



고려대학교
KOREA UNIVERSITY

COMP427 Embedded Systems

Lecture 6. VFP & NEON in ARM

Prof. Taeweon Suh

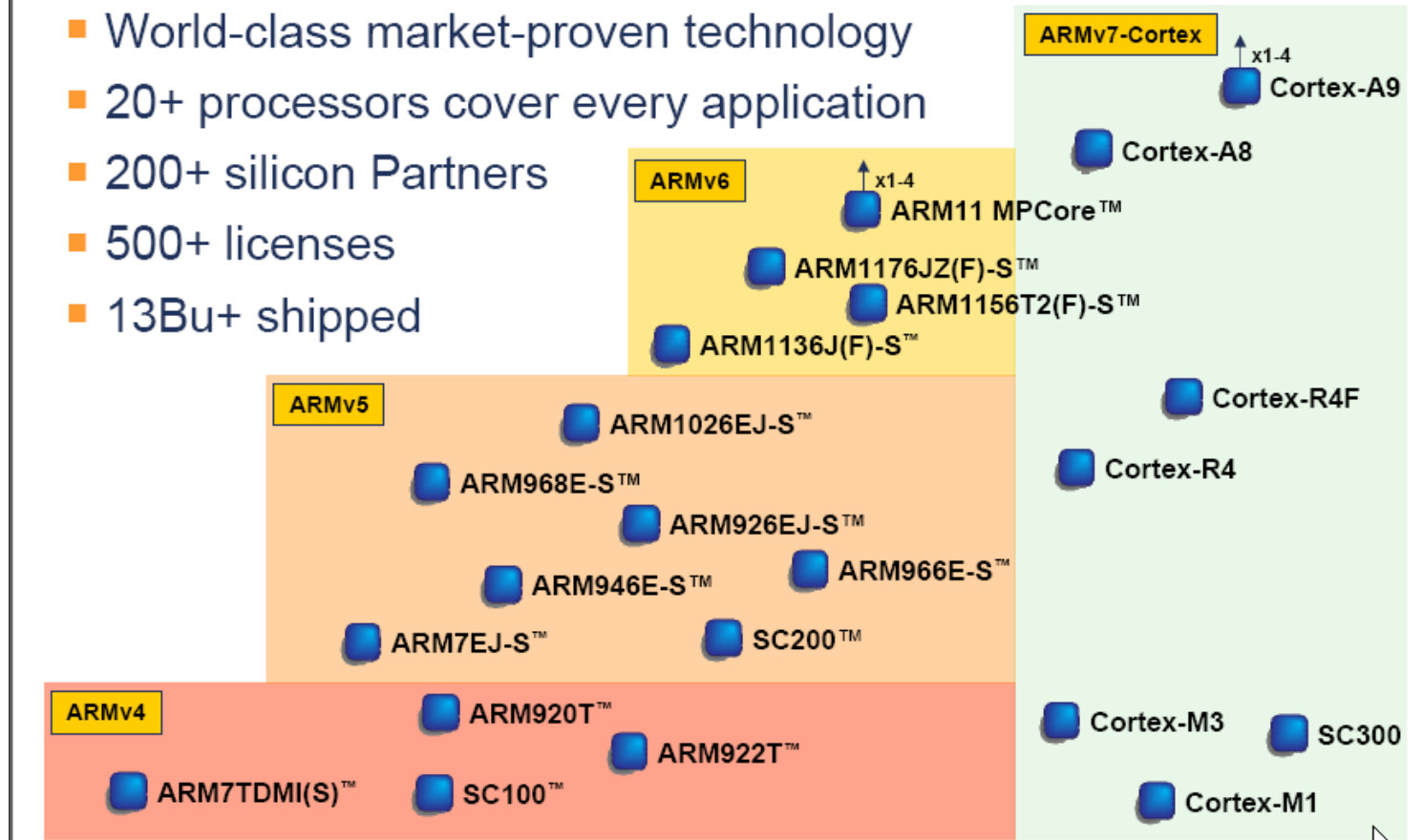
Computer Science & Engineering

Korea University

ARM Processor Portfolio

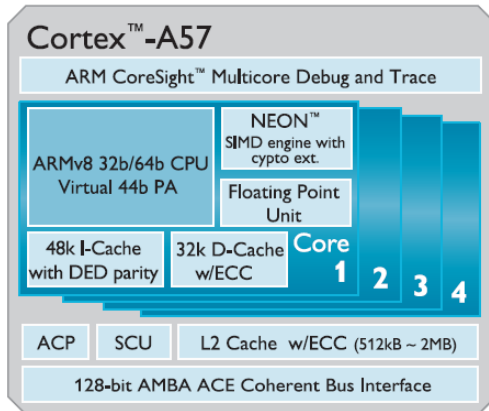
ARM Processor Portfolio

- World-class market-proven technology
- 20+ processors cover every application
- 200+ silicon Partners
- 500+ licenses
- 13Bu+ shipped

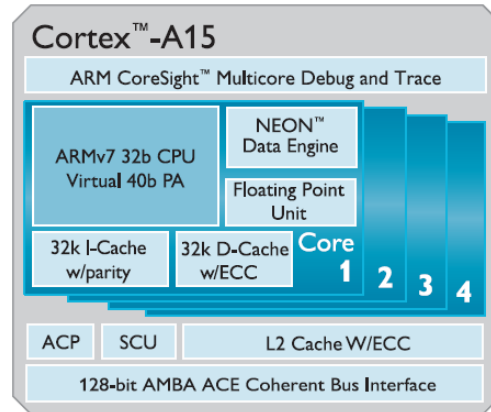


ARMv7-A

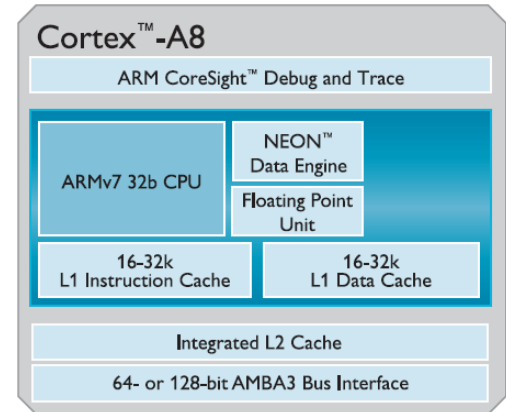
Cortex-A57 Processor



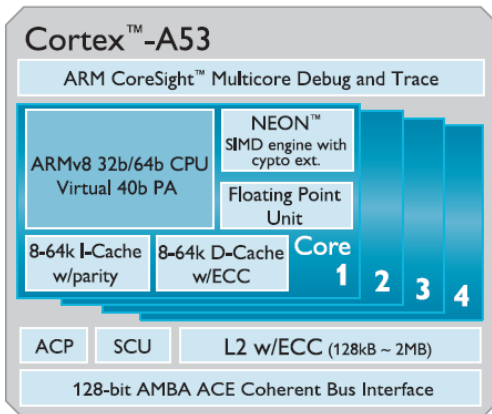
Cortex-A15 Processor



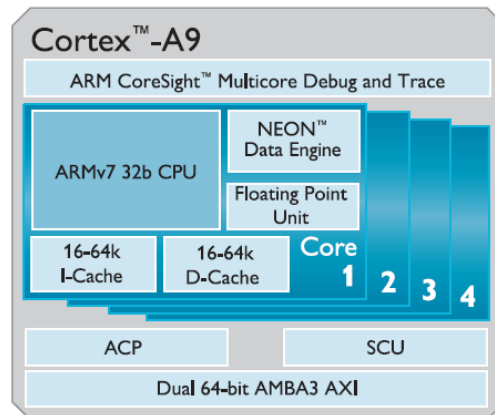
Cortex-A8 Processor



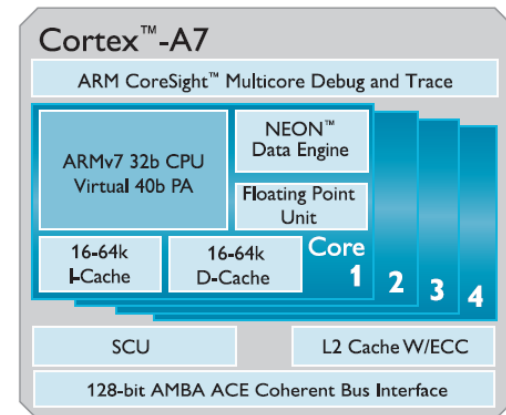
Cortex-A53 Processor



Cortex-A9 Processors



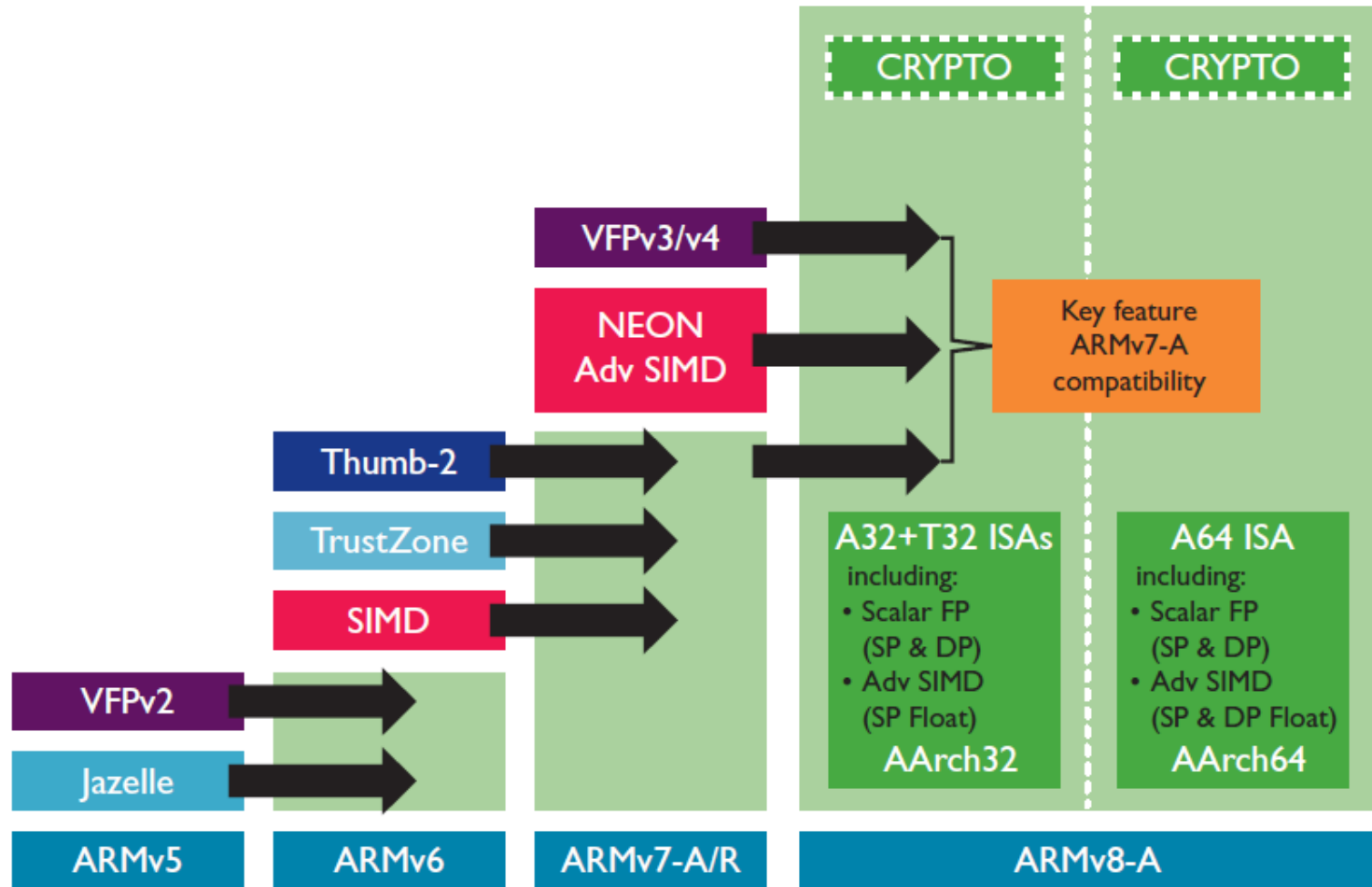
Cortex-A7 Processor



■ ACP: Accelerator Coherency Port

■ SCU: Snoop Control Unit

NEON & VFP

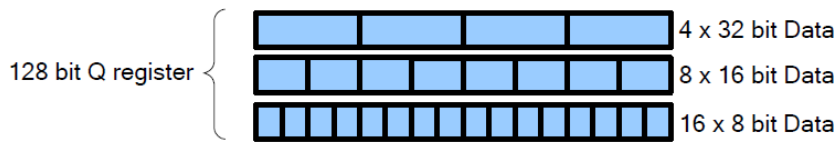


Register Mapping

Advanced SIMD and Floating-point register mapping

Figure A2-1 shows the different views of Advanced SIMD and Floating-point register banks, and the relationship between them.

- NEON Advanced SIMD and VFP use the same register set



S0-S31 VFP only	D0-D15 VFPv2, VFPv3-D16, or VFPv4-D16	D0-D31 VFPv3-D32, VFPv4-D32, or Advanced SIMD	Q0-Q15 Advanced SIMD only
S0	D0	D0	
S1			Q0
S2	D1	D1	
S3			
S4	D2	D2	
S5			Q1
S6	D3	D3	
S7			
⋮			
S28	D14	D14	Q7
S29			
S30	D15	D15	
S31			
		D16	Q8
		D17	
		⋮	
		D30	Q15
		D31	

The mapping between the registers is as follows:

- $S_{\langle 2n \rangle}$ maps to the least significant half of $D_{\langle n \rangle}$
- $S_{\langle 2n+1 \rangle}$ maps to the most significant half of $D_{\langle n \rangle}$
- $D_{\langle 2n \rangle}$ maps to the least significant half of $Q_{\langle n \rangle}$
- $D_{\langle 2n+1 \rangle}$ maps to the most significant half of $Q_{\langle n \rangle}$.

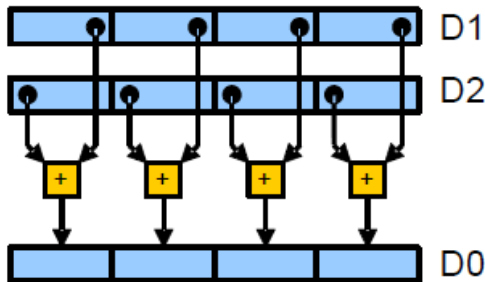
Figure A2-1 Advanced SIMD and Floating-point Extensions register set

For example, software can access the least significant half of the elements of a vector in Q6 by referring to D12, and the most significant half of the elements by referring to D13.

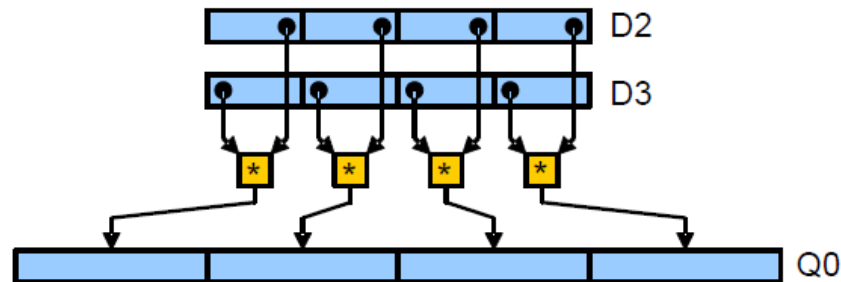
NEON

- Advanced SIMD (Single Instruction Multiple Data)
- It supports 8, 16, 32 and 64-bit integer and single-precision (32-bit) floating point data
- Up to 16 operations at the same time
 - $1\text{B} \times 16 = 16\text{B}$ (= 1 quad word)

VADD.I16 D0, D1, D2



VMUL.I32.S16 Q0, D2, D3; This instruction will cause four widening multiplies



VFP (Vector Floating Point)

- FPU (Floating Point Unit) coprocessor extension to ARM architecture
- Single-precision and double-precision FP computation
 - Compliant with IEEE 754-1985
- Intended to support execution of short “vector mode” instructions, but operated on “each” vector element **sequentially**
 - Thus, did not offer the performance of true SIMD
 - This vector mode was thus removed shortly after its introduction, to be replaced with the much more powerful NEON Advanced SIMD

ARM Processor Selector

ARM Processor	Architecture	Performance DMIPS/MHz	ARM instructions	Thumb-2 instructions	Jazelle-DBX JAVA bytecode execution	Jazelle-RCT Dynamic compiler support	TrustZone security	E' DSP extensions	Media SIMD extensions	NEON SIMD extensions	Floating point	Caches	Memory Management Unit (MMU)	Memory Protection Unit (MPU)	Hardware Cache coherency	Target OS	Trace support
ARM7TDMI/ARM7TDMI-S	ARMv4-T	0.95	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	Real Time	✓
ARM946E-S	ARMv5-E	1.23	✓	✗	✗	✗	✗	✓	✗	✗	Optional	✓	✗	✓	✗	Real Time	✓
ARM926EJ-S	ARMv5-EJ	1.06	✓	✗	✓	✗	✗	✓	✗	✗	Optional	✓	✓	✗	✗	Platform	✓
ARM1136J-S	ARMv6	1.18	✓	✗	✓	✗	✗	✓	✓	✗	Optional	✓	✓	✗	✗	Platform	✓
ARM1156T2-S	ARMv6-T2	1.45	✓	✓	✗	✗	✗	✓	✓	✗	Optional	✓	✗	✓	✗	Real Time	✓
ARM1176JZ-S	ARMv6-Z	1.26	✓	✗	✓	✗	✓	✓	✓	✗	Optional	✓	✓	✗	✗	Platform	✓
ARM11 MPCore	ARMv6	1.25	✓	✗	✓	✗	✗	✓	✓	✗	Optional	✓	✓	✗	✓	Platform/SMP	✓
Cortex-M0+	ARMv6-M	0.90	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	Real Time	✗
Cortex-M0	ARMv6-M	0.90	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	Real Time	✗
Cortex-M1	ARMv6-M	0.79	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	Real Time	✗
Cortex-M3	ARMv7-M	1.25	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	Optional	✗	Real Time	Instruction only
Cortex-M4	ARMv7-ME	1.25	✓	✓	✗	✗	✗	✓	✓	✗	Optional	✓	✗	Optional	✗	Real Time	✓
Cortex-A5 MPCore	ARMv7+MP	1.58	✓	✓	✓	✓	✓	✓	✓	Optional	Optional	✓	✓	✗	✓+ACP	Platform/SMP	✓
Cortex-R4	ARMv7	1.66	✓	✓	✗	✗	✗	✓	✓	✗	Optional	✓	✗	Optional	✗	Real Time	✓
Cortex-R5	ARMv7	1.66	✓	✓	✗	✗	✗	✓	✓	✗	Optional	✓	✗	Optional	✗	Real Time	✓
Cortex-R7	ARMv7	2.53	✓	✓	✗	✗	✗	✓	✓	✗	Optional	✓	✗	Optional	✗	Real Time	✓
Cortex-A7	ARMv7+MP	1.90	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓+ACP	Platform/SMP	PTM
Cortex-A8	ARMv7	2.07	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	Platform	✓
Cortex-A9 MPCore	ARMv7+MP	2.50	✓	✓	✓	✓	✓	✓	✓	Optional	Optional	✓	✓	✗	✓+ACP	Platform/SMP	PTM
Cortex-A15 MPCore	ARMv7+MP	2.50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓+ACP	Platform/SMP	PTM
Cortex-A53	ARMv8	2.3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓+ACP	Platform/SMP	PTM
Cortex-A57	ARMv8	>4.0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓+ACP	Platform/SMP	PTM