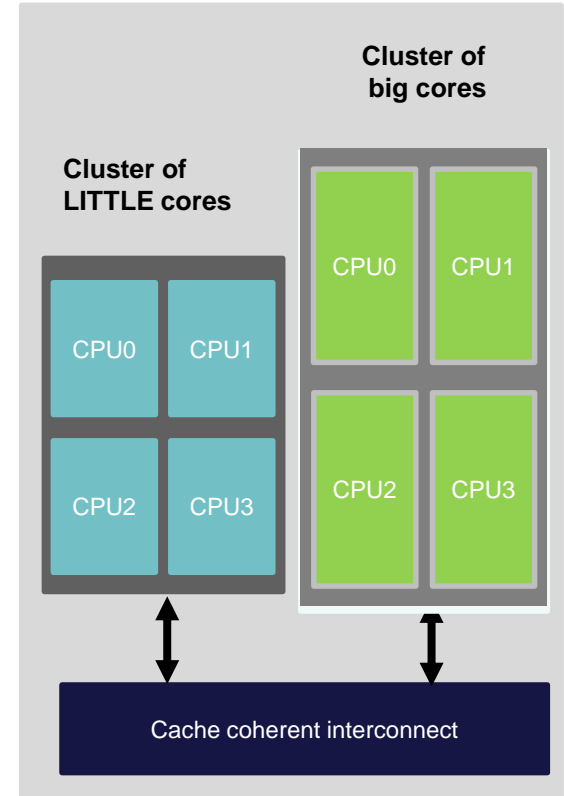
From the ARM11 MPCore to the big.LITTLE technology

ARM11 MPCore [115]



(2004)

big.LITTLE

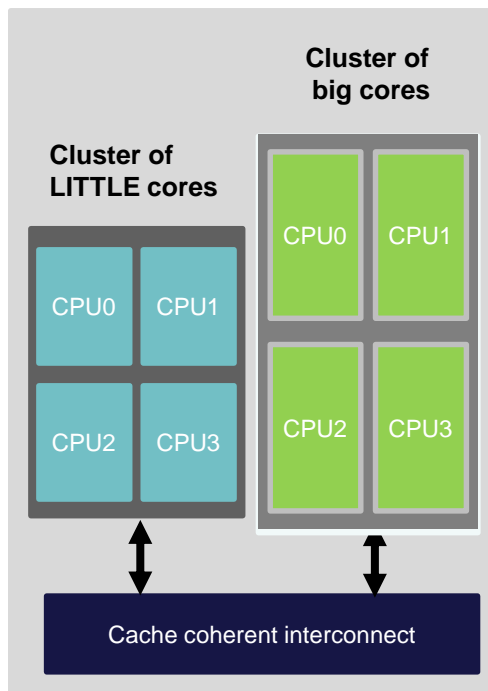


(2011)

6.2 DynamIQ core clusters (4)

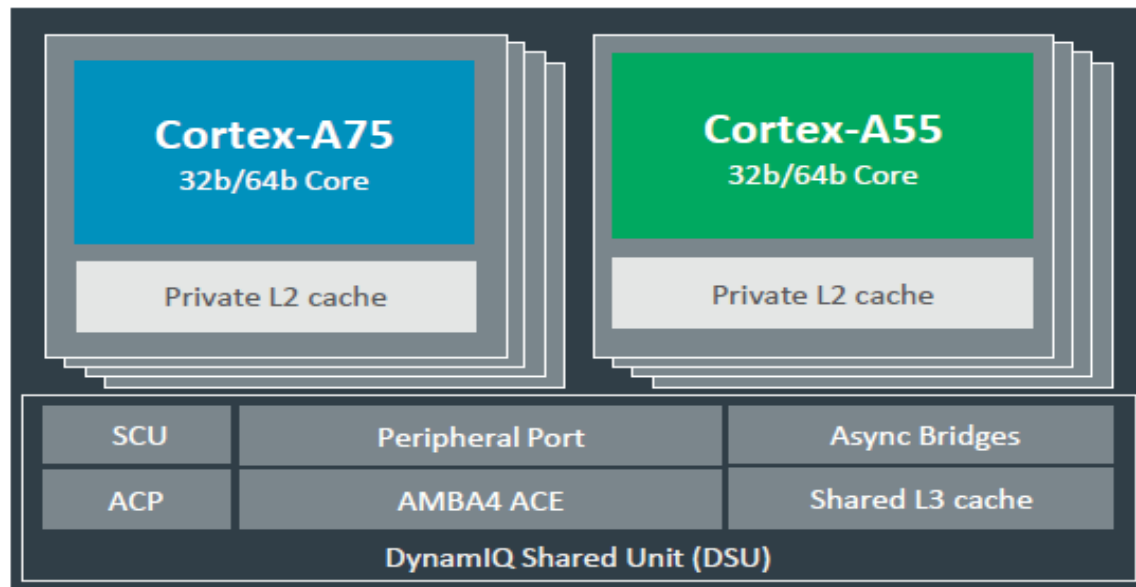
The DynamIQ technology as an evolution of the big.LITTLE technology

Two big.LITTLE core clusters



Two stand alone clusters with up to 4 cores (2011)

A single DynamIQ core cluster (118)



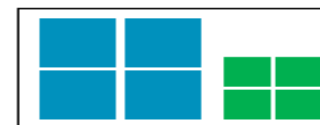
To cache coherent interconnect through the AMBA4 ACE bus



1b+7L



2b+6L



4b+4L



1b+2L



1b+3L



1b+4L

A single cluster of up to 8 cores of up to two core types (but up to 4 big cores) (2017)

DynamIQ core clusters -3

- **Benefits** of the DynamIQ cluster technology:
 - **greater flexibility** with or without LITTLE cores
 - redesigned memory subsystem with **higher bandwidth and lower access time**
 - **improved power efficiency** through **intelligent power management (EAS)**
EAS (Energy Aware Scheduling).

Remarks on EAS (Energy Aware Scheduling) [109]

- Existing Linux schedulers (like **the CFS (Completely Fair Scheduler)**) optimize for the throughput.
E.g. if a new task enters and there is an idle CPU the scheduler will always assign the new task to the idle CPU.
- However, this may not be the best scheduling decision for the lowest energy consumption.
- **EAS** is designed to **optimize for lowest possible energy consumption without affecting performance**.
- ARM started the development of EAS in 2015 while working with the Linux community.

DynamIQ core clusters -4

- It assumes [ARMv8.2 compatible cores](#), such as the Cortex-A55 and Cortex-A75
- or Cortex-A76.

Main enhancements of DynamIQ core clusters

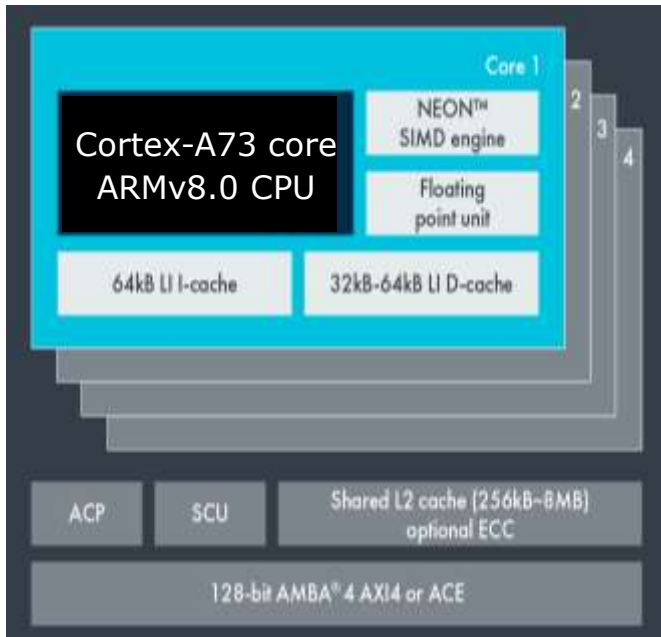
- a) Up to 8 CPU cores of up to 2 ARMv8.2 ISA based core types
- b) Private (per-core) L2 caches in the CPU cores
- c) DynamIQ Shared Unit (DSU) with a shared L3 cache and snoop filter
- d) Capability for partitioning the cores and the L3 cache
- e) Finer-grain frequency and voltage control
- f) Mesh interconnect (CMN-600) for server systems
- g) Use of a scratch pad system cache to increase throughput
- h) Cache stashing

(Features g) and h) will not be discussed)

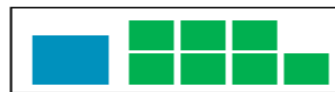
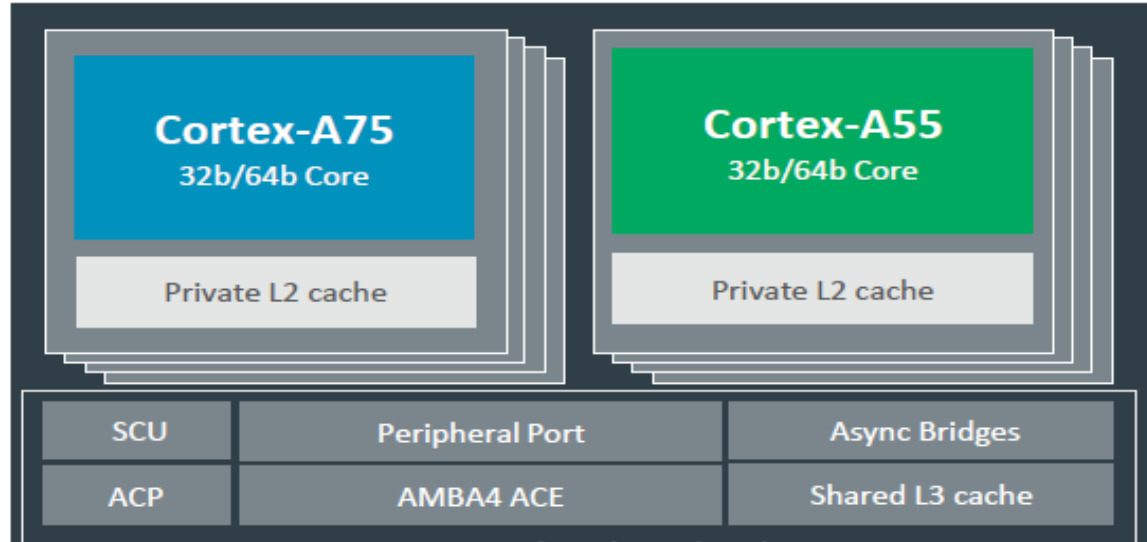
6.2 DynamIQ core clusters (7)

a) Up to 8 CPU cores of up to 2 ARMv8.2 ISA based core types

big.LITTLE core cluster (2015) [102]



DynamIQ core cluster (2017) [118]



1b+7L



2b+6L



4b+4L



1b+2L



1b+3L



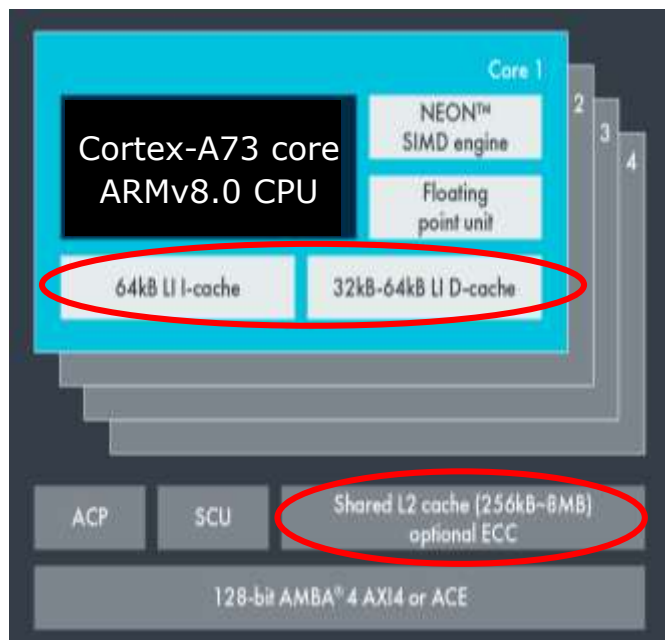
1b+4L

- ARMv8.0 support,
- e.g. Cortex-A53/A57/A72/A73,
- single CPU type, up to 4 cores/cluster
- ARMv8.2 support,
- Cortex-A55/75
- 2 CPU types, up to 8 cores but only up to 4b/cluster*

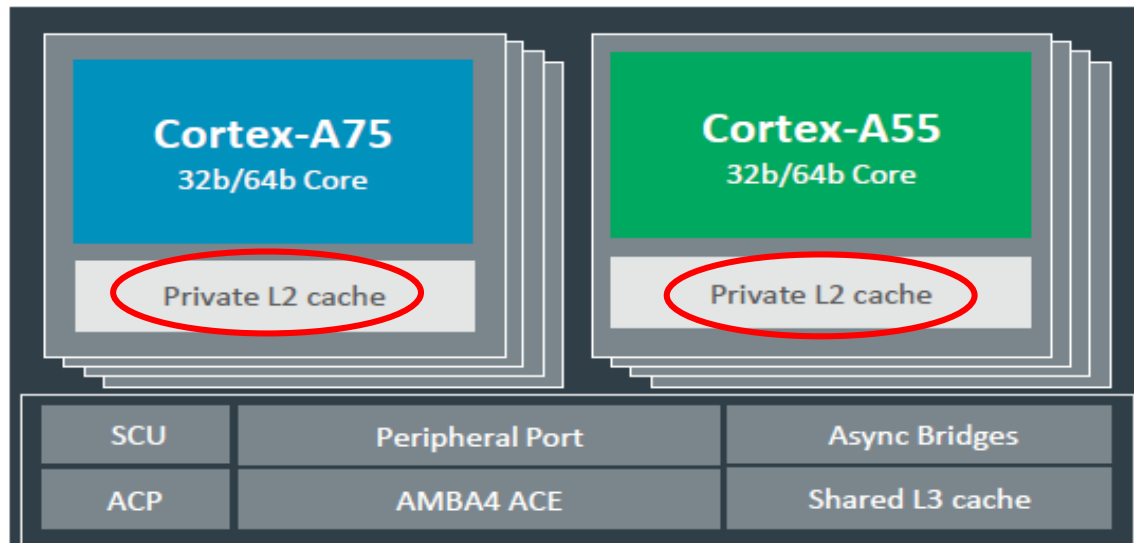
6.2 DynamIQ core clusters (8)

b) Private (per-core) L2 caches in the CPU cores

big.LITTLE core cluster (2015) [55]



DynamIQ core cluster (2017) [56]



1b+7L



2b+6L



4b+4L



1b+2L



1b+3L



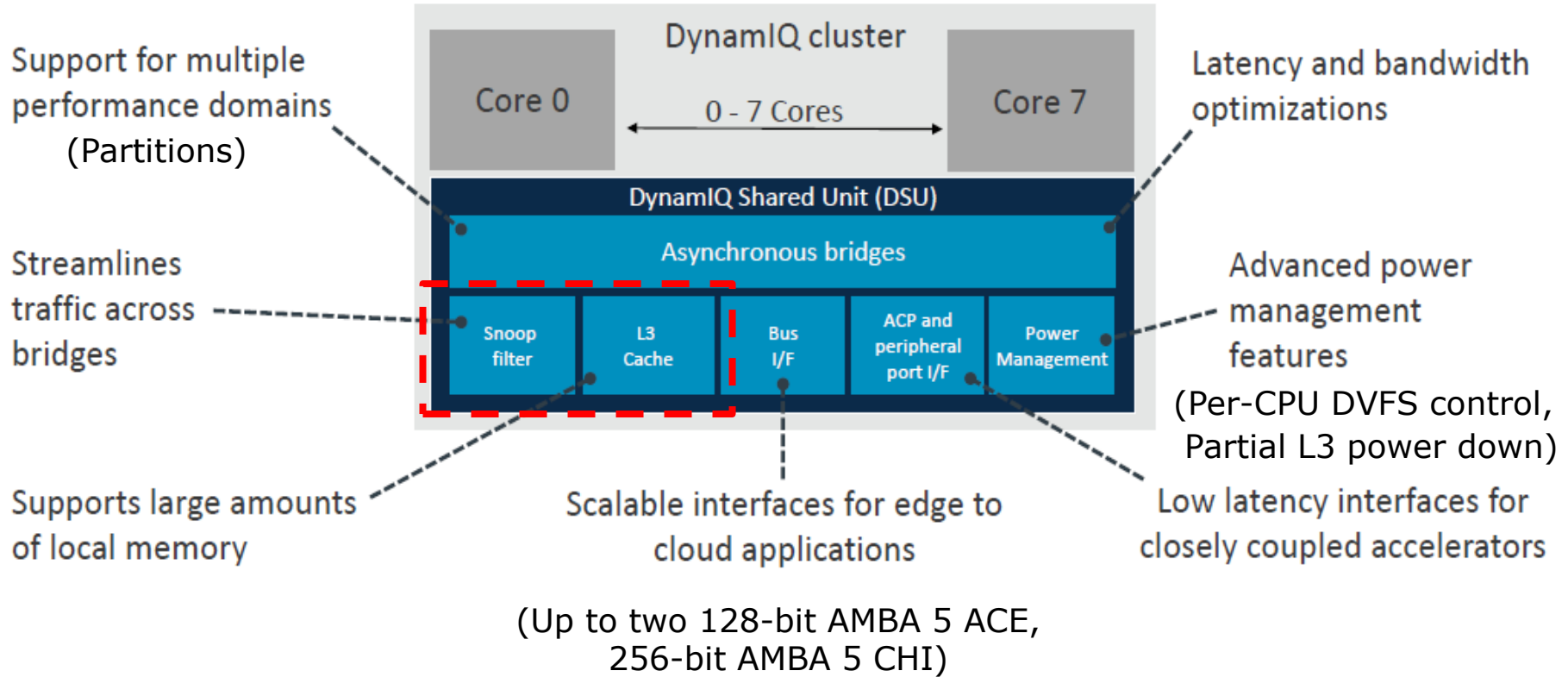
1b+4L

- ARMv8.0 support, e.g. Cortex-A53/A57/A72/A73, up to 4 cores/cluster
- Shared L2 cache

- ARMv8.2 support, (Cortex-A55/75)
- 2 CPU types, up to 8 cores but only 4b/cluster
- Private L2 caches, shared L3 cache

6.2 DynamIQ core clusters (9)

c) DynamIQ Shared Unit (DSU) with a shared L3 cache and snoop filter [106]



6.2 DynamIQ core clusters (10)

Reduced cache latencies [106]

Load to Use Cycles*	Cortex-A53	Cortex-A55	Cortex-A73	Cortex-A75
L1 hit	3	2	3	3
L2 hit	13	6	19	8
L3 hit	-	21	-	25

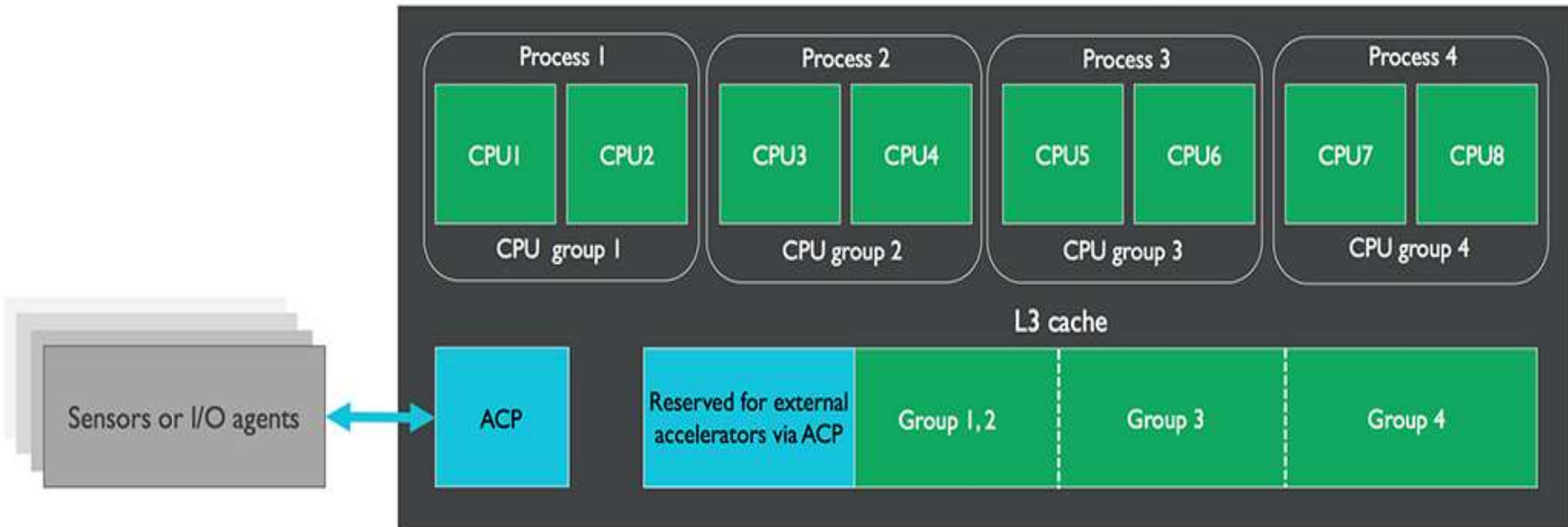
d) Capability for partitioning the cores and the L3 cache [98]

- It is feasible to set up **up to four partitions** by assigning the cores, the external accelerators attached to the DSU via the ACP and sections of the L3 cache into a maximum of 4 partitions.
- Partitioning may be unbalanced, e.g. 1 CPU could be assigned 2 GB whereas the other CPUs the remaining 2 GB (assuming an 8 GB configuration).
- The partitions are dynamic and can be created/adjusted during runtime by the OS or hypervisor.
- Partitioning can be useful for embedded systems that run a fixed workload or applications that require a more deterministic runtime.

6.2 DynamIQ core clusters (12)

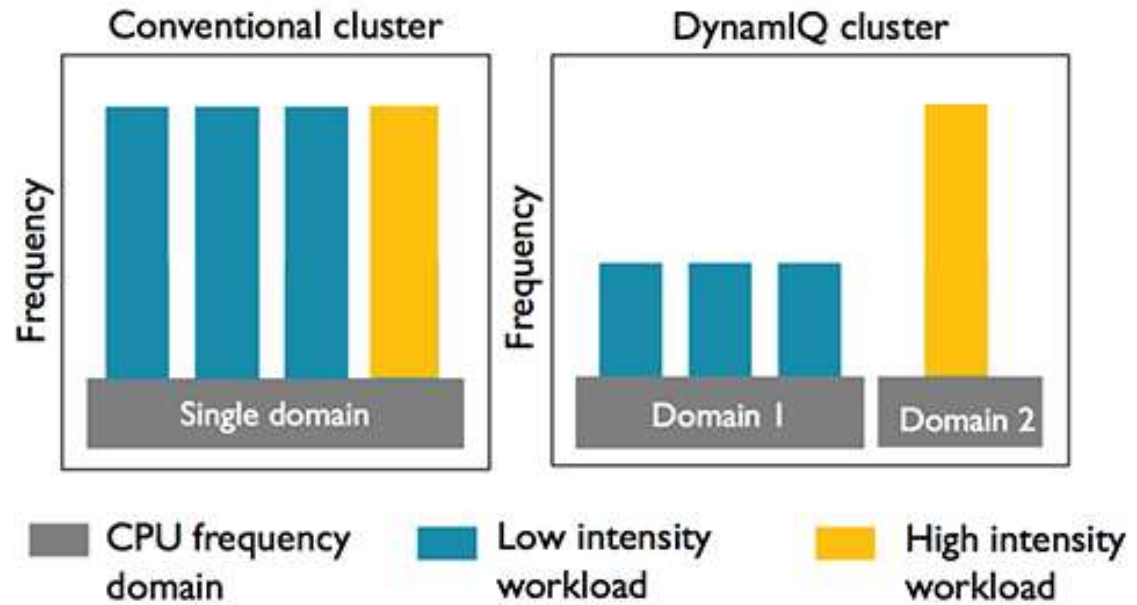
Example for partitioning the cores and the L3 [98]

Example configuration with four groups of CPUs in a DynamIQ cluster



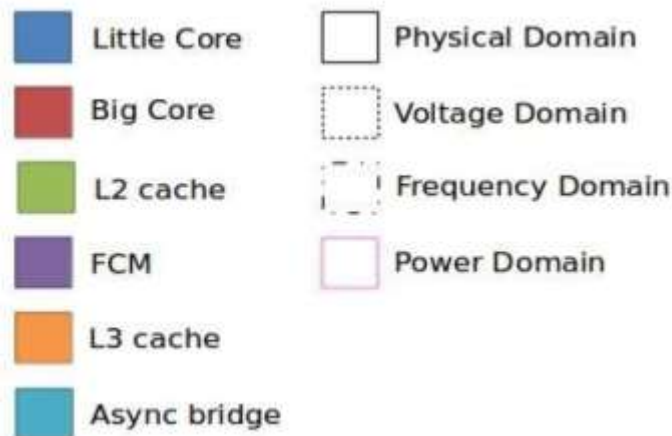
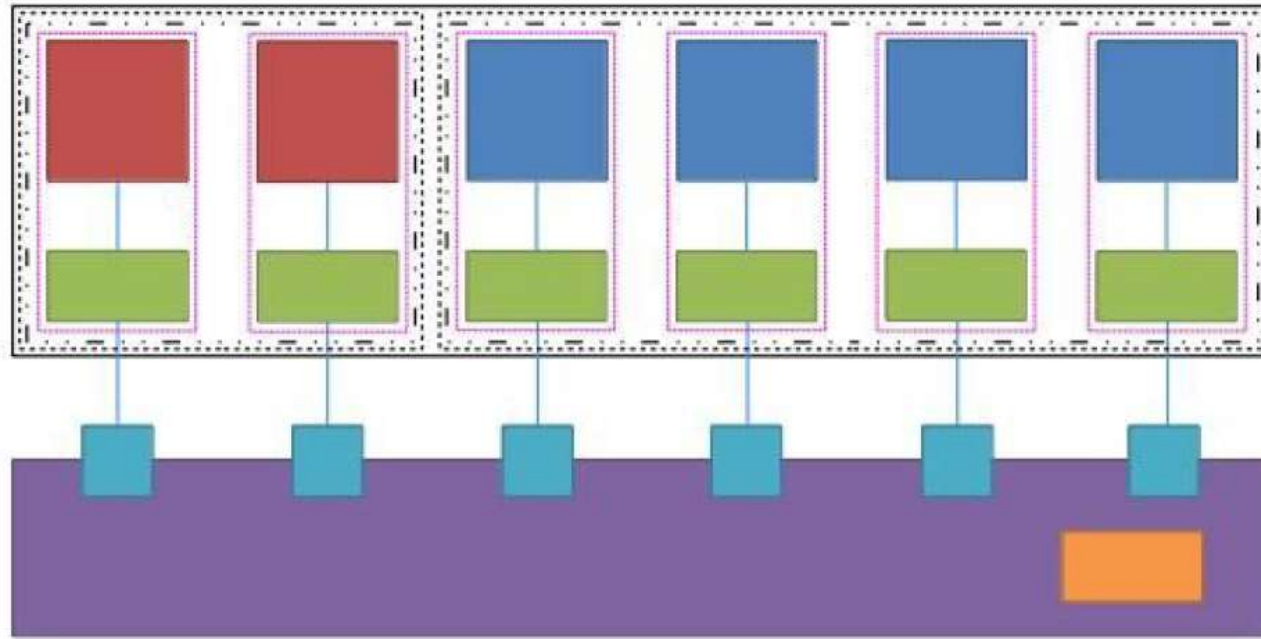
e) Finer-grain frequency and voltage control [98]

Groups of CPUs (i.e. partitions) can run at different frequencies and voltages.



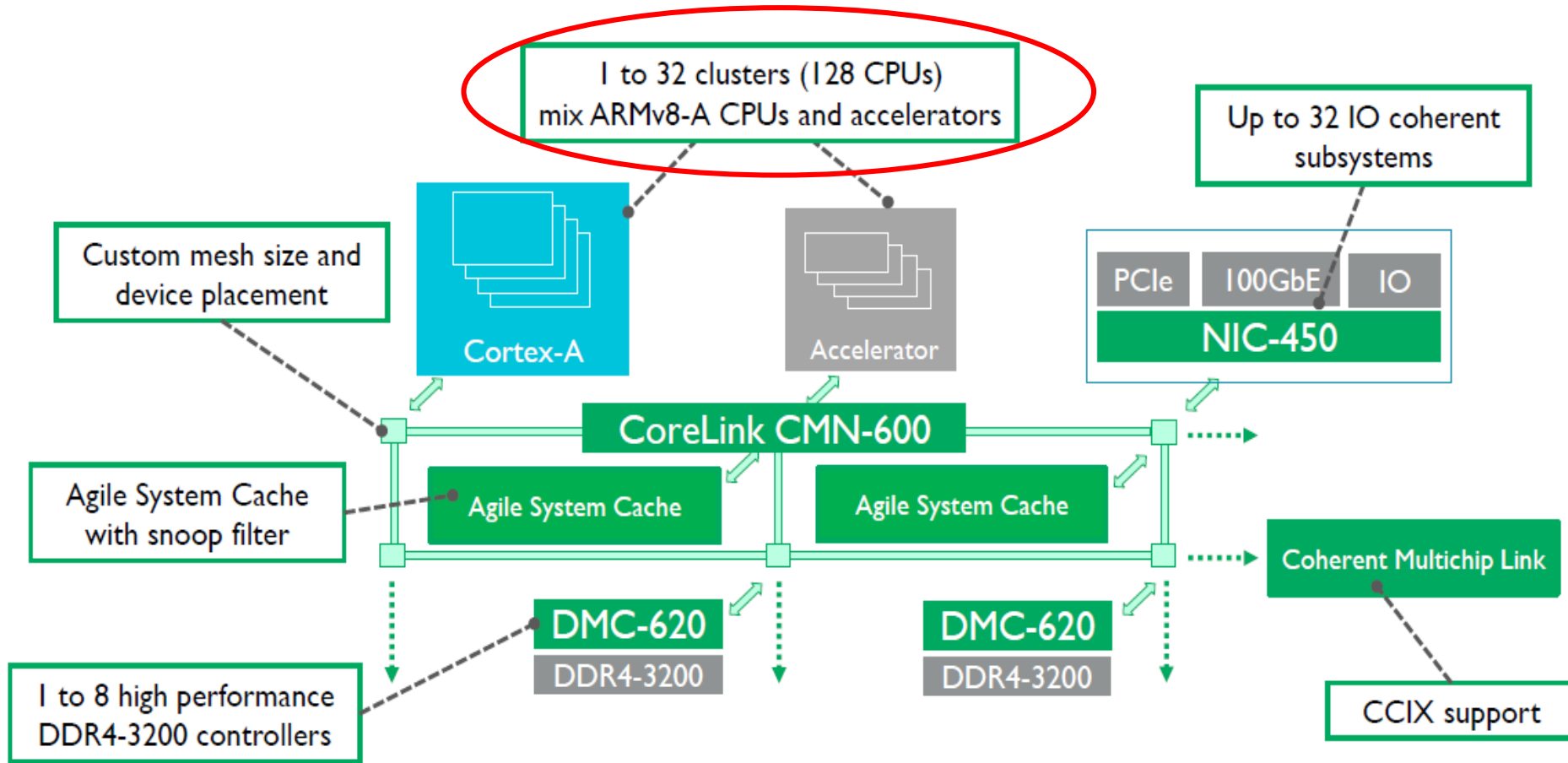
6.2 DynamIQ core clusters (16)

Example for two voltage and frequency domains [104]



6.2 DynamIQ core clusters (17)

f) Mesh interconnect (CMN-600) for server systems [105]



6.2 DynamIQ core clusters (18)

The reason for changing the interconnect topology for servers [105]

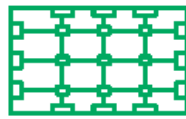
Ring
interconnect

CoreLink CCN

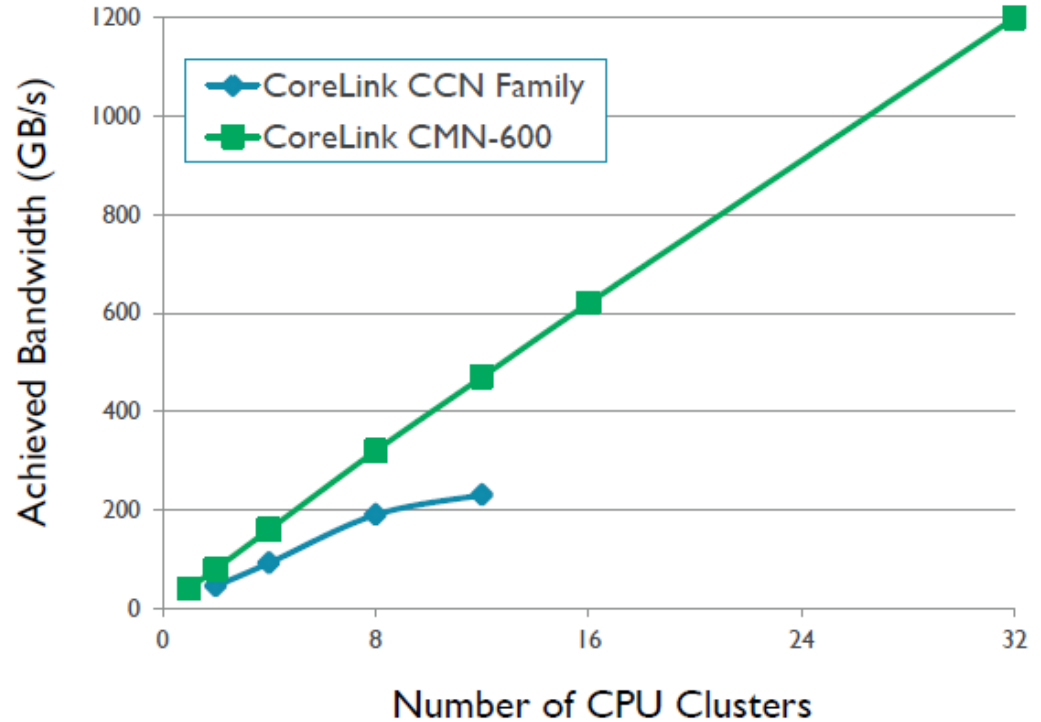


Mesh
interconnect

CoreLink CMN



Bandwidth scaling comparison

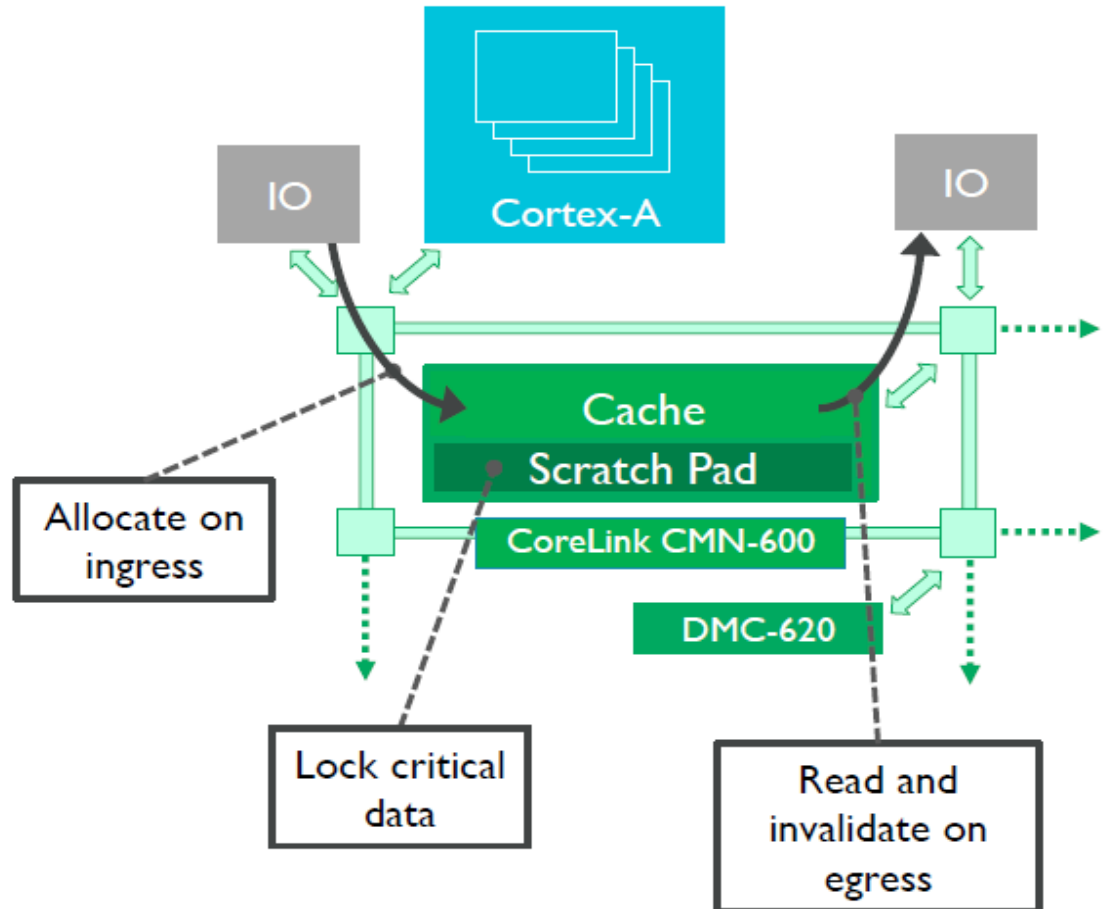


- Achieved coherent bandwidth as observed by requestors
- Same process node and test conditions

6.2 DynamIQ core clusters (14)

g) Use of a scratch pad system cache to increase throughput [105]

- It is termed also as **Agile cache**.
- It includes a snoop filter.

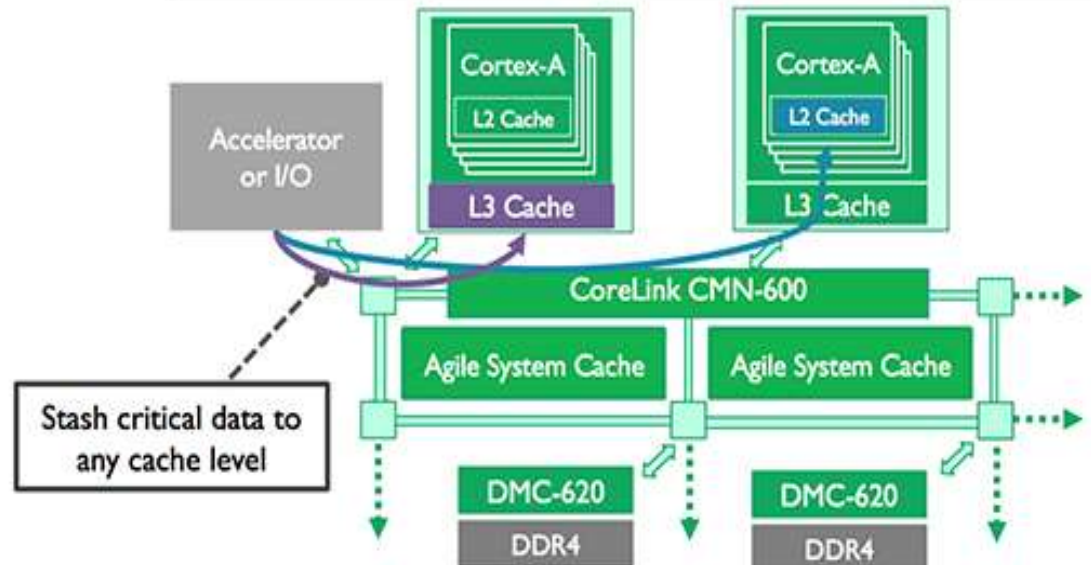
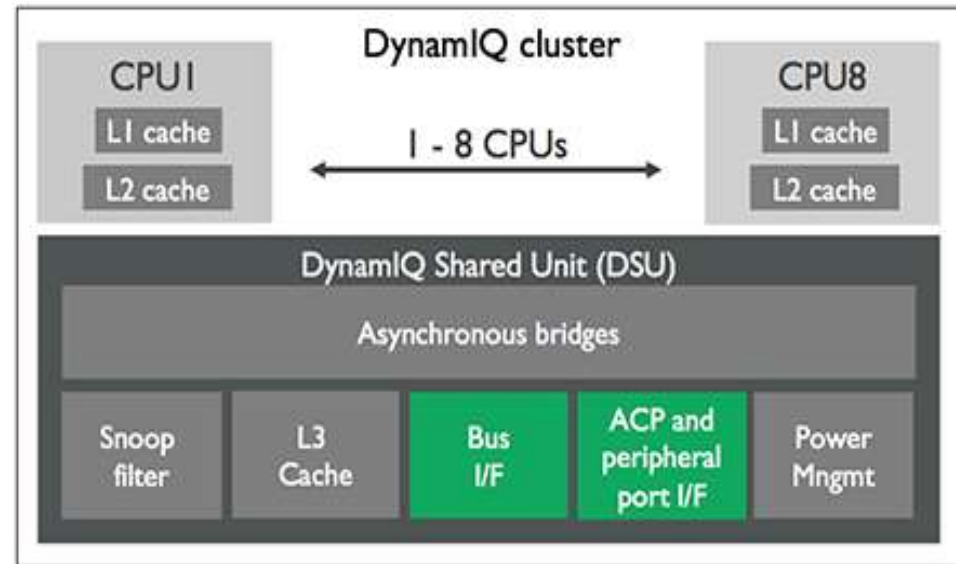


- Ingress traffic: inbound network traffic that originates from outside of the network's routers and proceeds toward a destination inside of the network.
- Egress traffic: outbound network traffic that originates inside of the network and proceeds to another network.

6.2 DynamIQ core clusters (13)

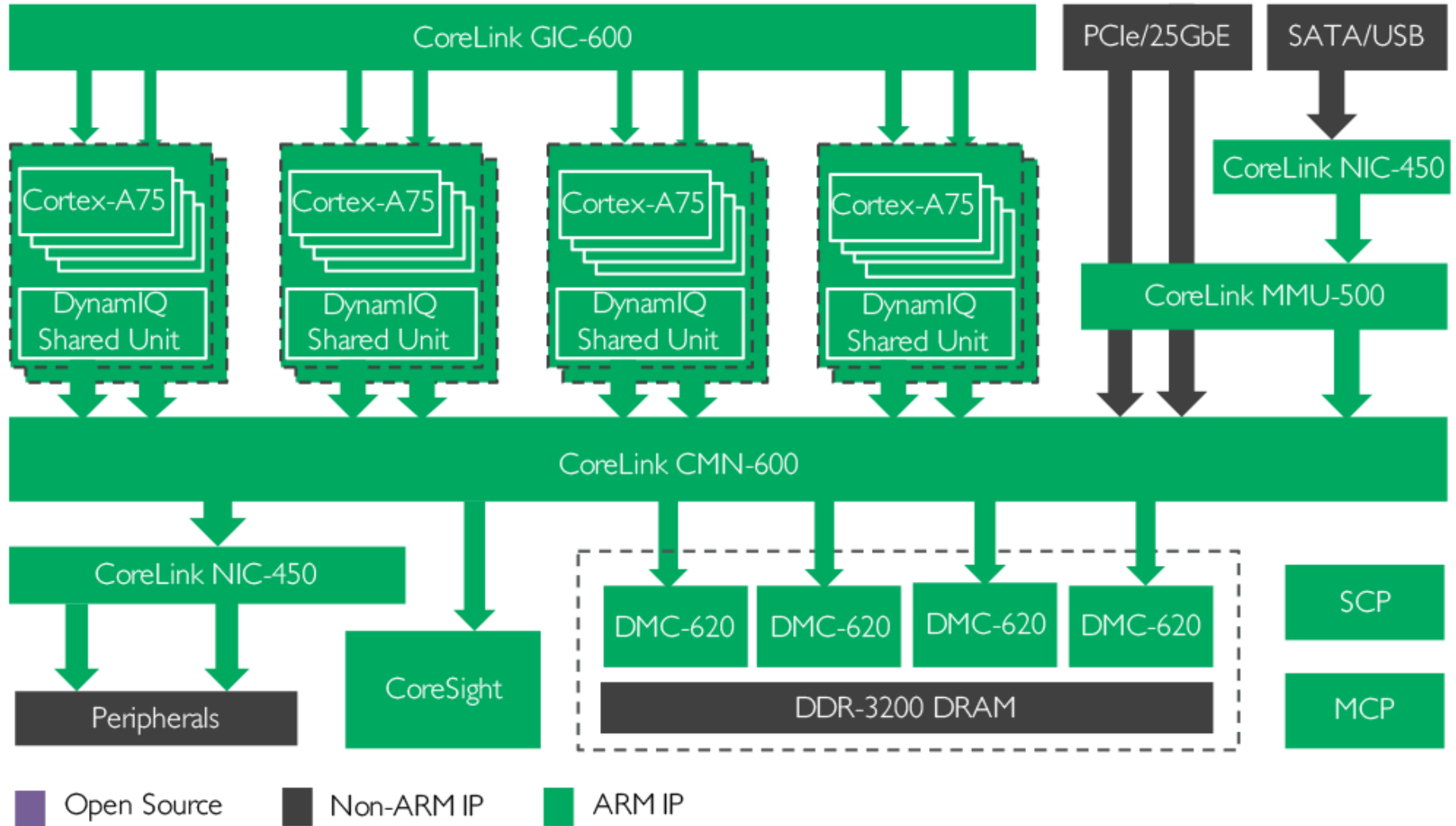
h) Cache stashing [98]

- It enables reads/writes into the shared L3 cache or per-core L2 cache.
- It allows closely coupled accelerators and I/O agents to access the CPU memory via the AMBA 5 CHI (PP) or ACP port.
- Cache stashing increases throughput.



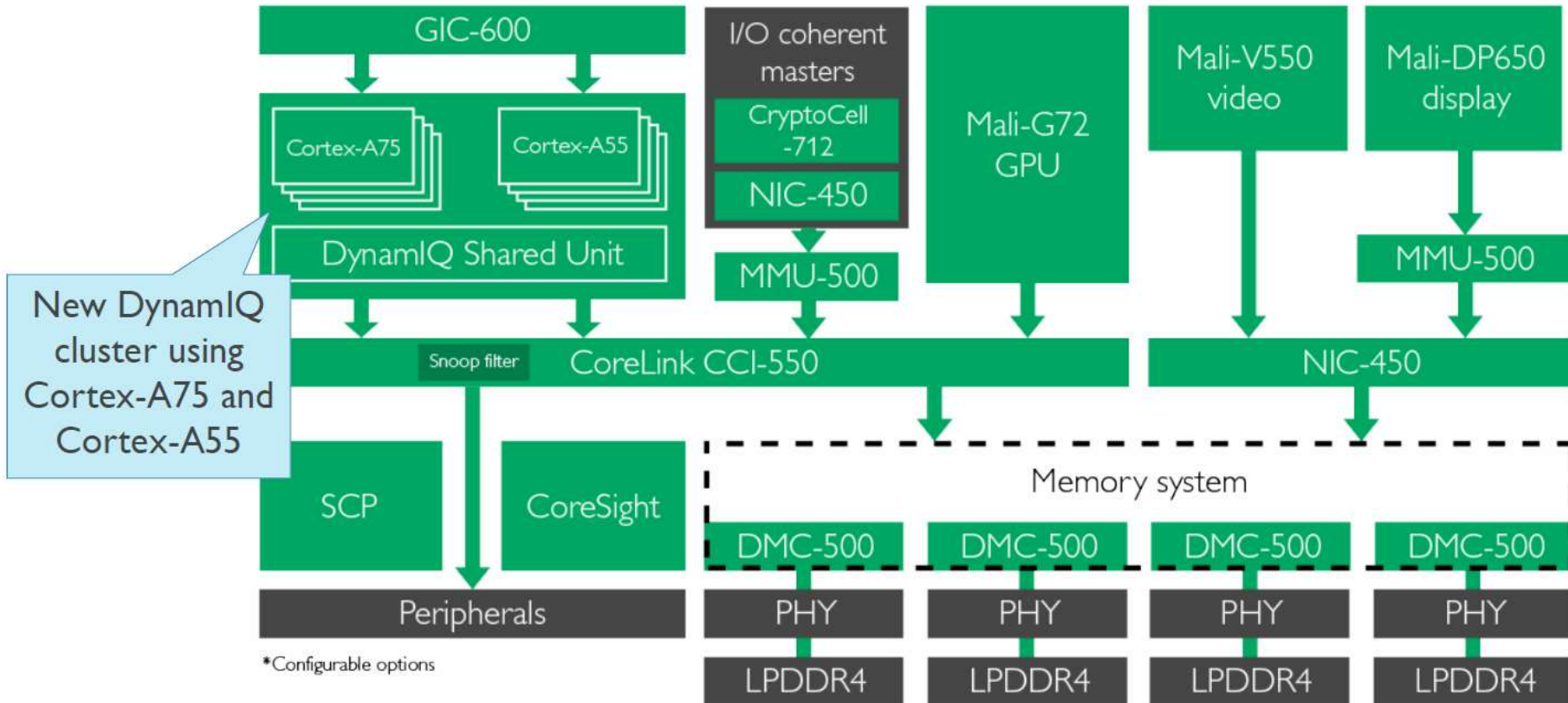
6.2 DynamIQ core clusters (19)

Use of the mesh interconnect (CMN-600) to build a server [112]



6.2 DynamIQ core clusters (20)

Use of a previous ring interconnect (CCI-550) to build premium mobile systems for 2018 [108]



CCI-550: Cache Coherent interconnect

MMU-500: Memory Management Unit (responsible for virtualization and caching)

NIC-450: Network Interconnect (interface converter)

GIC-600: General Interrupt Controller

SCP: System Control Processor

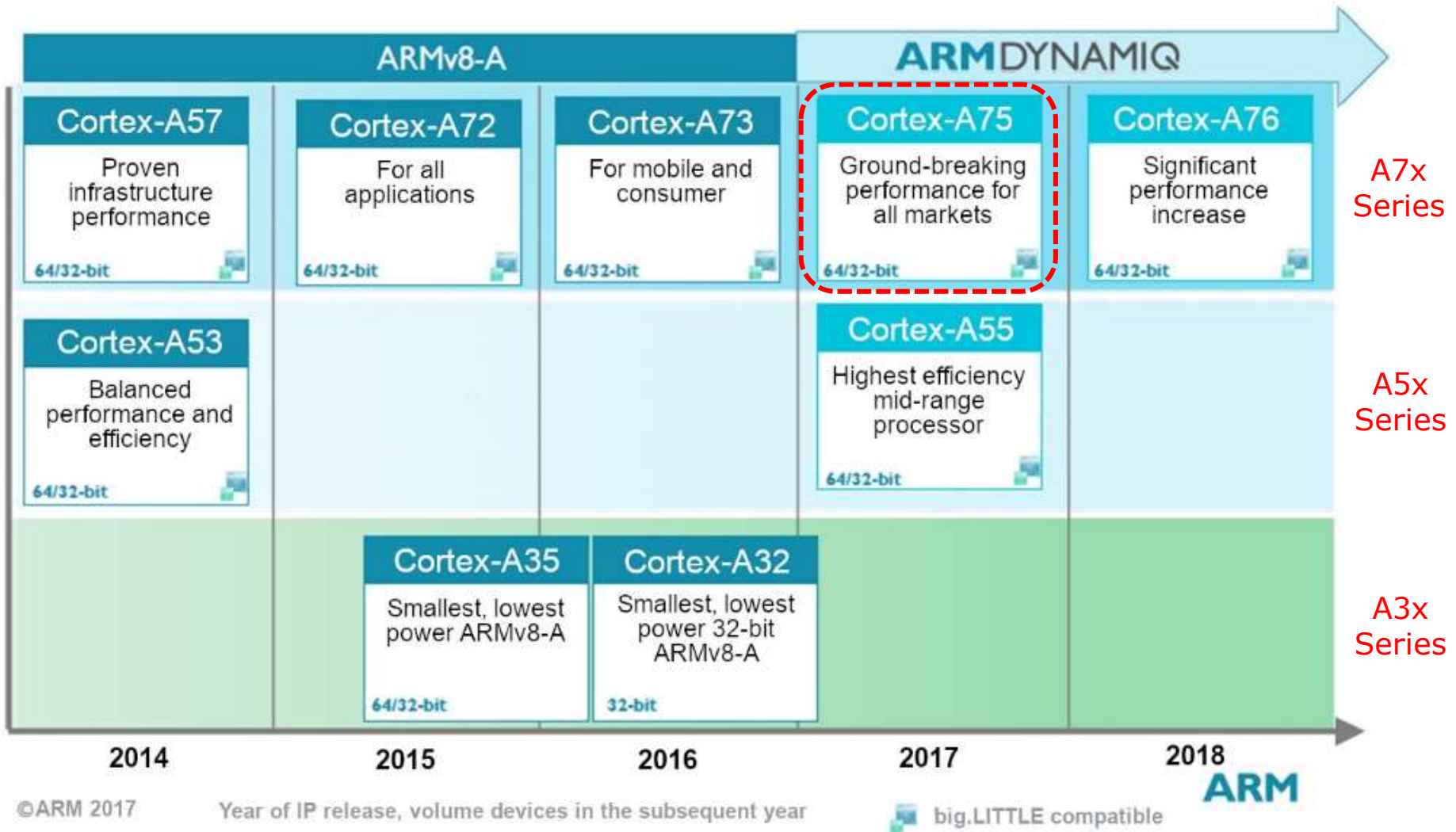
*

6.3 The high performance Cortex-A75

(It won't be discussed)

6.3 The high performance Cortex-A75 (1)

6.3 The high performance Cortex-A75 -1

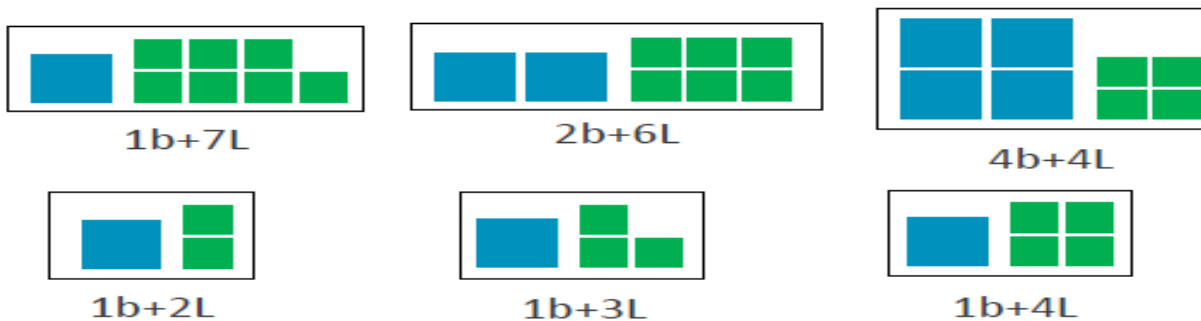
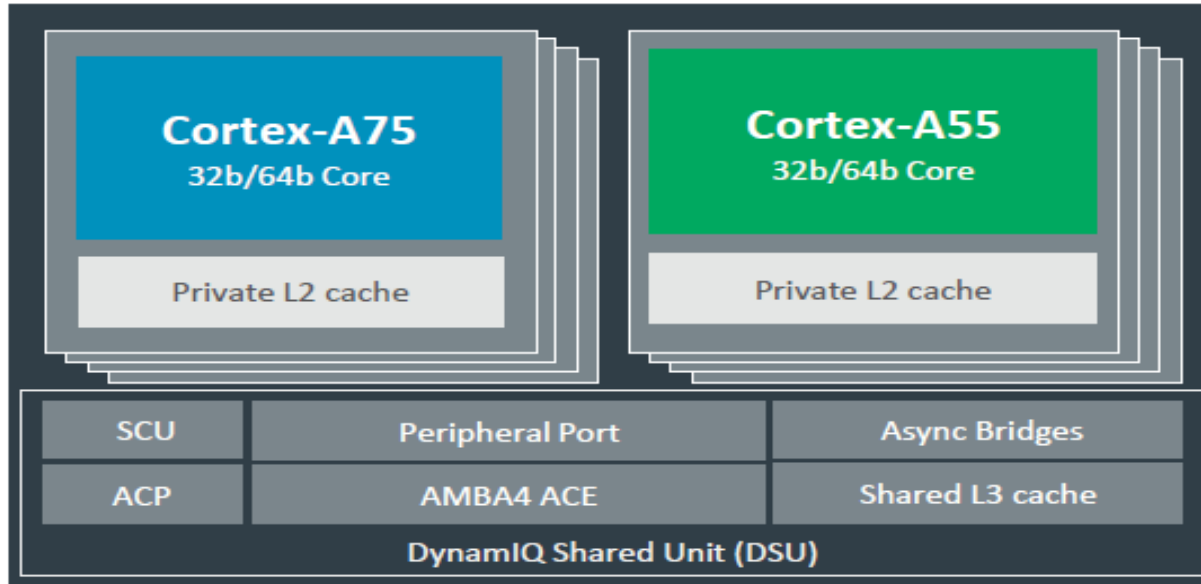


The high performance Cortex-A75 -2

- 05/2017: **Announced** along with the high-efficiency midrange Cortex-A55 and the MALI 72 GPU.
- Target process technology: **10 nm**.

6.3 The high performance Cortex-A75 (3)

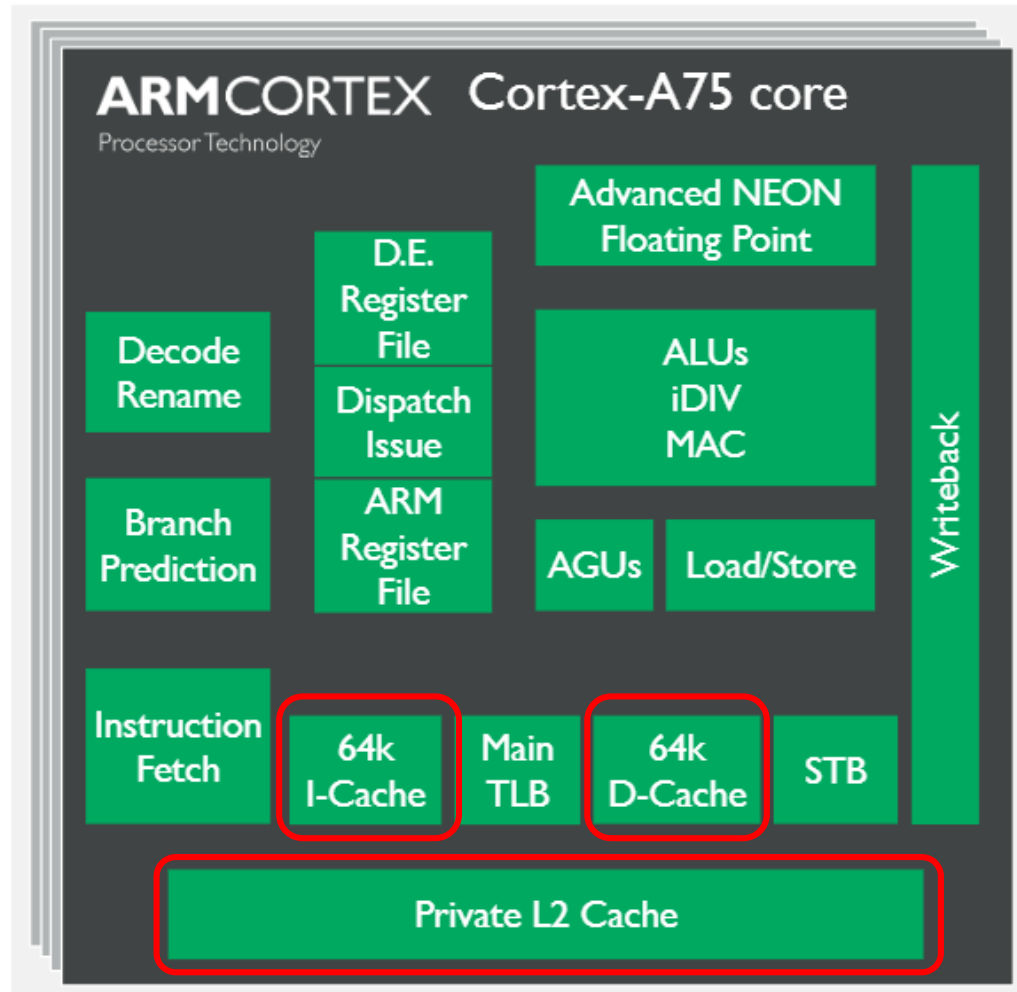
The Cortex-A75 as the building block of ARM's DynamIQ core clusters [103]



- Implements the ARMv8.2 ISA
- Up to 8 cores (up to 4 Cortex-A75, up to 8 Cortex-A55)

6.3 The high performance Cortex-A75 (4)

Block diagram of the Cortex-A75 [110]

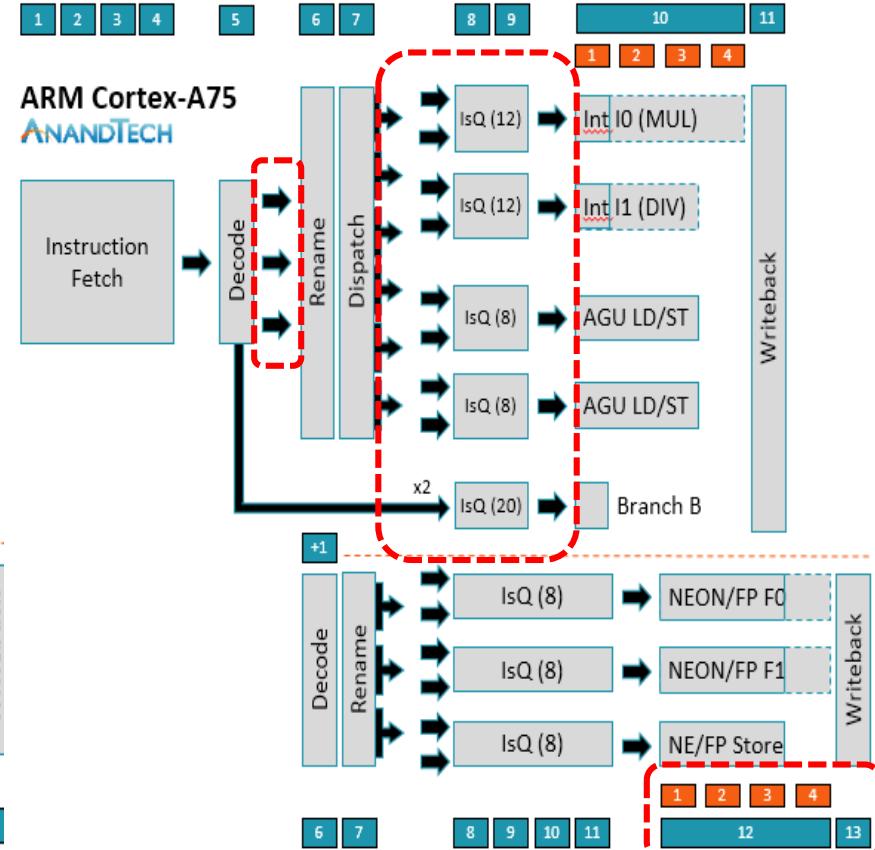
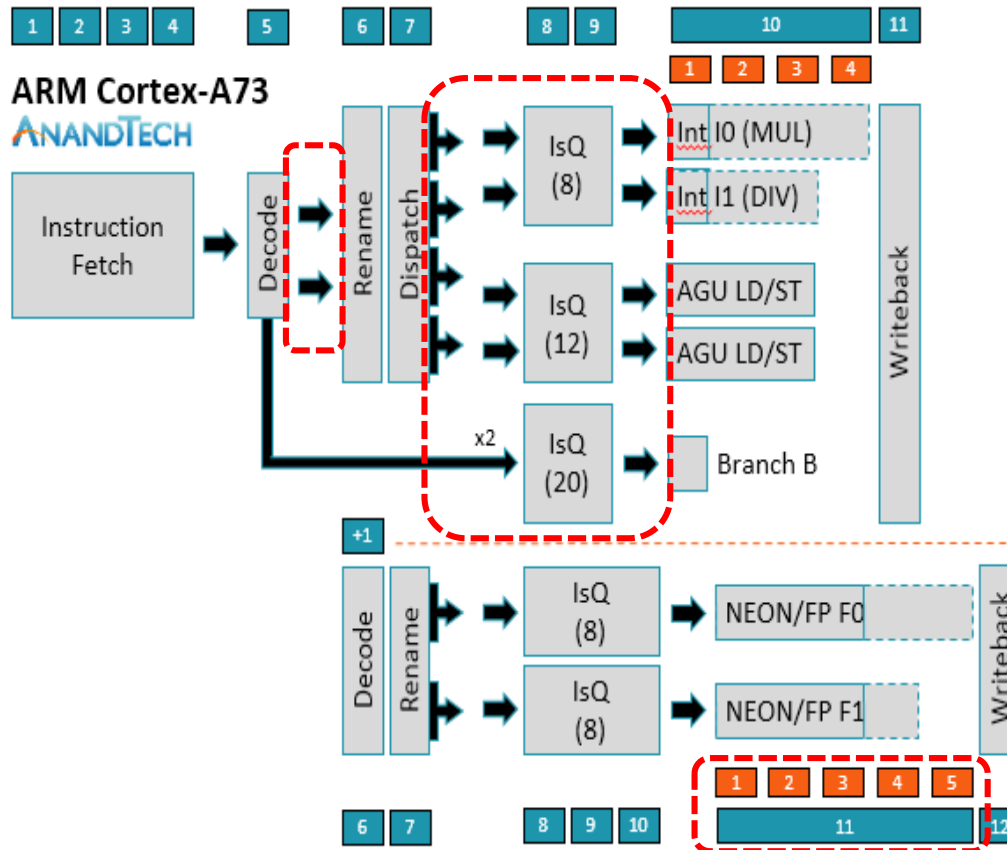


D.E.: Default and Exception

STB: Store Buffer

6.3 The high performance Cortex-A75 (5)

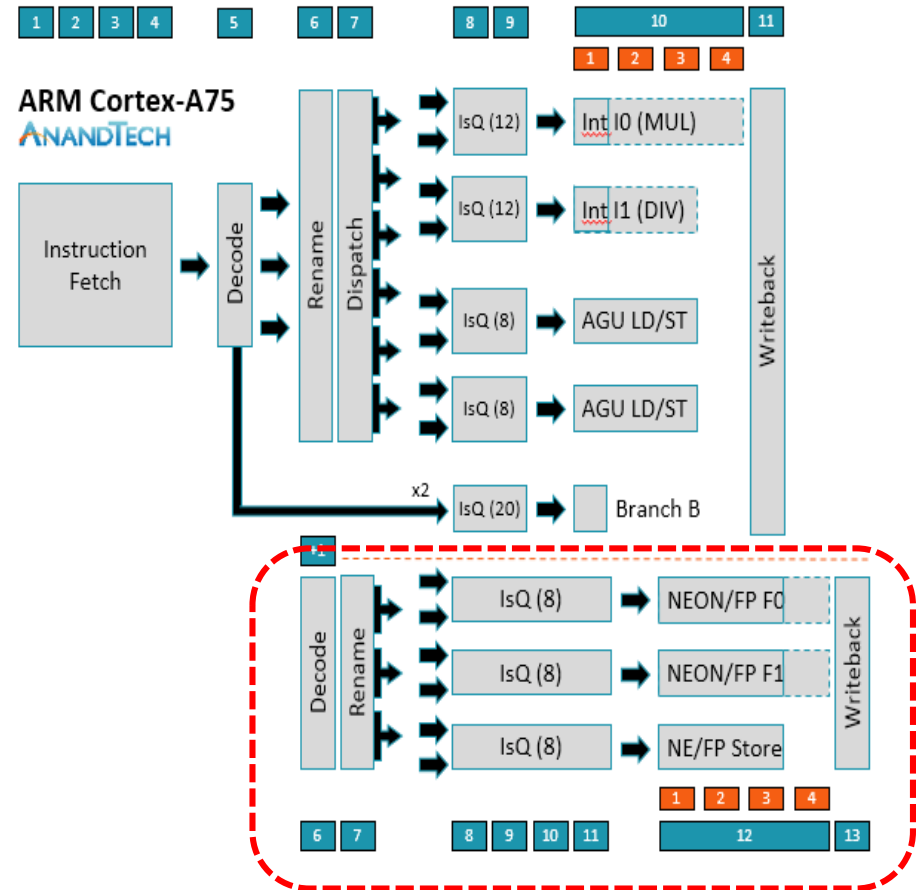
Enhancements of the Cortex-A75 pipeline vs. the Cortex-A73 pipeline [98]



6.3 The high performance Cortex-A75 (6)

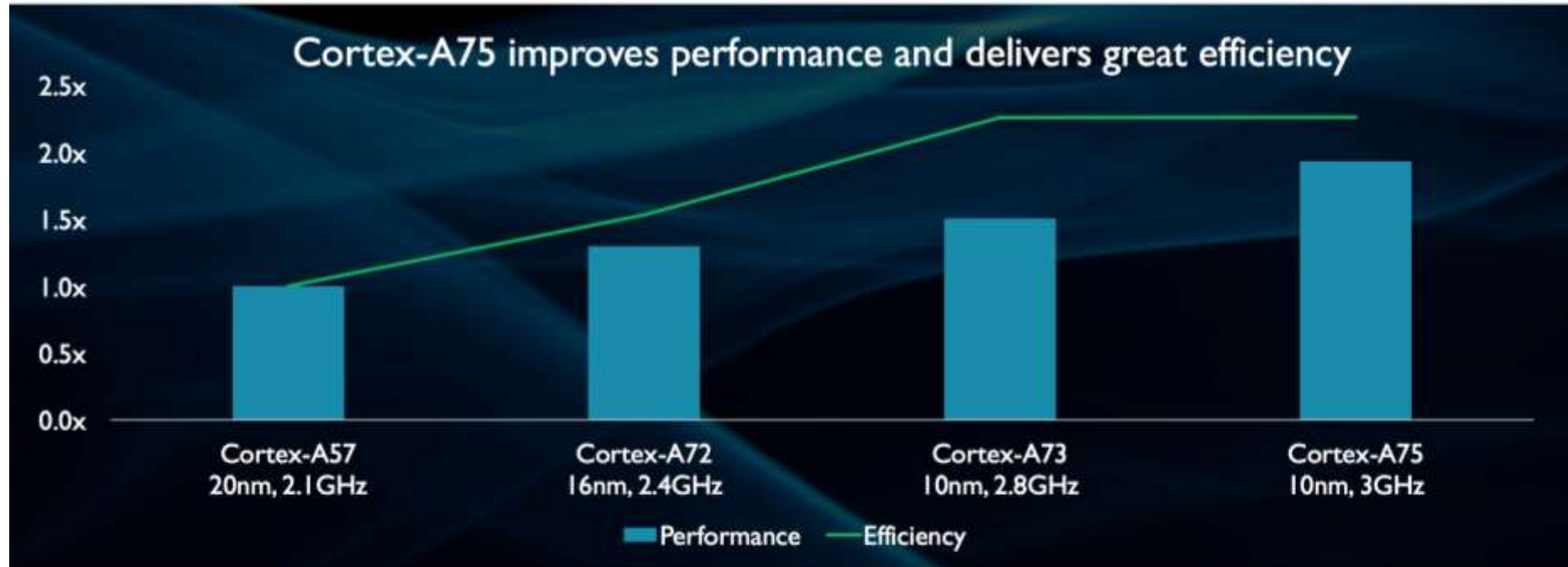
Innovations introduced by the NEON/FP pipelines [121], [98]

- Dedicated renaming engine for NEON/FPU
- Support for FPI6 half-precision processing
 - Double throughput compared to single precision
 - Significant performance uplift for image processing
- Support for Int8 dot product
 - Increased performance on neural network algorithms
- Enhanced floating-point MAC throughput
- Dedicated data store queue



6.3 The high performance Cortex-A75 (7)

Per-core performance and efficiency of the Cortex-A75 [98]



Performance and efficiency per-core, at-speed using target process node

6.3 The high performance Cortex-A75 (8)

Contrasting the performance of the Cortex-A75 vs. the Cortex-A73 [106]

Cortex-A75



Baseline to Cortex-A73

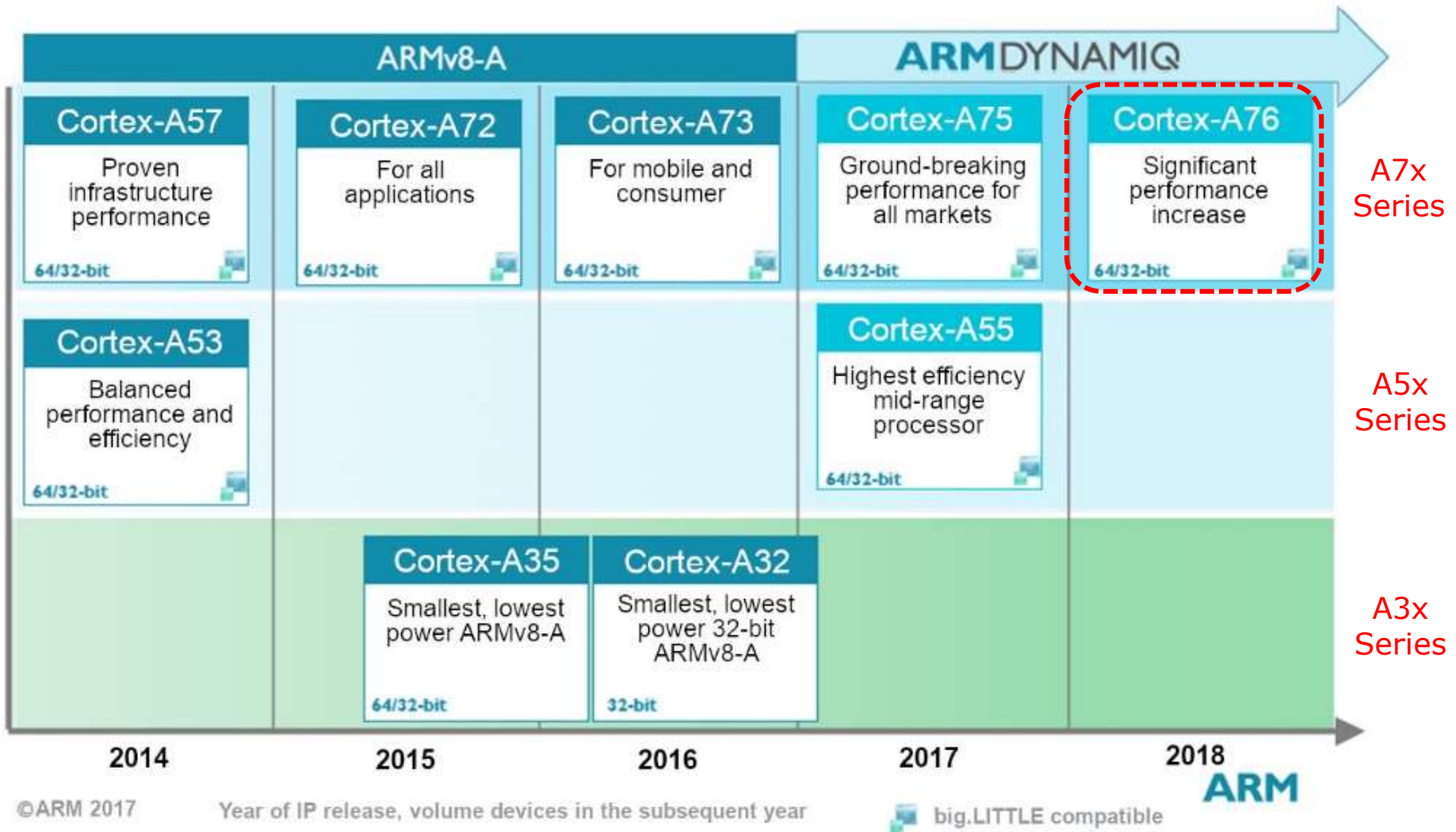
*All comparisons at ISO
process and frequency*



6.4 The high performance Cortex-A76

6.3 The high performance Cortex-A75 (1)

6.4 The high performance Cortex-A76 -1



Source: AMD

The high performance Cortex-A76 -2

Main design goals

- Laptop-class performance with mobile efficiency while outperforming the competition at half by silicon area and power consumption.

To achieve this

- remove bottlenecks throughout the design to improve performance
- optimize every microarchitectural feature to extract maximal performance at minimal power and area
- brand new, wider microarchitecture vs. the preceding ones
- Enhanced, 2. gen. DynamIQ core cluster.
- Targeting 12, 7 or 5 nm process technology.
- The Cortex-A76 can be combined with the Cortex-A55 CPU.

6.4 The high performance Cortex-A76 (2)

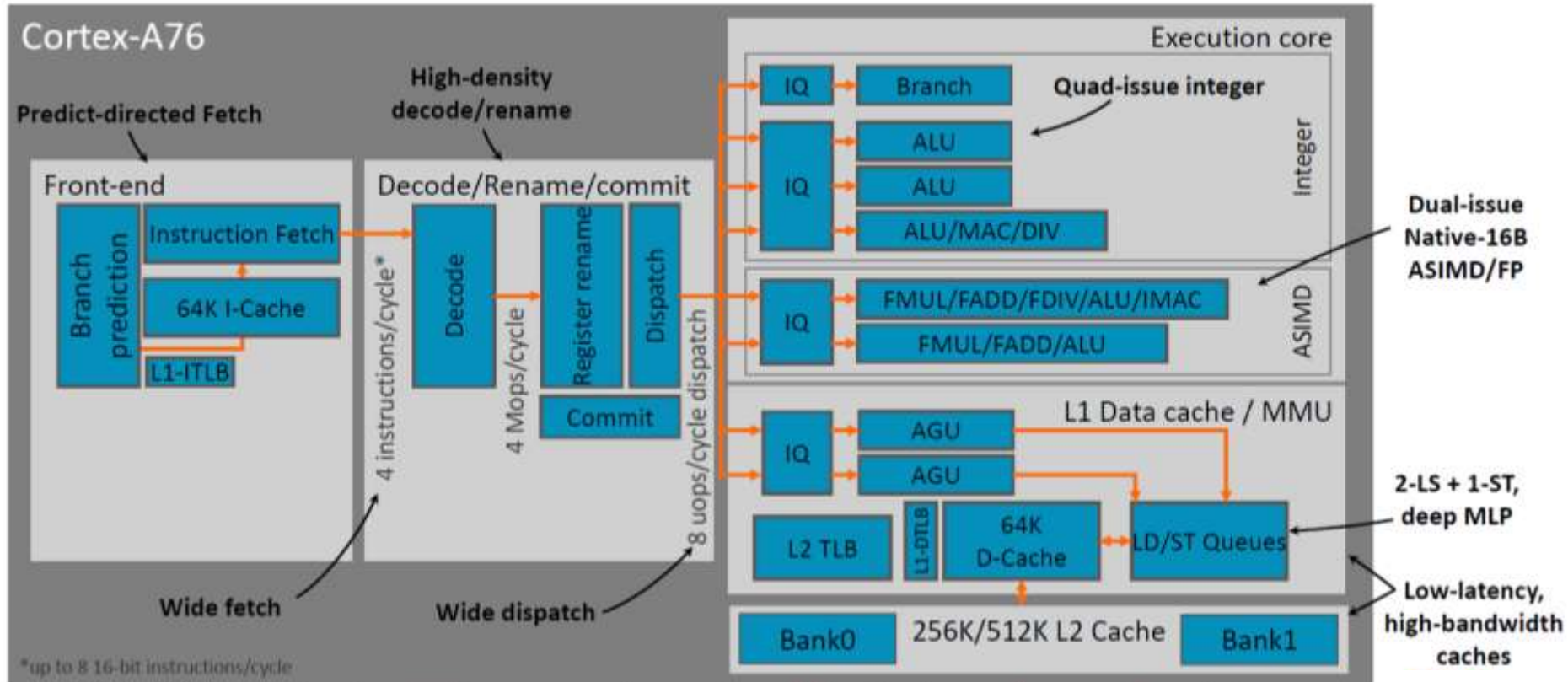
Performance and power efficiency improvements of the Cortex-A76 [119]



Figure: Performance and power efficiency of a 3.0 GHz 7 nm Cortex-A76 vs. a 2.8 GHz 10 nm Cortex-A75 []

6.4 The high performance Cortex-A76 (3)

Overview of the microarchitecture of the Cortex-A76 [119]



MLP: Memory Level Parallelism

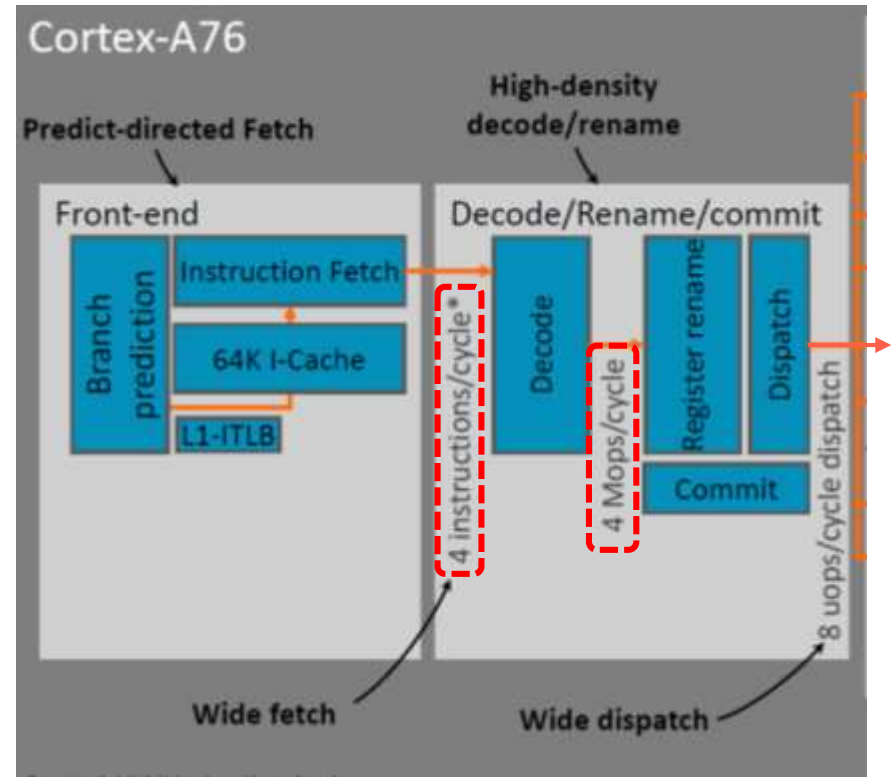
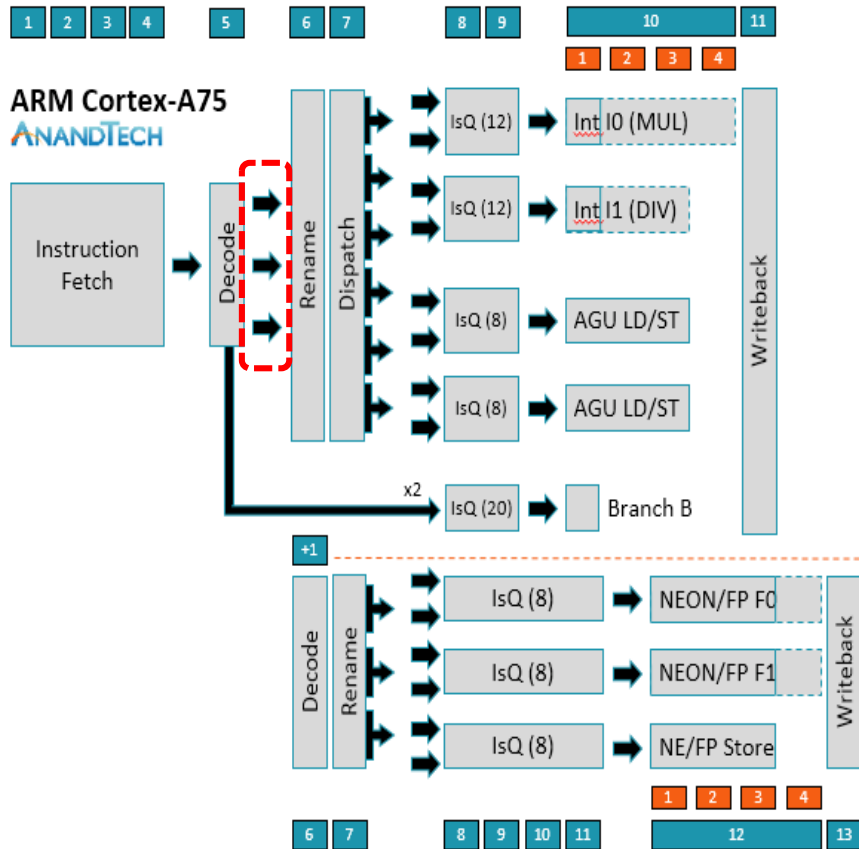
ASIMD (Advanced SIMD): FMAD etc. operations are executed interleaved (called with late forwarding) i.e. FMAD operations will already be started when the multiply operands are available and the accumulate operand will be added later after the multiply operation has already been calculated in order to speed up the execution of a chain of FMAD etc. instructions.

Key enhancements of the microarchitecture

- 4-wide front-end, up from 3 of the previous Cortex-A75
- dispatch rate of 8, up from 6 in the previous Cortex-A75
- larger issue buffers

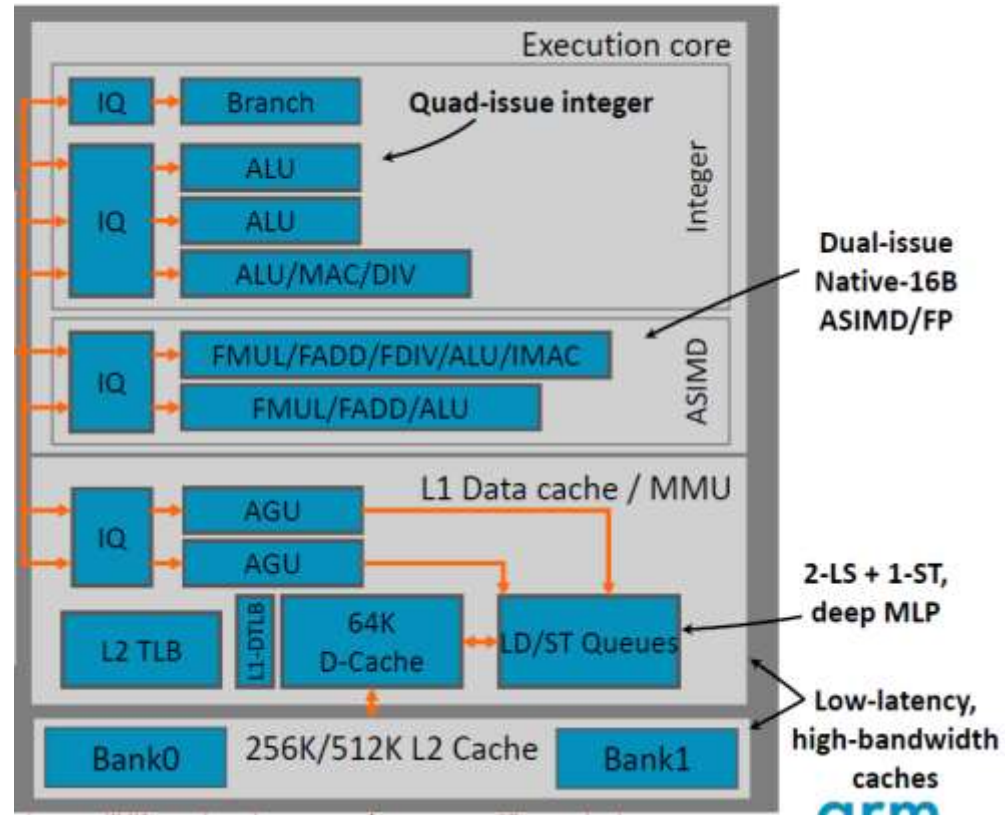
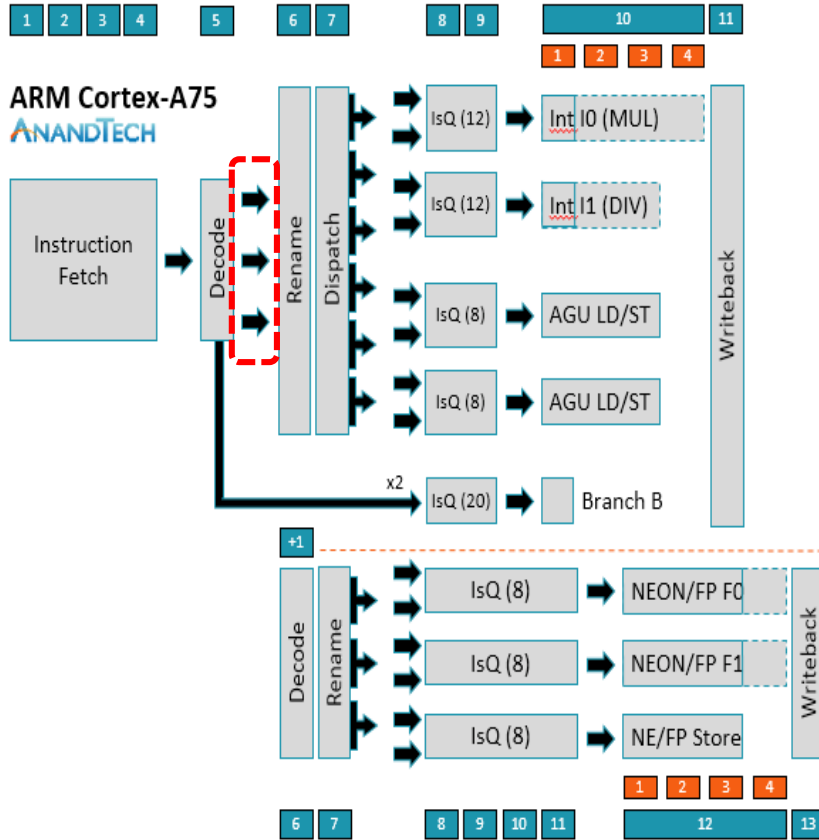
6.4 The high performance Cortex-A76 (5)

4-wide front-end (up from 3 in the previous Cortex-A75) [119]



6.4 The high performance Cortex-A76 (6)

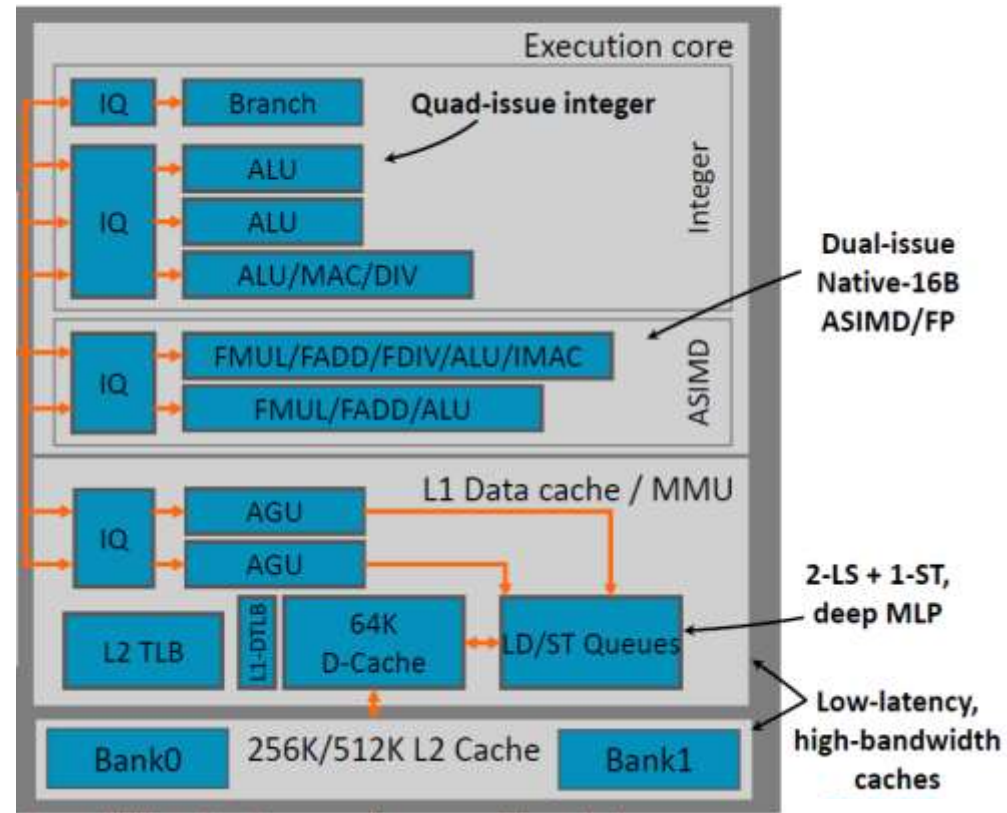
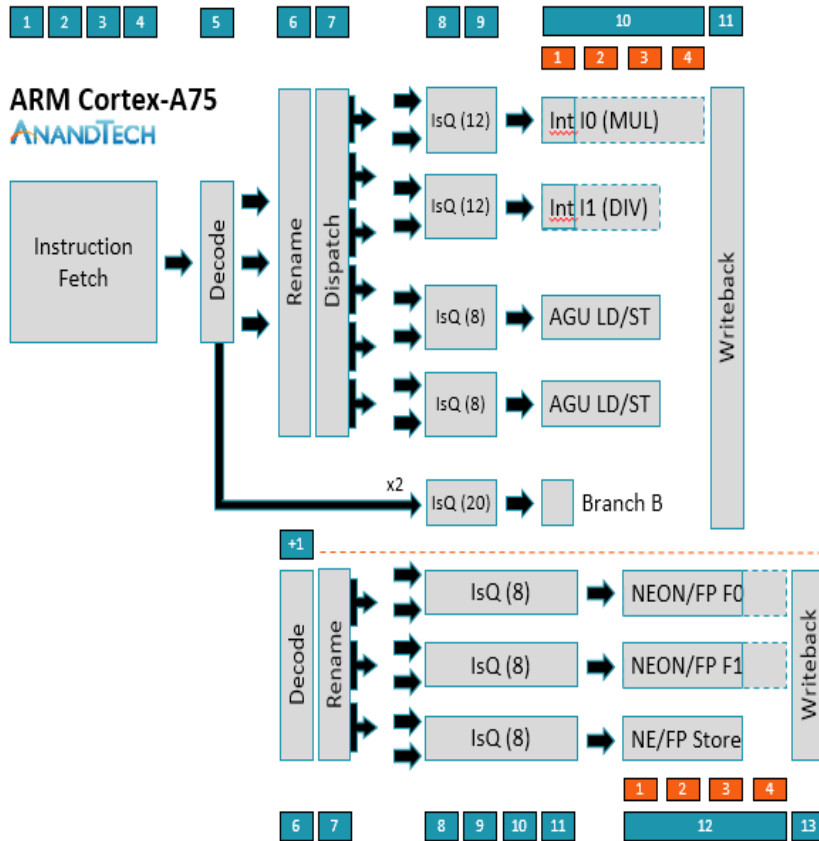
Dispatch rate of 8 (up from 6 in the previous Cortex-A75) to 8) [119]



MLP: Memory Level Parallelism

6.4 The high performance Cortex-A76 (7)

Larger issue buffers (IsQ/IQ) [119]



MLP: Memory Level Parallelism

Note that the issue queues of the Cortex-A76 typically have 16 entries rather than 12 seen in the Cortex-A75.

6.4 The high performance Cortex-A76 (8)

The cache hierarchy of the Cortex-A76 processor [119]

Full cache hierarchy is co-optimized for latency and bandwidth

- No-compromise, get the best of both worlds

64K I-Cache, 64K D-Cache with 4-cycle LD-use

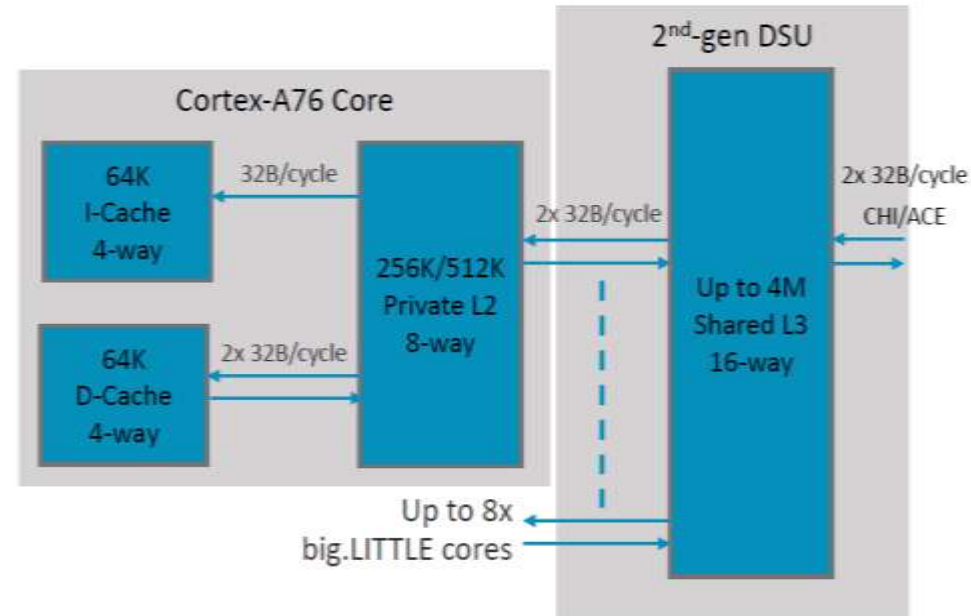
256KB-512KB private L2 with 9-cycle LD-use

- Adapts to system latency/BW characteristics

- Up to 46 outstanding misses

2M-4M DynamIQ L3 with 26-31 cycle LD-use

- 94 outstanding misses with flexible prefetch placement

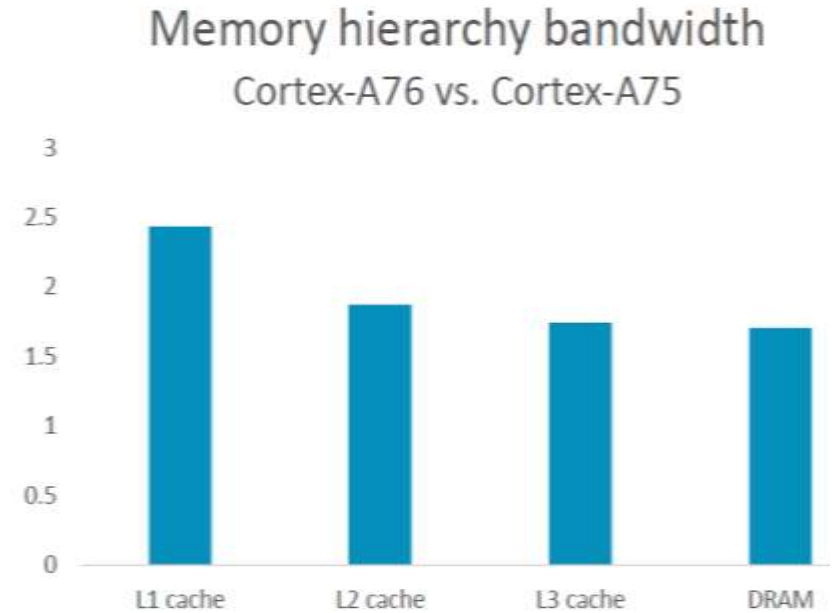


6.4 The high performance Cortex-A76 (9)

Bandwidth improvements of A76's caches vs. the A75 caches [119]

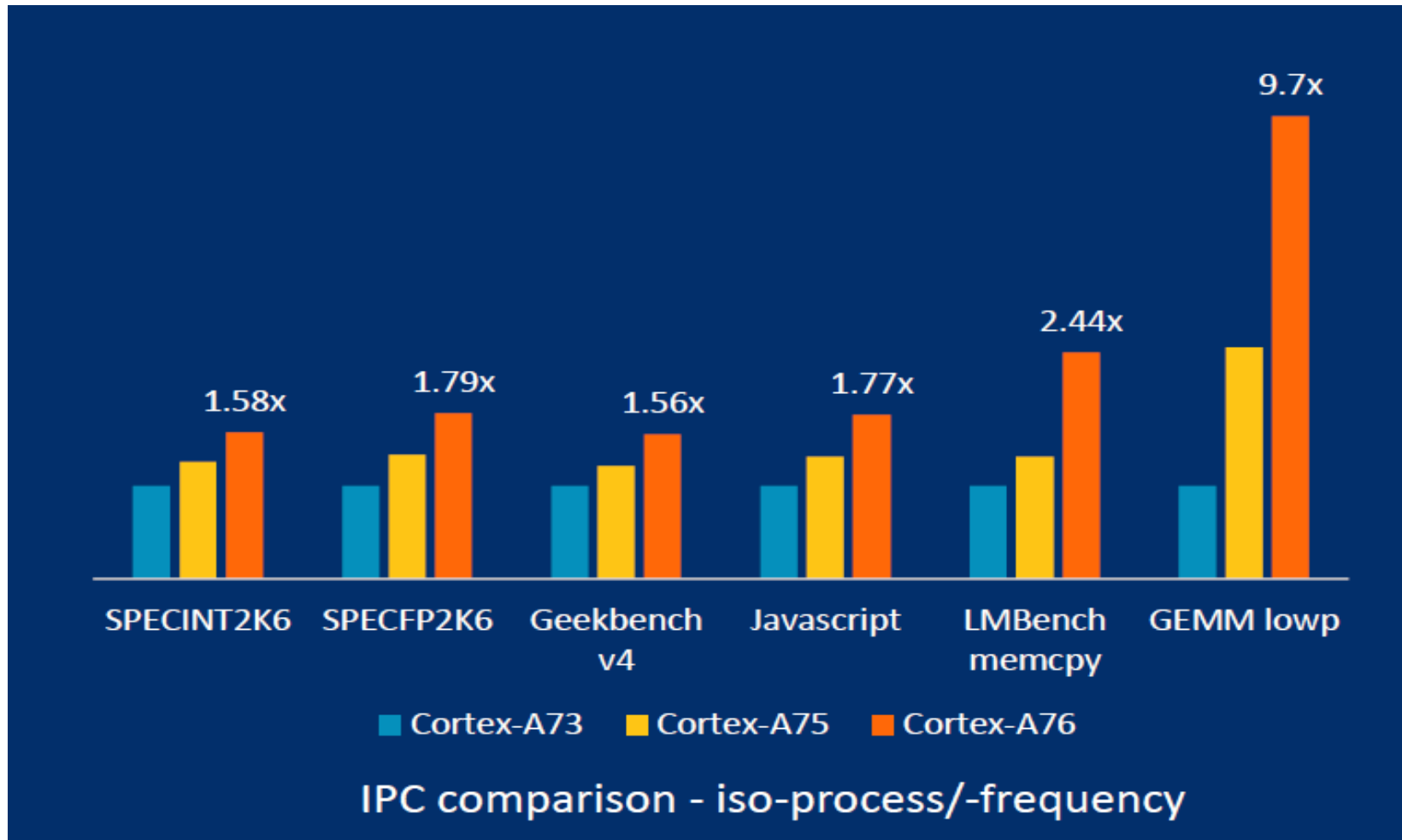
Increased bandwidth at low latency

- More than twice the performance of Cortex-A75
- True next-generation cache-hierarchy performance



6.4 The high performance Cortex-A76 (10)

Performance improvements in subsequent high-performance Cortex-A processors [119]



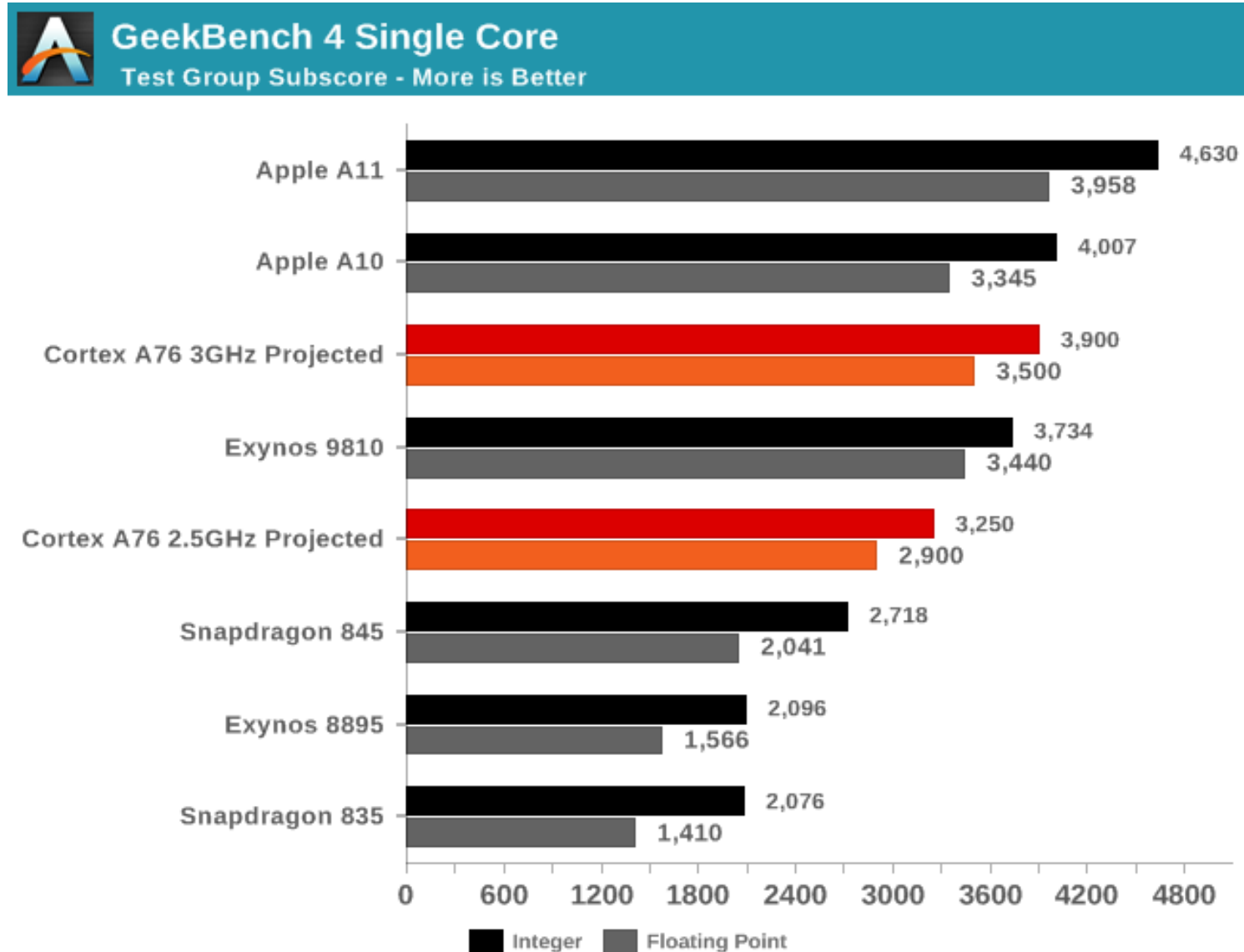
GEMM lowp is a library for multiplying matrices of 8-bit integers (used in NN apps.)

Performance improvements claims over the Cortex-A75 processors [119]

- **Single-thread performance improvements**
 - +25% more integer IPC than the Cortex-A75 CPU
 - +35 % higher ASIMD/FP performance
- **Mobile usage performance improvements**
 - +28% more Geekbench performance
 - +35 % more Javascript performance
- **Performance improvements while using AI applications**
 - 3.9x higher AI performance

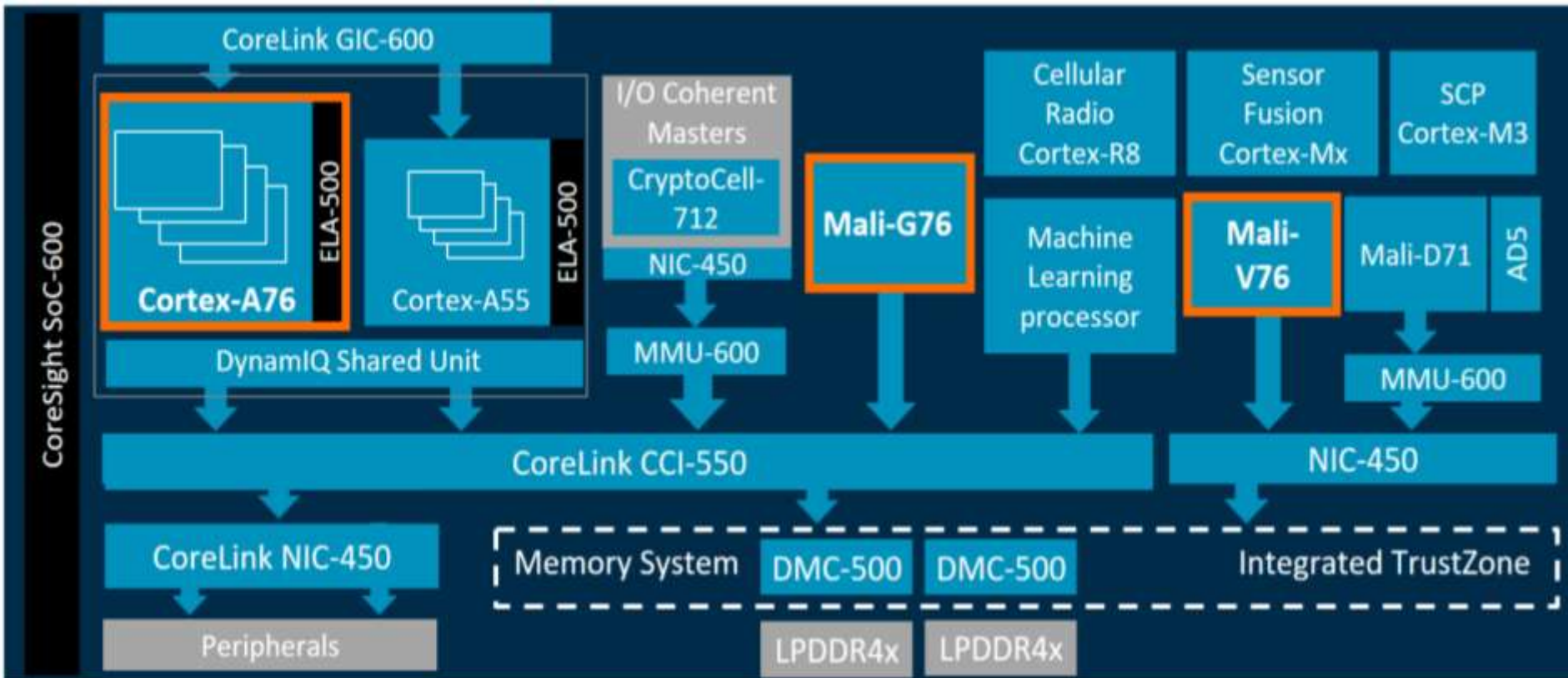
6.4 The high performance Cortex-A76 (12)

Single core performance of competing processors for Geekbench 4 [119]



6.4 The high performance Cortex-A76 (13)

System architecture of a Cortex-A76 based mobile system [120]

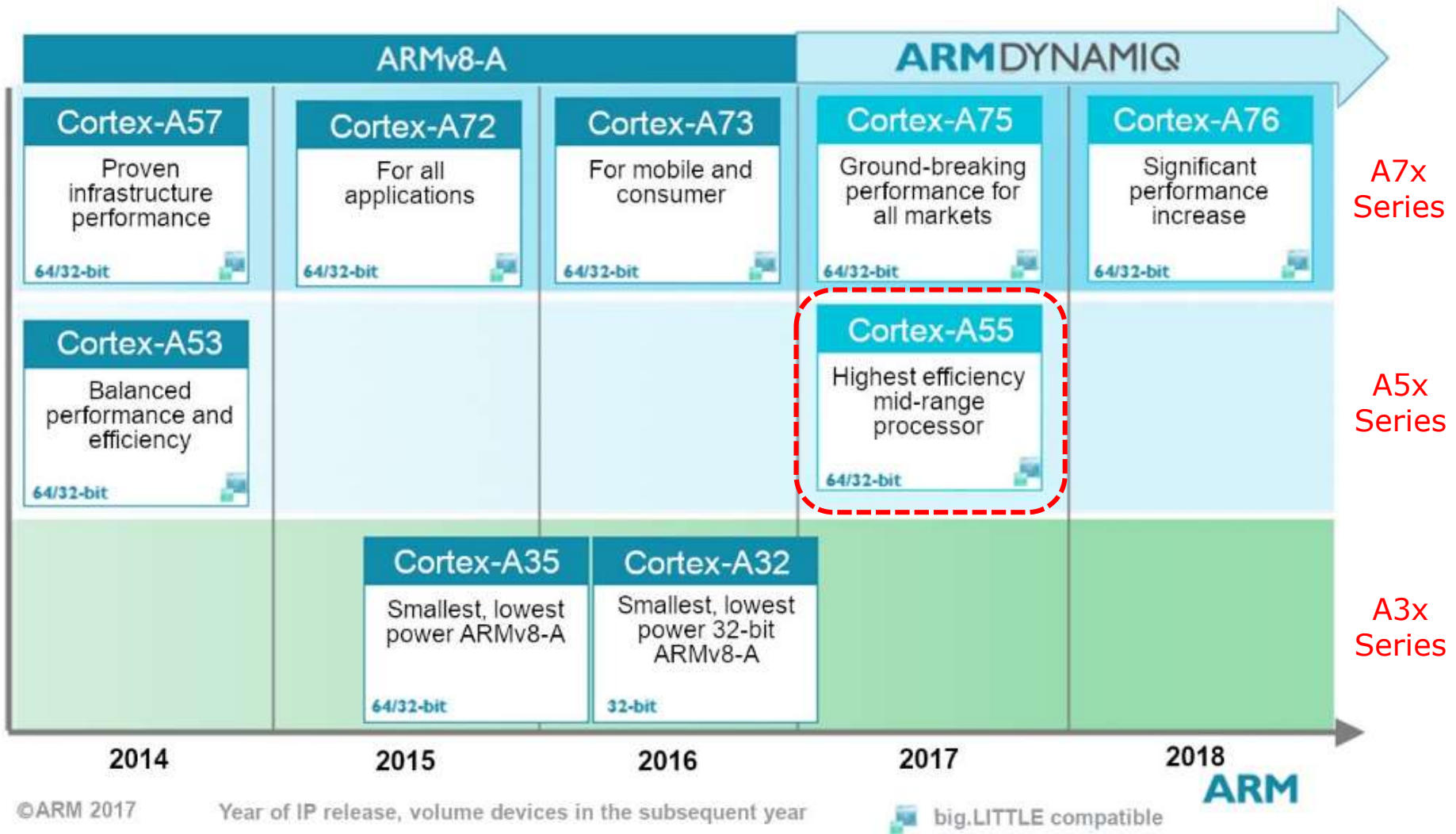


6.4 The high efficiency Cortex-A55

(It won't be discussed)

6.4 The high efficiency Cortex-A55 (1)

6.4 The high efficiency Cortex-A55 -1



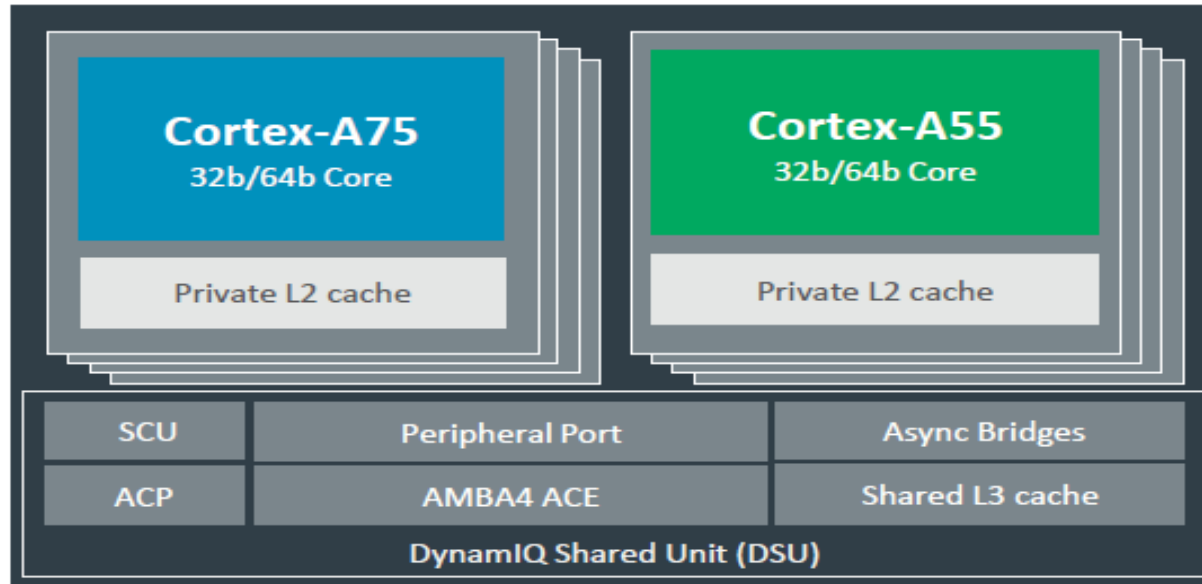
Source: AMD

The high efficiency Cortex-A55 -2

- **05/2017: Announced** along with the high-performance Cortex-A75 and the MALI 72 GPU.
- Target process technology: **10 nm**.

6.4 The high efficiency Cortex-A55 (3)

The Cortex-A55 as the building block of ARM's DynamIQ core clusters [103]



1b+7L



2b+6L



4b+4L



1b+2L



1b+3L

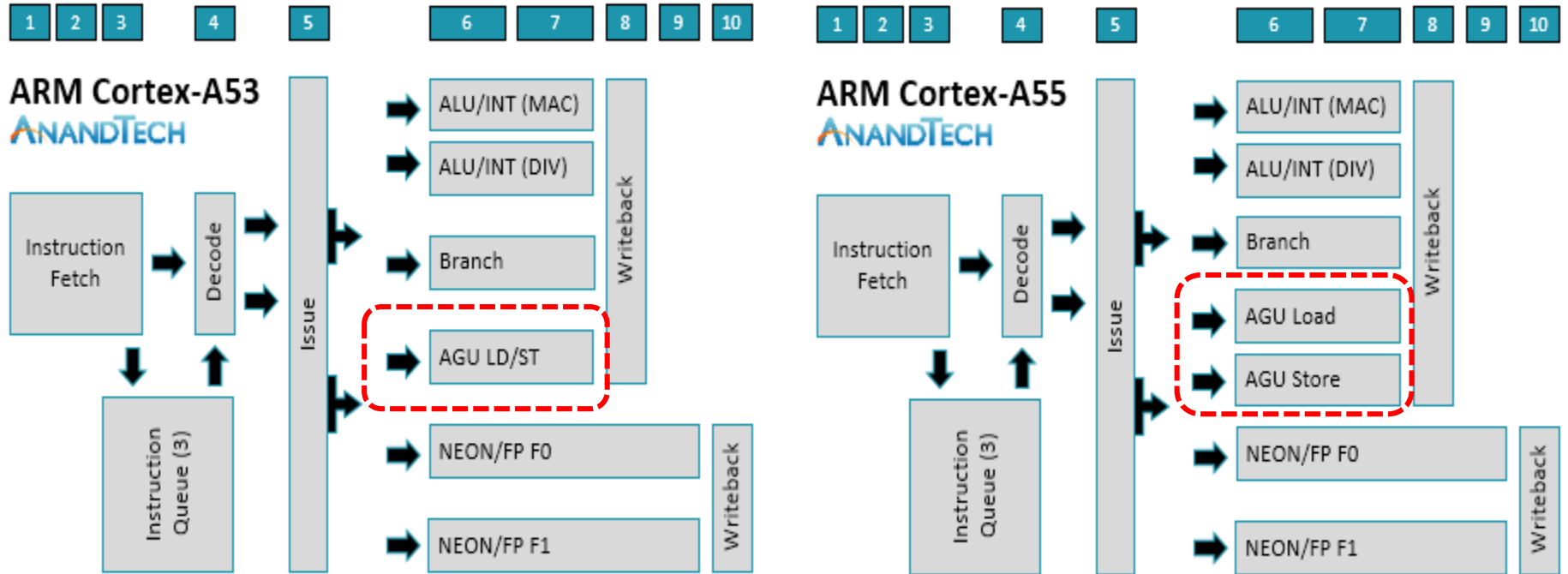


1b+4L

- Implements the ARMv8.2 ISA
- Up to 8 cores (up to 4 Cortex-A75, up to 8 Cortex-A55)

6.4 The high efficiency Cortex-A55 (4)

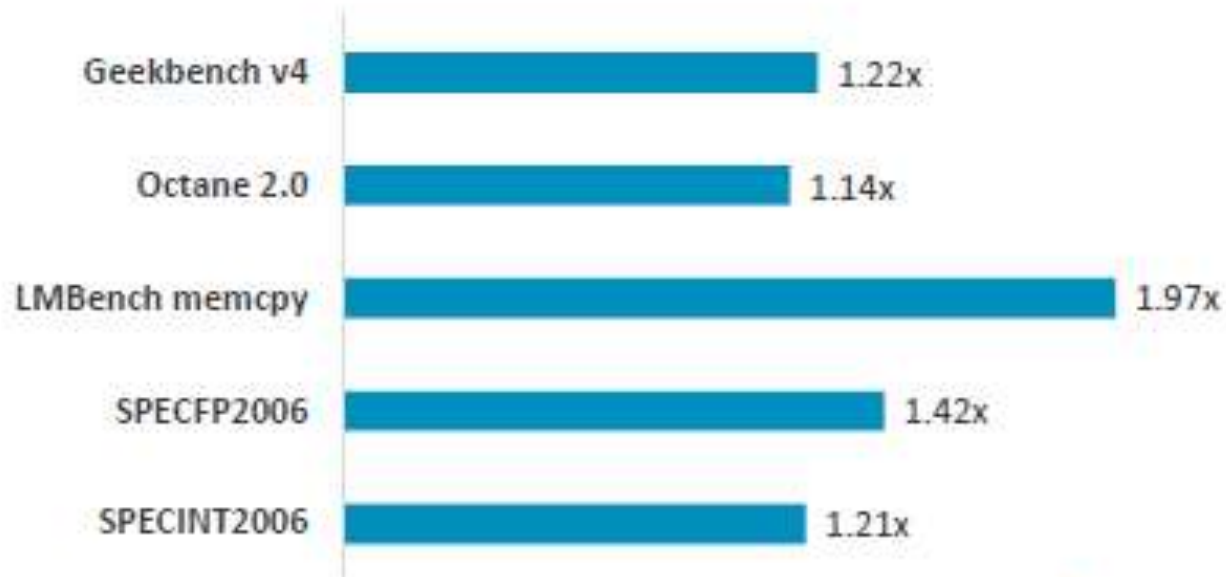
Enhancements of the Cortex-A55 vs. the Cortex-A53 [98]



6.4 The high efficiency Cortex-A55 (5)

Contrasting the performance of the Cortex-A55 vs. the Cortex-A53 [106]

Cortex-A55



Baseline to Cortex-A53

*All comparisons at ISO
process and frequency*



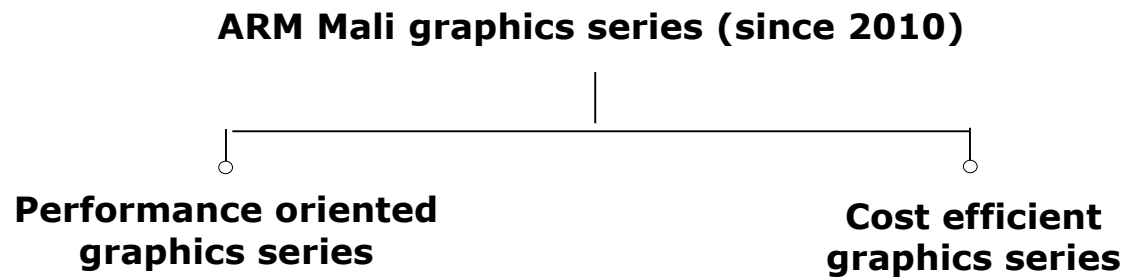
7. Overview of ARM's Mali graphics series

Will not be discussed.

7. Overview of ARM's Mali graphics series (1)

7. Overview of ARM's Mali graphics series (since 2010)

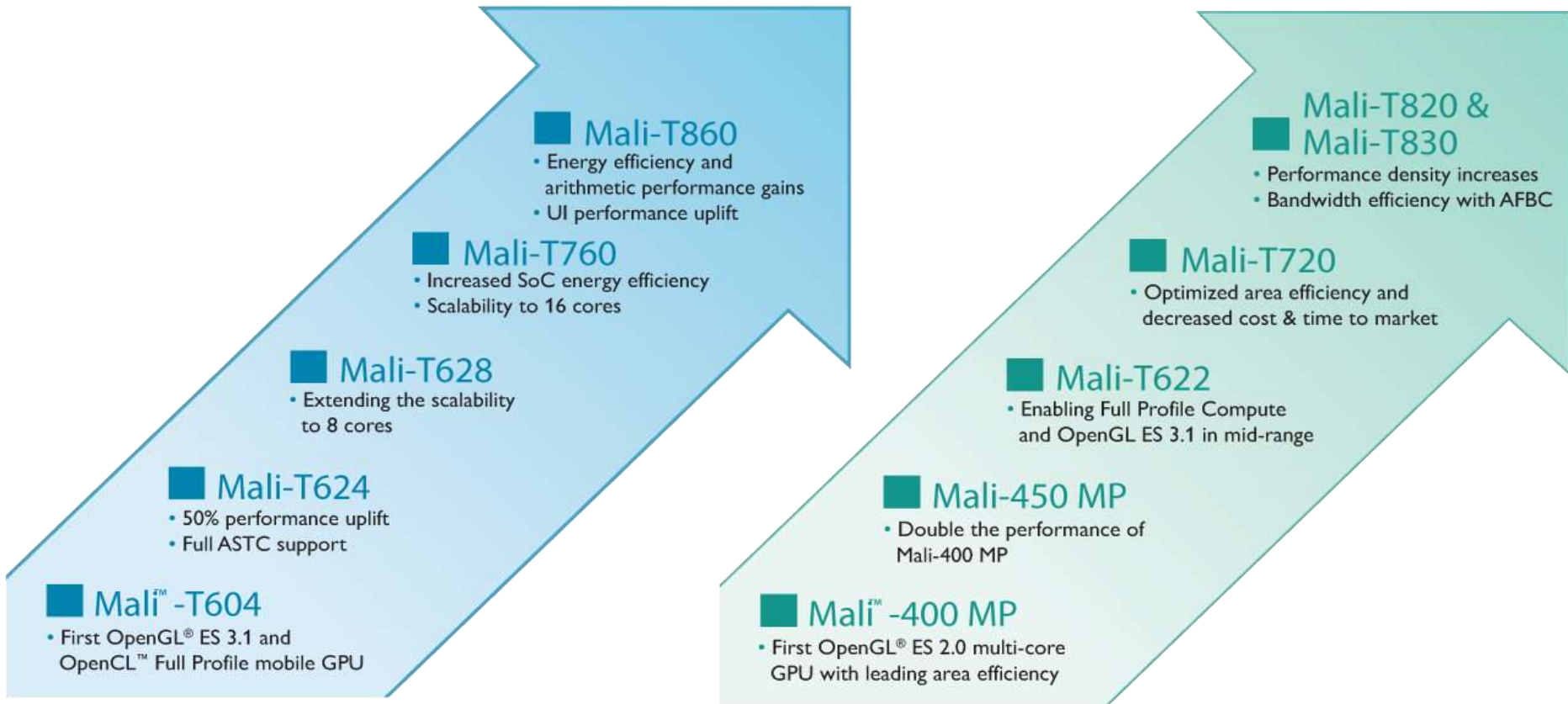
- ARM also designs and licenses **GPUs** to be used as companion chips to their CPU chips.
- ARM's GPU lines are designated as **Mali graphics series**.



7. Overview of ARM's Mali graphics lines (2)

Overview of ARM's Mali graphics series

Only GPU parts announced since about 2010 will be discussed [20]



Performance oriented graphics series

Mali-T6xx - Mali-T8xx

Cost efficient graphics series
Mali-T4xx - Mali-T8xx

7. Overview of ARM's Mali graphics lines (3)7

ARM's performance oriented Mali graphics series [20]

ARM® Mali™ -T604

Inter-Core Task Management

Shader Core Shader Core Shader Core Shader Core

Memory Management Unit

Level 2 Cache/SCU

AMBA®4 ACE-Lite

ARM® Mali™ -T624

Inter-Core Task Management

Shader Core Shader Core Shader Core Shader Core

Memory Management Unit

Level 2 Cache/SCU

AMBA®4 ACE-Lite

ARM® Mali™ -T628

Inter-Core Task Management

Shader Core Shader Core Shader Core Shader Core
Shader Core Shader Core Shader Core Shader Core

MMU

MMU

Level 2 Cache

Level 2 Cache

AMBA®4 ACE-Lite

AMBA®4 ACE-Lite

ARM® Mali™ -T760

Inter-Core Task Management

SC SC SC SC SC SC SC SC
SC SC SC SC SC SC SC SC

Advanced Tiling Unit

Memory Management Unit

L2 Cache

L2 Cache

AMBA®4 ACE-Lite

AMBA®4 ACE-Lite

ARM® Mali™ -T860 GPU

Inter-Core Task Management

SC SC SC SC SC SC SC SC
SC SC SC SC SC SC SC SC

Advanced Tiling Unit

Memory Management Unit

L2 Cache

L2 Cache

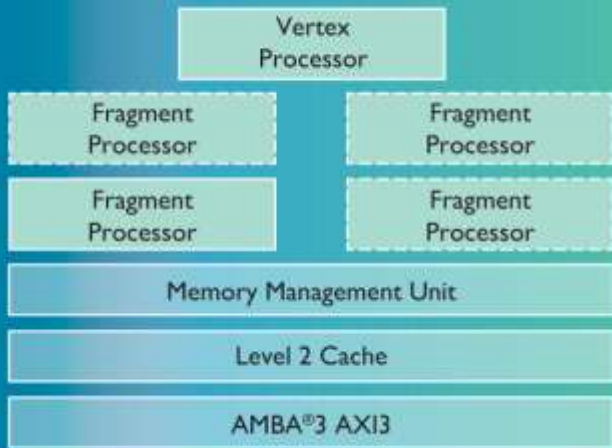
AMBA®4 ACE-Lite

AMBA®4 ACE-Lite

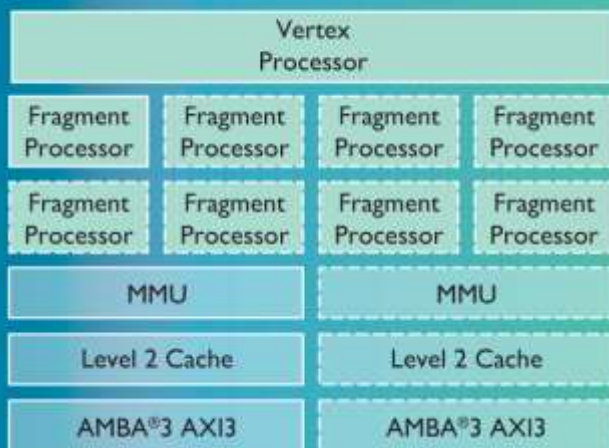
7. Overview of ARM's Mali graphics lines (4)

ARM's cost efficient Mali graphics series [20]

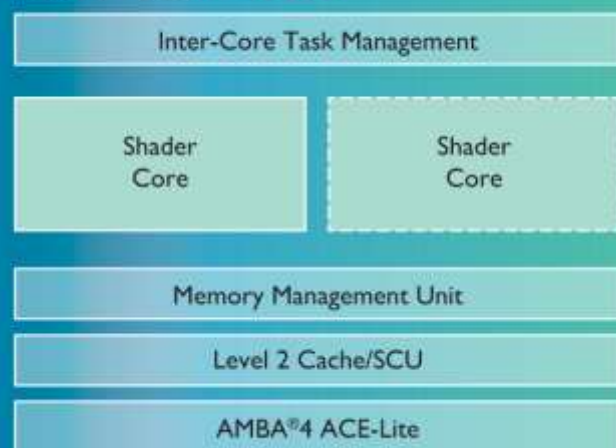
ARM® Mali™ -400



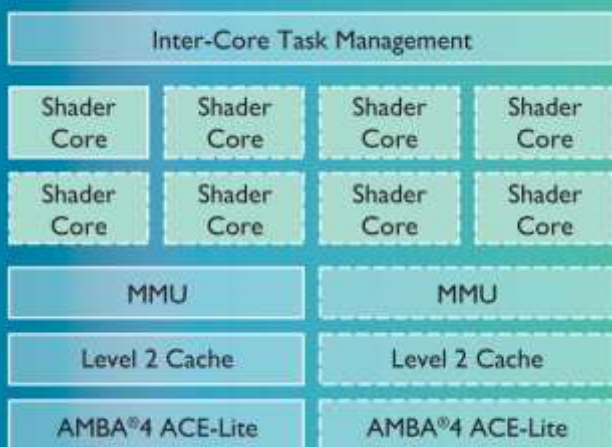
ARM® Mali™ -450 MP



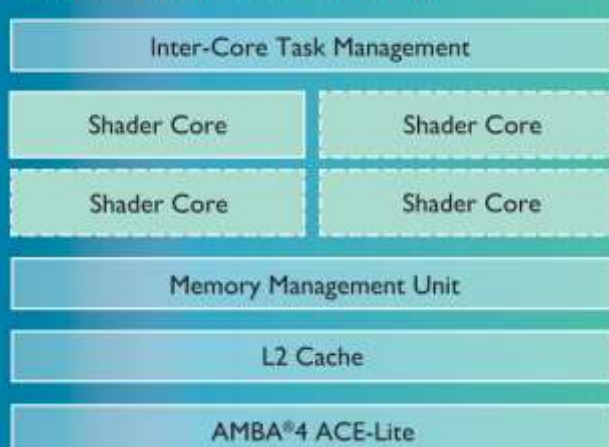
ARM® Mali™ -T622



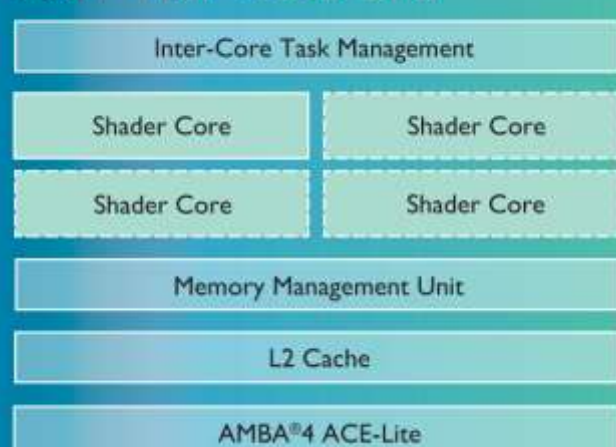
ARM® Mali™ -T720



ARM® Mali™ -T820 GPU

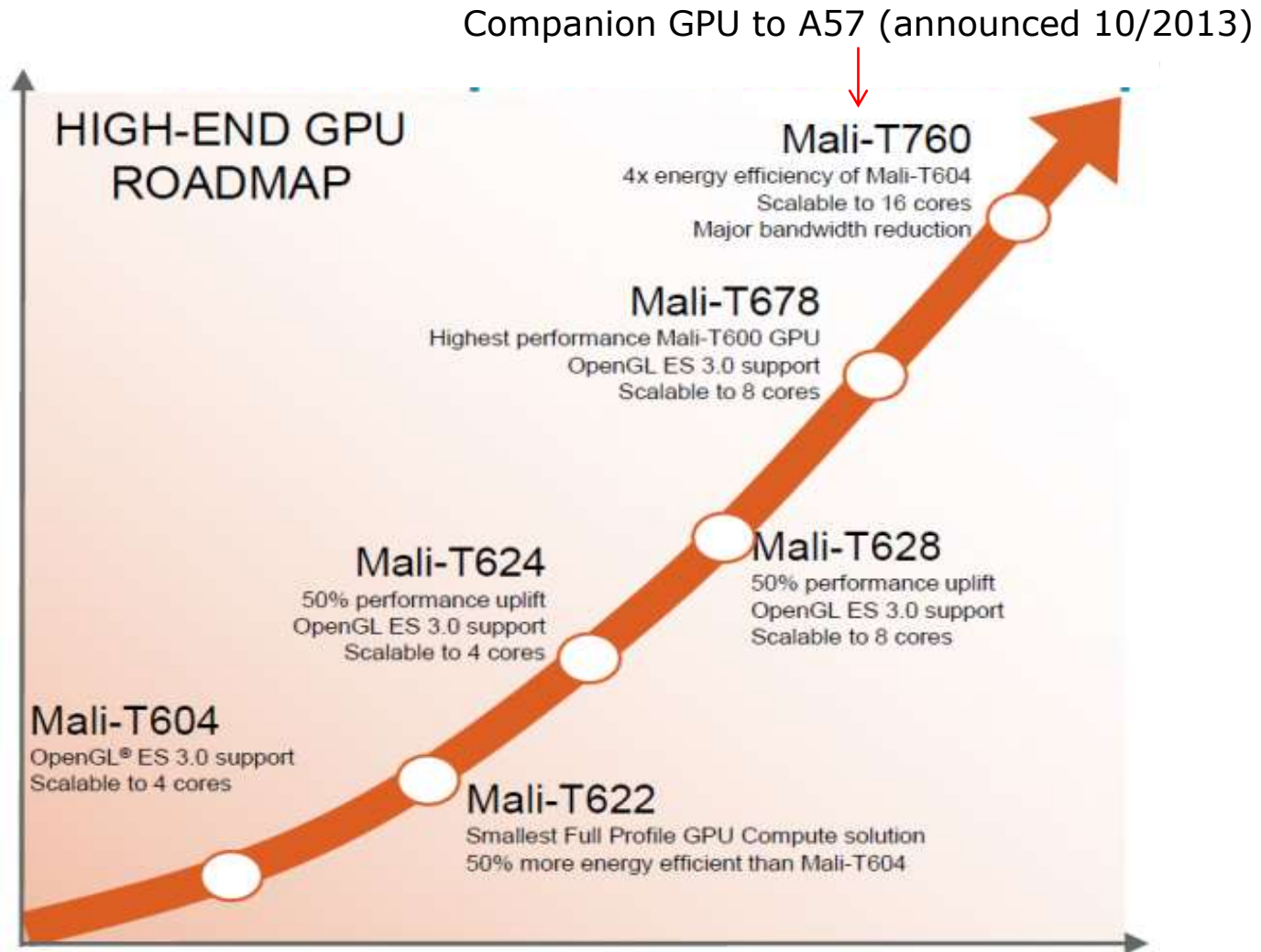


ARM® Mali™ -T830 GPU



7. Overview of ARM's Mali graphics lines (5)

ARM's high-end GPU roadmap [51]



8. References

8. References (1)

- [1]: Architecture and Implementation of the ARM Cortex-A8 Microprocessor, Design & Reuse, <http://www.design-reuse.com/articles/11580/architecture-and-implementation-of-the-arm-cortex-a8-microprocessor.html>
- [2]: Shimpi A.L., Answered by the Experts: ARM's Cortex A53 Lead Architect, Peter Greenhalgh, AnandTech, Dec. 17 2013, <http://www.anandtech.com/show/7591/answered-by-the-experts-arms-cortex-a53-lead-architect-peter-greenhalgh>
- [3]: Hill S., Design of a Reusable 1GHz, Superscalar ARM Processor, Hot Chips-18, Aug. 22 2006, http://www.hotchips.org/wp-content/uploads/hc_archives/hc18/3_Tues/HC18.S6/HC18.S6T3.pdf
- [4]: Details of a New Cortex Processor Revealed, Cortex-A9, ARM Developers' Conference, Oct. 2007, <http://rtcgroup.com/arm/2007/presentations/174%20-%20Details%20of%20a%20New%20Cortex%20Processor%20Revealed%20Cortex-A9.pdf>
- [5]: ARM Teaching Material, http://www.arm.com/files/ppt/ARM_Teaching_Material.ppt
- [6]: ARM Cortex Application Processors, <http://www.arm.com/products/processors/index.php>
- [7]: ARM SecurCore Processors, <http://www.arm.com/products/processors/securcore/index.php>
- [8]: Snyder C.D., ARM Family Expands at EPF, ARM11 Microarchitecture Stretches Pipe to Boost Frequency, Microprocessor, June 3 2002
- [9]: Hirata K., ARM11 MPCore, The streamlined and scalable ARM11 processor core, Jan. 2007, <http://www.aspdac.com/aspdac2007/pdf/archive/7D-2.pdf>

8. References (2)

- [10]: ARM Introduces The Cortex-M3 Processor To Deliver High Performance In Low-Cost Applications, Oct. 19 2004, <http://www.arm.com/about/newsroom/6750.php>
- [11]: NEON, ARM Technologies, <http://www.arm.com/products/processors/technologies/neon.php>
- [12]: Goodacre J., The Evolution of the ARM Architecture Towards Big Data and the Data-Centre, 8th Workshop on Virtualization in High-Performance Cloud Computing (VHPC'13), Nov. 17-22 2013, <http://www.virtical.eu/pub/sc13.pdf>
- [13]: Ferguson I., Redefining User Experiences, from IoT to Smart Mobile Devices, June 14 2013, http://www.arm.com/zh/files/event/arm_multimedia_seminar_shenzhen_2013_ianferguson.pdf
- [14]: Goto H., ARM Cortex – A Family Architecture, 2010, <http://pc.watch.impress.co.jp/video/pcw/docs/423/409/p1.pdf>
- [15]: Stevens H., Introduction to AMBA 4 ACE and big.LITTLE Processing Technology, White Paper, June 6 2011, http://www.arm.com/files/pdf/CacheCoherencyWhitepaper_6June2011.pdf
- [16]: Shimpi A.L., The ARM Diaries, Part 2: Understanding the Cortex A12, AnandTech, July 17 2013, <http://www.anandtech.com/show/7126/the-arm-diaries-part-2-understanding-the-cortex-a12>
- [17]: Shimpi A.L., ARM Cortex A17: An Evolved Cortex A12 for the Mainstream in 2015, AnandTech, Febr. 11 2014, <http://www.anandtech.com/show/7739/arm-cortex-a17>
- [18]: Ryan H., Intel, AMD & ARM Processors, University of Wisconsin DoIT Techstore <https://kb.wisc.edu/showroom/page.php?id=4927>

8. References (3)

- [19]: Rao A., ARM: a mandatory primer, Element14, 2014
- [20]: Mali Performance Efficient Graphics, ARM
<http://www.arm.com/products/multimedia/mali-performance-efficient-graphics/index.php>
- [21]: Cortex-A8 Processor, ARM Cortex-A Series,
<http://www.arm.com/products/processors/cortex-a/cortex-a8.php>
- [22]: Architecture and Implementation of the ARM Cortex-A8 Microprocessor, White Paper, Oct. 2005, https://www.pixhawk.ethz.ch/_media/software/optimization/neon_whitepaper.pdf
- [23]: Shimpi A.L., Understanding the iPhone 3GS, AnandTech, July 7 2009,
<http://www.anandtech.com/print/2798/>
- [24]: The ARM Cortex-A9 Processors, White Paper, Sept. 2009,
<http://www.arm.com/files/pdf/armcortexa-9processors.pdf>
- [25]: Cortex-A9 MPCore, Technical Reference Manual, 2008-2012,
http://infocenter.arm.com/help/topic/com.arm.doc.ddi0407i/DDI0407I_cortex_a9_mpcore_r4p1_trm.pdf
- [26]: Goto H., ARM Cortex-A9r4 Core Block Diagram,
<http://pc.watch.impress.co.jp/video/pcw/docs/614/543/08p.pdf>
- [27]: ARM Unveils Cortex-A9 Processors For Scalable Performance and Low-Power Designs, Oct. 3 2007, <http://www.arm.com/about/newsroom/18688.php>
- [28]: Goodacre J., The Effect and Technique of System Coherence in ARM Multicore Technology,
<http://www.mpsoc-forum.org/previous/2008/slides/8-6%20Goodacre.pdf>

8. References (4)

- [29]: Goto H., ARM Cortex-A7/A9 vs Bobcat, Atom, 2010,
<http://pc.watch.impress.co.jp/img/pcw/docs/487/030/html/9.jpg.html>
- [30]: Goto H., ARM Cortex-A12 Block Diagram, 2013,
<http://pc.watch.impress.co.jp/video/pcw/docs/621/747/gp1.pdf>
- [31]: Rosinger S., The Top 5 Things to Know about Cortex-A12, ARM Connected Community,
Oct. 26 2013, <http://community.arm.com/groups/processors/blog/2013/10/26/the-top-5-things-to-know-about-cortex-a12>
- [32]: ARM Cortex-A17 MPCore Processor, Technical Reference Manual, 2014,
http://infocenter.arm.com/help/topic/com.arm.doc.ddi0535b/DDI0535B_cortex_a17_r1p0_trm.pdf
- [33]: Cortex-A17 Processor, ARM Cortex-A Series,
<http://www.arm.com/products/processors/cortex-a/cortex-a17-processor.php>
- [34]: Rosinger S., ARM Cortex-A17/Cortex-A12 processor update, ARM Connected Community,
Oct. 1 2014, <http://community.arm.com/groups/processors/blog/2014/09/30/arm-cortex-a17-cortex-a12-processor-update>
- [35]: Cortex-A15 Processor, ARM Cortex-A Series,
<http://www.arm.com/products/processors/cortex-a/cortex-a15.php>
- [36]: Lanier T., Exploring the Design of the Cortex-A15 Processor,
http://www.arm.com/files/pdf/AT-Exploring_the_Design_of_the_Cortex-A15.pdf

8. References (5)

- [37]: Greenhalgh P., big.LITTLE Processing with ARM Cortex-A15 & Cortex-A7, White Paper, Sept. 2011, http://www.arm.com/files/downloads/big_LITTLE_Final_Final.pdf
- [38]: Stokes J., ARM fills out CPU lineup with Cortex A5, Ars Technica, Oct. 22 2009, <http://arstechnica.com/gadgets/2009/10/arm-fills-out-cpu-lineup-with-cortex-a5/>
- [39]: Hruska J., ARM Launches New Cortex-A5 As A Bulkward Against Future Atom processors, Hot Hardware, Oct. 23 2009, <http://hothardware.com/News/ARM-Launches-New-CortexA5-As-A-Bulkward-Against-Future-Atom-processors>
- [40]: Cortex-A5 Processor, ARM Cortex-A Series, <http://www.arm.com/products/processors/cortex-a/cortex-a5.php>
- [41]: Bhattacharya A., Small, Quiet, and Cool, Power Efficient Processing with the Cortex-A5 Processor, http://www.arm.com/files/pdf/at2_-_power_efficient_processing_with_the_cortex-a5_v1.pdf
- [42]: Cortex-A5, The smallest, lowest power ARMv7 application processor, <http://www.hitex.co.uk/fileadmin/uk-files/pdf/ARM%20Seminar%20Presentations%202013/Hitex%20Cortex-A5%20Overview.pdf>
- [43]: Flautner K., Heterogeneity to the rescue, Nov. 2011, <https://www.bscmsrc.eu/sites/default/files/media/arm-heterogenous-mp-november-2011.pdf>
- [44]: Shimpi A.L., ARM's Cortex A7: Bringing Cheaper Dual-Core & More Power Efficient High-End Devices, AnandTech, Oct. 19 2011, <http://www.anandtech.com/show/4991/>

8. References (6)

- [45]: Cortex-A7 Processor, ARM Cortex-A Series,
<http://www.arm.com/products/processors/cortex-a/cortex-a7.php>
- [46]: Goto H., New design of mid-range CPU that ARM has announced "Cortex-A12", PC Watch,
June 4 2013, http://pc.watch.impress.co.jp/docs/column/kaigai/20130604_602106.html
- [47]: Shimpi A.L., ARM's Cortex A57 and Cortex A53: The First 64-bit ARMv8 CPU Cores,
AnandTech, Oct. 30 2012, <http://www.anandtech.com/show/6420/arms-cortex-a57-and-cortex-a53-the-first-64bit-armv8-cpu-cores>
- [48]: Smith K., Next-Generation Solutions: One Size Does Not Fit All, Nov. 2012,
http://www.armtechforum.com.cn/2012/3_Next-generation_Solutions_One_Size_does_not_Fit_All.pdf
- [49]: Smythe I., Building the future of 64-bit computing with ARMv8-A, June 2014,
http://www.arm.com/files/event/2014_ARM_Multimedia_Seminar_ARM_Ian_Smythe.pdf
- [50]: Ferguson I., ARM Servers, Why, Where, when, Nov. 27 2012,
http://openserversummit.com/English/Collaterals/Proceedings/2012/20121127_SA103_Ferguson.pdf
- [51]: Crijns K., ARM: 20 dollar and 64-bit Android smartphones to be expected, Hardware.info,
June 24 2014, <http://us.hardware.info/reviews/5386/2/arm-20-dollar-and-64-bit-android-smartphones-to-be-expected-android-64-bitn>
- [52]: Scaling Mobile Compute to the Data Centre, MPSoC'13,
<http://www.mpsoc-forum.org/previous/2013/slides/2-Goodacre.pdf>

8. References (7)

- [53]: ARM Launches Cortex-A50 Series, the World's Most Energy-Efficient 64-bit Processors, TechPowerUp, Oct. 30 2012, <http://www.techpowerup.com/174709/arm-launches-cortex-a50-series-the-worlds-most-energy-efficient-64-bit-processors.html>
- [54]: Mandyam L., Smartphone Powered Data Centers: Shifting Toward Energy Efficiency, <http://sites.ieee.org/scv-cs/files/2013/03/IEEE-event-April-slide-upload.pdf>
- [55]: Anthony S., ARM says \$20 smartphones coming this year, shows off 64-bit Cortex-A53 and A57 performance, Extreme Tech, May 6 2014, <http://www.extremetech.com/computing/181935-arm-says-20-smartphones-coming-this-year-shows-off-64-bit-cortex-a53-and-a57-performance>
- [56]: Merritt R., ARM stretches out with A5 core, graphics, FPGAs, Oct. 21 2009, <http://www.embedded.com/print/4085371>
- [57]: Cormie D., The ARM11 Microarchitecture, April 2002
- [58]: Exynos 5 Octa, Block Diagram, <http://www.samsung.com/global/business/semiconductor/product/application/detail?productId=7978&iaId=2341>
- [59]: Grabham D., From a small Acorn to 37 billion chips: ARM's ascent to tech superpower, Techradar, July 19 2013, <http://www.techradar.com/news/computing/from-a-small-acorn-to-37-billion-chips-arm-s-ascent-to-tech-superpower-1167034>
- [60]: The ARM Architecture, <http://www.eng.auburn.edu/~strouce/DaTseminar/UniPres07s.pdf>
- [61]: Wikipedia, ARM architecture, http://en.wikipedia.org/wiki/ARM_architecture

8. References (8)

- [62]: Levy M., The History of The ARM Architecture: From Inception to IPO,
<http://www.reds.ch/share/cours/ReCo/documents/TheHistoryOfTheArmArchitecture.pdf>
- [63]: Lemieux J., Introduction to ARM thumb, Embedded, Sept. 24 2003, <http://www.embedded.com/electronics-blogs/beginner-s-corner/4024632/Introduction-to-ARM-thumb>
- [64]: ARM Processor Architecture,
<http://www.arm.com/products/processors/instruction-set-architectures/index.php>
- [65]: ARM Architecture Reference Manual, ARMv5, DDI0100E, June 2000
- [66]: ARM Architecture Reference Manual, ARMv8, 2013, http://www.myir-tech.com/down/arm/arch/ARMv8-A_Architecture_Reference_Manual_%28Issue_A.a%29.pdf
- [67]: Porthouse C., Use ARM DBX hardware extensions to accelerate Java in space-constrained embedded apps, Embedded, Oct. 18 2007, <http://www.embedded.com/design/prototyping-and-development/4007206/3/PRODUCT-HOW-TO-Use-ARM-DBX-hardware-extensions-to-accelerate-Java-in-space-constrained-embedded-apps>
- [68]: ARM Security Technology, Building a Secure System using TrustZone Technology, 2009,
http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf
- [69]: ARM Cortex-A53 MPCore Processor Technical Reference Manual, Cryptography Extension, 2013-2014, <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0500e/CJHDEBAF.html>

8. References (9)

- [70]: Shilov A., ARM's high-end 'Ares' core for 10nm SoCs may be unveiled next year, KitGuru, May 16 2015, <http://www.kitguru.net/components/cpu/anton-shilov/arms-high-performance-ares-core-may-be-unleashed-next-year/>
- [71]: Goto H., ARM Cortex-A57 Block Diagram, <http://images.anandtech.com/doci/8718/Hiroshige.Goto.png>
- [72]: Wasson S., Inside ARM's Cortex-A72 microarchitecture, TechReport, May 1 2015, <http://techreport.com/review/28189/inside-arm-cortex-a72-microarchitecture>
- [73]: Citizendium, Java platform, http://en.citizendium.org/wiki/Java_platform
- [74]: Wasson S., Samsung's Galaxy Note 4 with the Exynos 5433 processor, TechReport, 01/31/2015, <http://techreport.com/review/27539/samsung-galaxy-note-4-with-the-exynos-5433-processor/2>
- [75]: Frumusanu A. & Smith R., ARM A53/A57/T760 investigated - Samsung Galaxy Note 4 Exynos Review, AnandTech, February 10, 2015, <http://www.anandtech.com/show/8718/the-samsung-galaxy-note-4-exynos-review>
- [76]: Riemenschneider F., Intel sei dank: Programmierbare Logik mit 1 GHz takten, Elektroniknet, 29.10.2013 von Frank Riemenschneider, <http://www.elektroniknet.de/halbleiter/programmierbare-logik/artikel/102160/>
- [77]: Frumusamu A., ARM Announces New Cortex-A35 CPU - Ultra-High Efficiency For Wearables & More, AnandTech, *November 10, 2015*, <http://anandtech.com/show/9769/arm-announces-cortex-a35>

8. References (10)

- [78]: Quested T., ARM scales up again in Cambridge, Businessweekly, 20 October, 2014,
<http://www.businessweekly.co.uk/news/property-and-construction/17695-arm-scales-again-cambridge#sthash.Pzg7jnpK.dpuf>
- [79]: ARM Architecture Reference Manual, Thumb-2 Supplement, 2004-2005,
<http://read.pudn.com/downloads159/doc/709030/Thumb-2SupplementReferenceManual.pdf>
- [80]: Wanted: Java bytecode disassembler that shows addresses, opcodes, operands, in hex, Reverse Engineering, 2013, <http://reverseengineering.stackexchange.com/questions/2036/wanted-java-bytecode-disassembler-that-shows-addresses-opcodes-operands-in-h>
- [81]: Barr M., KVM: A Small Java Virtual Machine for J2ME, Barr Group, May 4 2016,
<http://www.barrgroup.com/Embedded-Systems/How-To/KVM-J2ME-Java-Virtual-Machine>
- [82]: What is Bytecode ?, March 21 2015,
<http://interview-question-and-answers.blogspot.hu/2015/03/what-is-bytecode.html>
- [83]: The JVM – Java Virtual Machine, Android Developer, May 8 2015,
<http://androiddeveloper.co.il/the-jvm-java-virtual-machine/>
- [84]: Java Virtual Machine, Free Download Java Virtual Machine, How to Download JVM, Dev Manuals, Sept. 28 2010,
<http://www.devmanuals.com/tutorials/java/corejava/javavirtualmachine.html>
- [85]: Srinivasan K., What is the difference between JRE,JVM and JDK?, Javabeat, Febr. 21 2013,
<http://www.javabeat.net/what-is-the-difference-between-jrejvm-and-jdk/>

8. References (11)

- [86]: ARM Architecture Reference Manual, ARMv7-A and ARMv7-R edition, 2004-2012,
<http://liris.cnrs.fr/~mmrissa/lib/exe/fetch.php?media=armv7-a-r-manual.pdf>
- [87]: Cortex-A7 MPCore, Technical Reference Manual, Rev. r0p3, 2011-2012,
http://infocenter.arm.com/help/topic/com.arm.doc.ddi0464d/DDI0464D_cortex_a7_mpcore_r0p3_trm.pdf
- [88]: Pardoe A., New Innovations in .NET Runtime, dotnetConf, 2014, Jun. 2014,
<https://view.officeapps.live.com/op/view.aspx?src=http%3a%2f%2ffiles.channel9.msdn.com%2fthumbnail%2f47cb5ae5-eb38-404d-80e8-7bae0a4efbaf.pptx>
- [89]: Cortex-A73 Processor, ARM,
<https://www.arm.com/products/processors/cortex-a/cortex-a73-processor.php>
- [90]: Frumusanu A., ARM Announces New Cortex-A35 CPU - Ultra-High Efficiency For Wearables & More, AnandTech, *Nov. 10, 2015*,
<http://www.anandtech.com/show/9769/arm-announces-cortex-a35>
- [91]: Frumusanu A., ARM Reveals Cortex-A72 Architecture Details, AnandTech, Apr. 23, 2015,
<http://www.anandtech.com/show/9184/arm-reveals-cortex-a72-architecture-details>
- [92]: Frumusanu A., The ARM Cortex A73 - Artemis Unveiled, AnandTech, May 29, 2016,
<http://www.anandtech.com/show/10347/arm-cortex-a73-artemis-unveiled>
- [93]: Cortex-A73 Overview, ARM Developer, 2016,
<https://developer.arm.com/products/processors/cortex-a/cortex-a73>
- [94]: Cortex-A35, ARM, 2015,
<http://www.arm.com/products/processors/cortex-a/cortex-a35-processor.php>

8. References (12)

- [95]: Wasson S., Inside ARM's Cortex-A72 microarchitecture - The next-gen CPU core for mobile devices and servers, Tech Report, May 1, 2015,
<http://techreport.com/review/28189/inside-arm-cortex-a72-microarchitecture>
- [96]: Stephens N., ARMv8-A Next-Generation Vector Architecture for HPC, Hot Chips 2016,
<https://community.arm.com/groups/processors/blog/2016/08/22/technology-update-the-scalable-vector-extension-sve-for-the-armv8-a-architecture>
- [97]: Brash D., The ARMv8-A architecture and its ongoing development, ARM, 2014
<https://community.arm.com/processors/b/blog/posts/the-armv8-a-architecture-and-its-ongoing-development>
- [98]: Humrick M., Exploring DynamIQ and ARM's New CPUs: Cortex-A75, Cortex-A55, AnandTech, May 29, 2017,
<https://www.anandtech.com/show/11441/dynamiq-and-arms-new-cpus-cortex-a75-a55>
- [99]: Frumusanu A., The ARM Cortex A73 - Artemis Unveiled, AnandTech, May 29, 2016,
<http://www.anandtech.com/show/10347/arm-cortex-a73-artemis-unveiled>
- [100]: Turner C., ARM instruction sets and CPUs for wide ranging applications, ARM Tech Forum, Taipei, July 4, 2017,
https://www.arm.com/files/event/20170704_ATF_TW_A3.pdf
- [101]: Cortex-A75, ARM Developer,
<https://developer.arm.com/products/processors/cortex-a/cortex-a75>
- [102]: Cortex-A73, ARM Developer,
<https://developer.arm.com/products/processors/cortex-a/cortex-a73>

8. References (13)

- [103]: ARM Unveils Cortex-A75, A55 Processors And Mali-G72 GPU, ARM Developer, <https://www.xda-developers.com/arm-unveils-cortex-a75-a55-processors-and-mali-g72-gpu/>
- [104]: DynamIQ power management support, Vers. 1.1, ARM ECM 0640541 Oct. 30 2017
- [105]: Parris N., Boost SoC performance from edge to cloud ARM CoreLink System IP, ARM, Nov. 2016, http://www.armtechforum.com.cn/attached/article/2016ATS_C1_Neil_Parris_20161206151154.pdf
- [106]: Greenhalgh P., ARM DynamIQ. Intelligent Solutions Using Cluster Based Multiprocessing, Hot Chips 29, 2017 Aug. 12, <https://www.slideshare.net/ARMHoldings/arm-dynamiq-intelligent-solutions-using-cluster-based-multiprocessing>
- [107]: Wathan G. : Arm DynamIQ: Technology for the next era of compute, ARM Community, <https://community.arm.com/processors/b/blog/posts/arm-dynamiq-technology-for-the-next-era-of-compute>
- [108]: Walrath J., ARM Tech Day 2017: DynamIQ, Cortex-A55, A75, and Mali-G72, PC Perspective, May 29, 2017, <https://www.pcper.com/reviews/General-Tech/ARM-Tech-Day-2017-DynamIQ-Cortex-A55-A75-and-Mali-G72/Cortex-A75-and-Mali-G72>
- [109]: Rickards I. and Kucheria A., Energy Aware Scheduling (EAS) progress update, Linaro Company, Sept. 18, 2015, <http://www.linaro.org/blog/core-dump/energy-aware-scheduling-eas-progress-update/>

8. References (14)

- [110]: Triggs R., A closer look at ARM's new Cortex-A75 and Cortex-A55 CPUs, Android Authority, May 31, 2017
<https://www.androidauthority.com/arm-cortex-a75-cortex-a55-breakdown-770380/>
- [111]: Arm Cortex-A55: Efficient performance from edge to cloud, ARM Community,
<https://community.arm.com/processors/b/blog/posts/arm-cortex-a55-efficient-performance-from-edge-to-cloud>
- [112]: System Guidance for Infrastructure, ARM Developer,
<https://developer.arm.com/products/system-design/system-guidance/system-guidance-for-infrastructure>
- [113]: ARM® Architecture Reference Manual ARMv8, for ARMv8-A architecture profile, ARM DDI 0487A.k_iss10775 (ID092916),2013-2016
https://silver.arm.com/download/ARM_and_AMBA_Architecture/AR150-DA-70000-r0p0-01eac1/DDI0487A_k_armv8_arm_iss10775.pdf
- [114]: Arm DynamIQ: Technology for the next era of compute, ARM Community,
<https://community.arm.com/processors/b/blog/posts/arm-dynamiq-technology-for-the-next-era-of-compute>
- [115]: ARM unveils multi-processor core with Linux SMP support, LinuxDevices, May 17, 2004,
<http://linuxdevices.linuxgizmos.com/arm-unveils-multi-processor-core-with-linux-smp-support/>
- [116]: ARM Architecture Reference manual Supplement - The Scalable Vector Extension (SVE), for ARMv8-A, ARM DDI 0584Ab (ID081717), 21 Aug. 2017,
<https://developer.arm.com/docs/ddi0584/latest/arm-architecture-reference-manual-supplement-the-scalable-vector-extension-sve-for-armv8-a>

8. References (15)

- [117]: ARM Q4 2016 Roadshow Slides, ARM Holdings, 2017
file:///C:/Users/sima/Downloads/ARM_SB_Q4_2016_Roadshow_Slides_FINAL.pdf
- [118]: Triggs R., Everything you need to know about ARM's DynamIQ, Android Authority, May 29, 2017,
<https://www.androidauthority.com/arm-dynamiq-need-to-know-770349/>
- [119]: Frumusanu A., Arm's Cortex-A76 CPU Unveiled: Taking Aim at the Top for 7nm, AnandTech, May 31, 2018,
<https://www.anandtech.com/show/12785/arm-cortex-a76-cpu-unveiled-7nm-powerhouse>
- [120]: Hruska J., ARM's New Cortex-A76 SoC Targets Windows Laptop Market, ExtremeTech, May 31, 2018,
<https://www.extremetech.com/mobile/270362-arm-cortex-a76-targets-laptop-market>
- [121]: Humrick M., Exploring DynamIQ and ARM's New CPUs: Cortex-A75, Cortex-A55, AnandTech, May 29, 2017,
<https://www.anandtech.com/show/11441/dynamiq-and-arms-new-cpus-cortex-a75-a55/3>
- [122]: Cortex-A57 Software Optimization Guide, ARM, UAN 0015B, 2016,
http://infocenter.arm.com/help/topic/com.arm.doc.uan0015b/Cortex_A57_Software_Optimization_Guide_external.pdf
- [123]: ASIMD multiply-accumulate instruction, ARM Community, 2016,
<https://community.arm.com/processors/f/discussions/7028/asimd-multiply-accumulate-instruction>

[124]: Gianos C., Intel Xeon Processor E5-2600 v3 Product Family Architectural Overview ,
HPCC'14, Nov. 16 2014