ARM Roadmap
Spring 2017

ARM

Robert Boys
bob.boys@arm.com

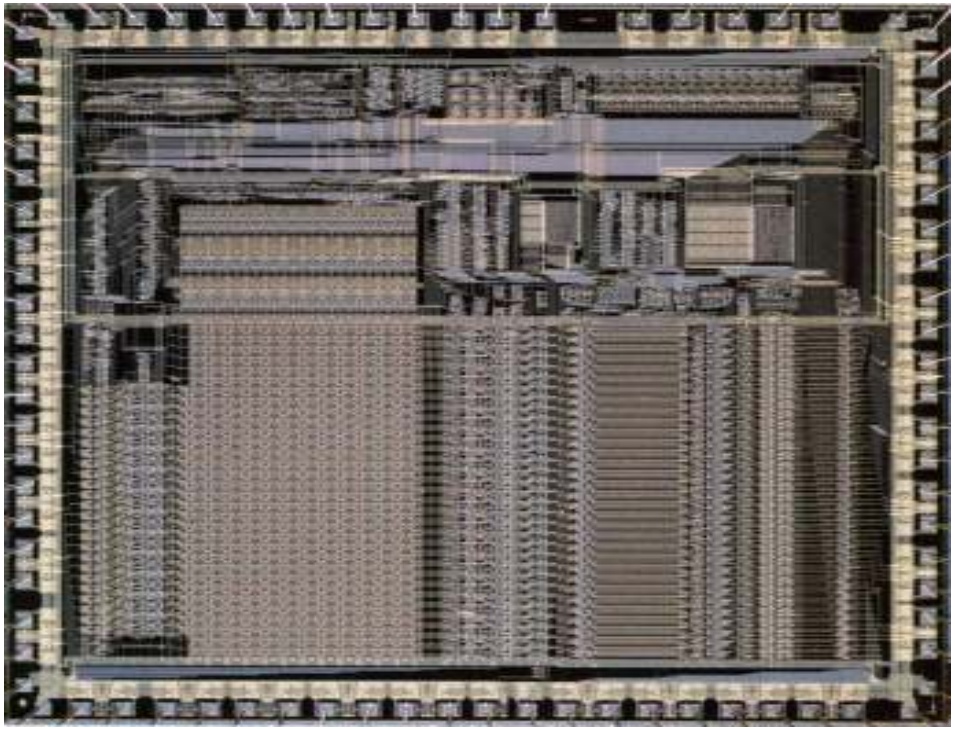
Version 9.0

SoftBank

Agenda

- Roadmap
- Architectures
- Issues
- What is NEW !
  - big.LITTLE™
  - 64 Bit
  - Cortex®-A15
  - 64 BIT
  - DynamIQ

ARM1™ die



In the Beginning…

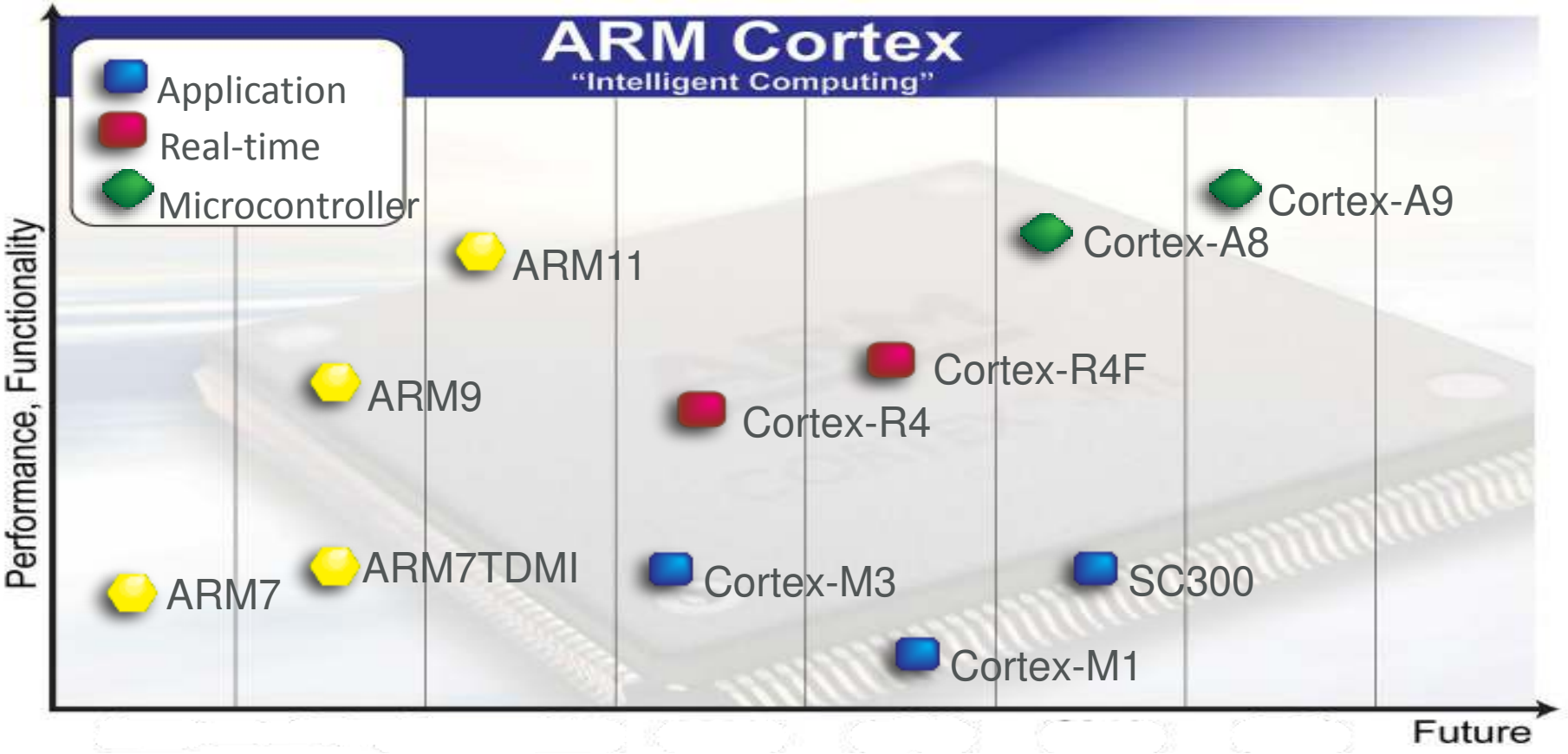
- 1985 32 years ago in a barn....
- 12 engineers
- Cash from Apple and VLSI
- IP from Acorn Computers
- Proof of concept
- No patents, no independent customers, product not ready for mass market.
- A barn, some energy, experience and belief:

  "We're going to be the Global Standard"

SoftBank

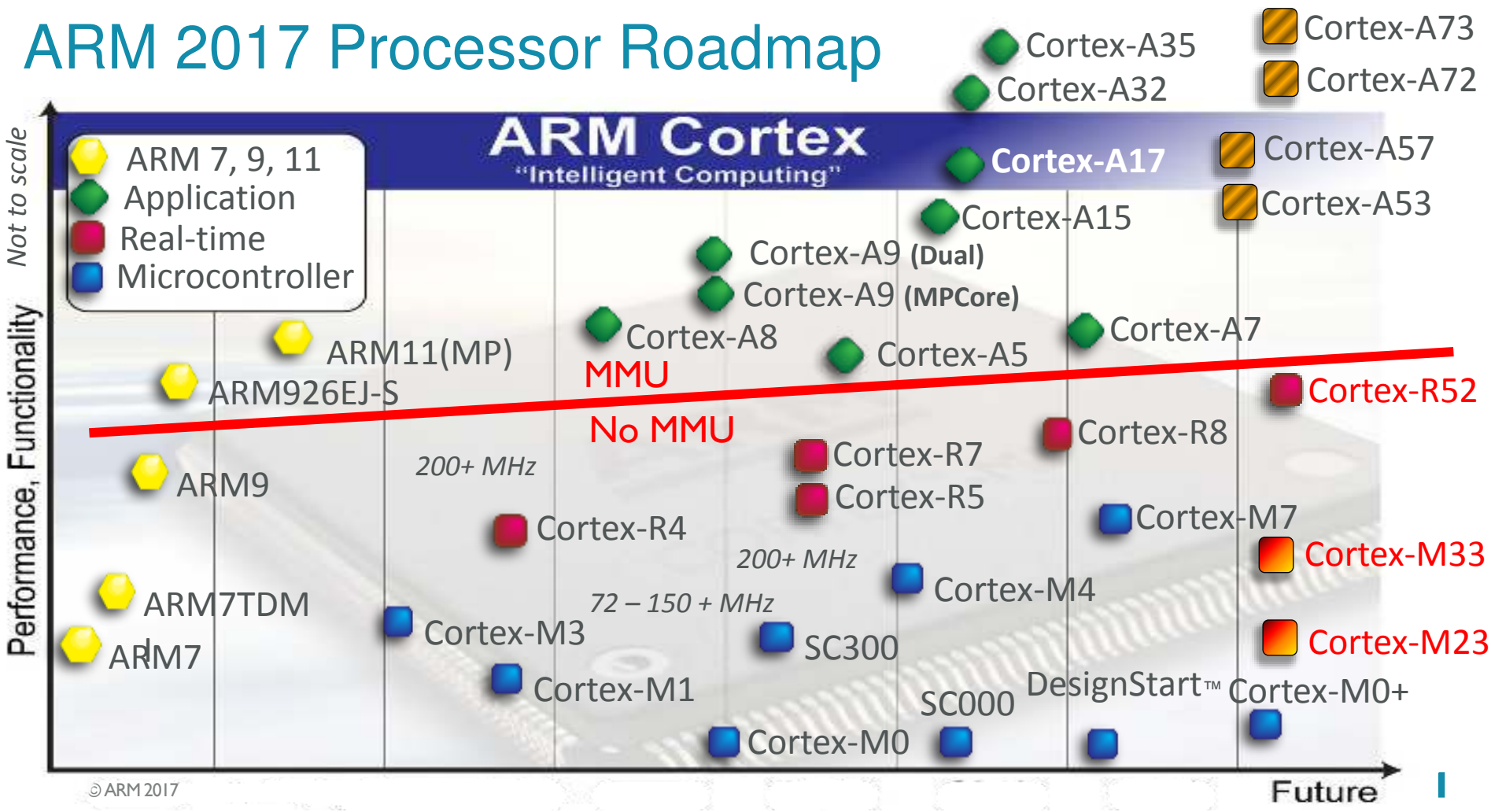
ARM

The Cortex Processor Roadmap in 2008



ARM

ARM 2017 Processor Roadmap



Versions, cores and architectures ?

Family	Architecture	Cores
ARM7TDMI	ARMv4T	ARM7TDMI(S)
ARM9 ARM9E	ARMv5TE	ARM926EJ-S, ARM966E-S
ARM11	ARMv6 (T2)	ARM1136(F), 1156T2(F)-S, 1176JZ(F), ARM11 MPCore™
Cortex-A **Cortex-R** **Cortex-M**	ARMv7-A ARMv7-R ARMv7-M ARMv6-M	Cortex-A5, A7, A8, A9, A12, A15, A17 Cortex-R4(F), Cortex-R5, R7, R8 … Cortex-M3, M4, M7 (M7 is ARMv7-ME) Cortex-M1, M0, M0+
NEW !	ARMv8-A	64 Bit: Cortex-A35/A53/57/A72 Cortex-A73 Cortex-A32
NEW !	ARMv8-R	32 Bit: Cortex-R52
NEW !	ARMv8-M	32 Bit: Cortex-M23 & M33 TrustZone®

What is New ? DynamIQ !

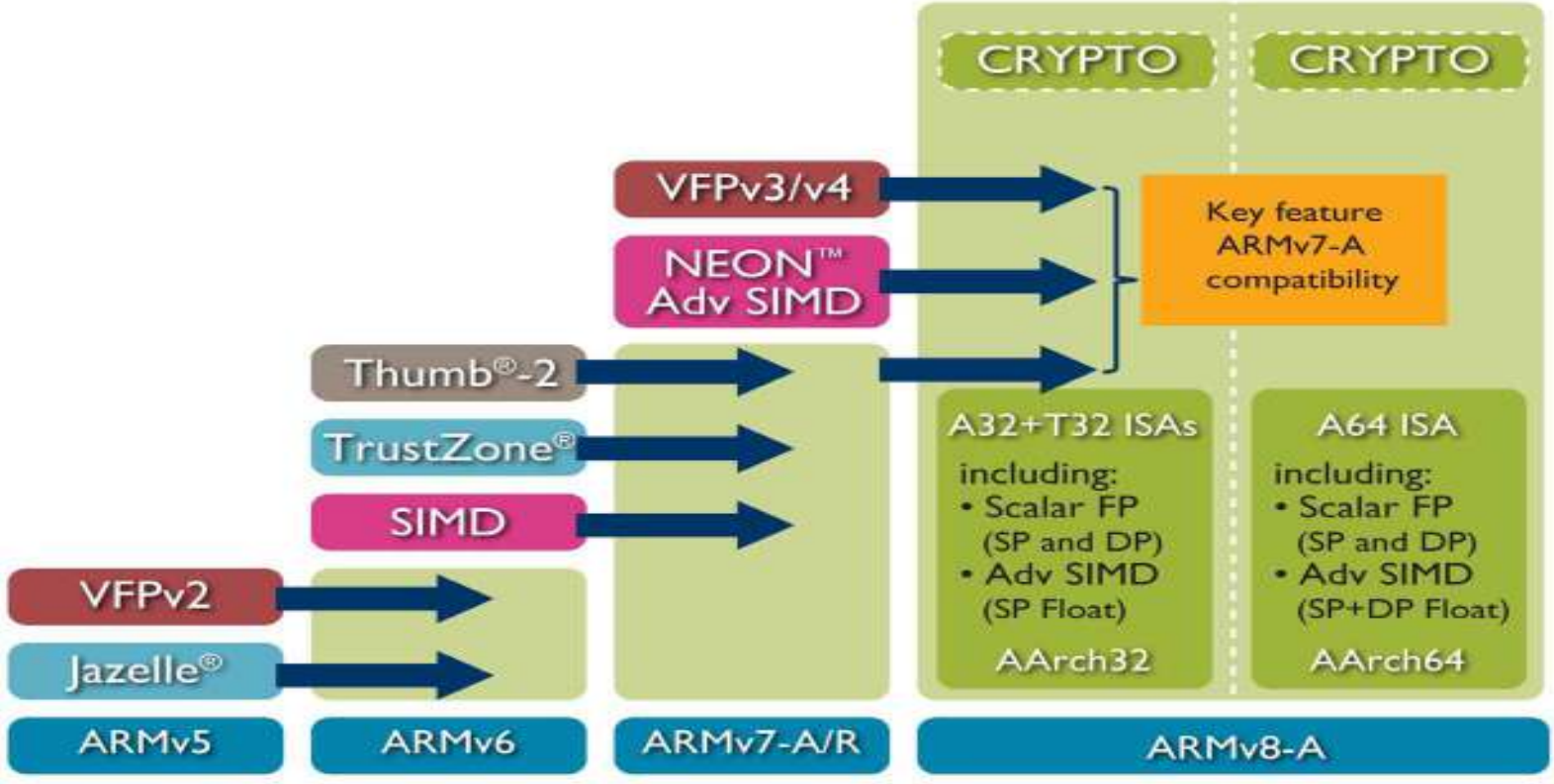
- AI (Artificial Intelligence), Machine Learning, autonomous vehicles
- For multi-core Cortex-A processors.
- Up to 8 cores on a single cluster. Can be different processors.
- Such as 1+3, 1+7 – different configurations possible.
- Makes big.LITTLE better. 2011. Is the next evolutionary step.
- New memory subsystems.
- ASIL D certification possible for the 1st time.
- This is the future for ARM multi-core Cortex-A…
- Search for ARM DynamIQ for more information.
- One more thing…..

Processor Licenses (list is no longer complete)

- **ARMv8-A:** **27 !** AMD, Broadcom, Huawei (HiSilicon), STMicroelectronics, Samsung, MediaTek, Huawei, Altera(Intel), Qualcomm and Rockchip, NXP, Xilinx…
- **Cortex-A15:** Broadcom, HiSilicon, Texas Instruments, Samsung, nVIDIA
- **Cortex-A9:** Xilinx, Altera, NEC, nVIDIA, STMicroelectronics, Toshiba, Broadcom Corporation, NXP, NEC, Texas Instruments, Toshiba, Mindspeed Technologies, ZiLABS, Open-Silicon, eSilicon
- **Cortex-A8:** Broadcom Corporation, NXP, Panasonic, Samsung, STMicroelectronics, Texas Instruments, PMC-Sierra, Matsushita
- **Cortex-A7:** Broadcom, NXP, Fujitsu, HiSilicon, LGE, Samsung, STEricsson, Texas Instruments
- **Cortex-A5:** AMD, Atmel, NXP, Cambridge Silicon Radio, Open-Silicon, eSilicon
- **Cortex-R:** Broadcom, Texas Instruments, Toshiba, Infineon, Open-Silicon, eSilicon, Samsung, Marvell, LSI, Fujitsu, Cypress (Spansion)
- ***NEW !*** **Cortex-M7:** NXP, Atmel, ST.
- **Cortex-M4:** NXP, Atmel, ST, Texas Instruments, Open-Silicon, eSilicon, Spansion, Ambiq
- **Cortex-M3:** Microsemi (Actel), Broadcom, Energy Micro, NXP, ST, TI, Toshiba, Zilog, Accent Srl, Broadcom Corporation, Cypress Semiconductor, Ember, Fuzhou Rockchip Electronics CO. Ltd., Open-Silicon, eSilicon, Spansion (Fujitsu)
- **Cortex-M0:** Austriamicrosystems, Chungbuk Technopark, NXP, Triad Semiconductor, Melfas, Open-Silicon, eSilicon, Cypress, Infineon, Nuvoton, STMicroelectronics
- **Cortex-M0+:** NXP, Atmel, Cypress (Spansion), Silicon Labs

Total: Cortex-A 178, **Cortex-R** 45, **Cortex-M** 240, **ARM7** 172, **ARM9** 271, **ARM11** 82

Feature Set of Various ARM Processors



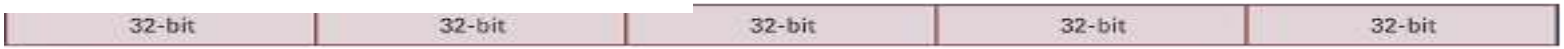
- ARM 9 ARM 11 Cortex-M,R,A 64 Bit

ARM

Instruction Sets

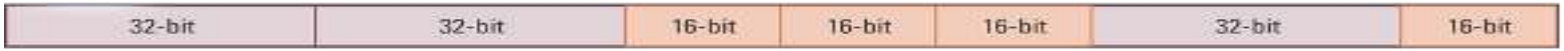
- ARM (32 bit) now referred as AArch32
- Thumb® (16 bit)
- Thumb2: Cortex-Mx processors. Cortex-R, A have Thumb2 + ARM.
- A64 (64 bit) referred as AArch64

ARM now called AArch32

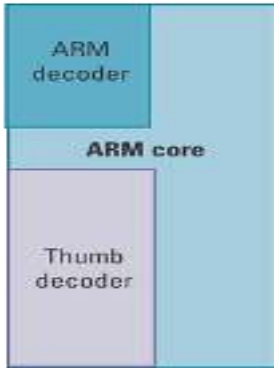


Thumb (actually includes all ARM 32 bit instructions)

Thumb-2



A64 AArch64



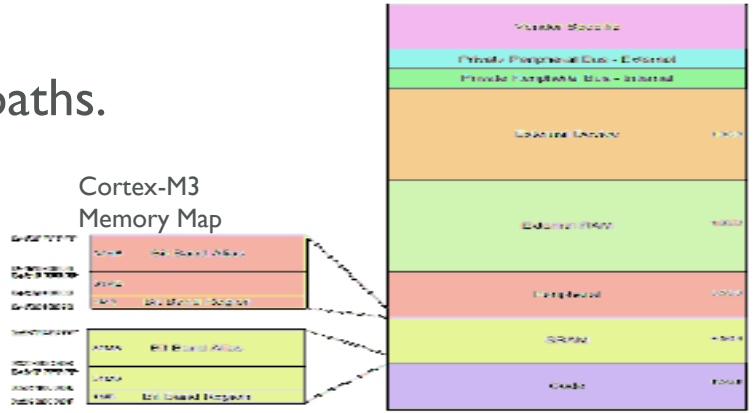
ARM

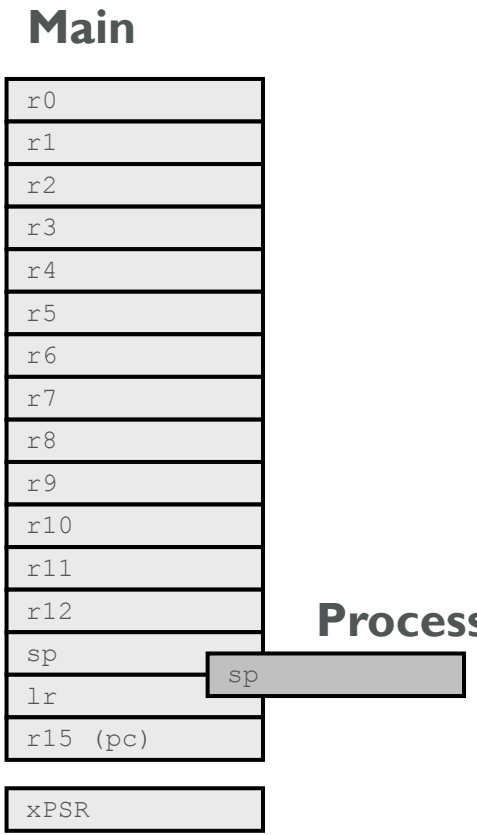

VABS
VADD
VCMP
VCMPE
VCVT
VCVTR
VCVTB
VCVTT
VDIV
VFNA
VFMS
VFNMA
VFNMS
VLDM
VLDR
VMLA
VMLS
VMOV
VMRS
VMSR
VMUL
VNEG
VNMLA
VNMLS
VNMUL
VPOP
VPUSH
VSQRT
VSTM
VSTR
VSUB
VCVTA
VCVTN
VCVTP
VCVTM
VMAXNM
VMINNM
VRINTA
VRINTN
VRINTP
VRINTM
VRINTX
VRINTZ
VRINTR
VSEL
PKHBT
PKHTB
QADD
QADD16
QADD8
QASX
QDADD
QDSUB
QSAX
QSUB
QSUB16
QSUB8
SADD16
SADD8
SASX
SEL
SHADD16
SHADD8
SHASX
SHSAX
SHSUB16
SHSUB8
SMLABB
SMLABT
SMLALD
SMLALDX
SMLAWB
SMLAWT
SMLSD
SMLSDX
SMLSLD
SMLSLDX
SMMLA
SMMLAR
SMMLS
SMMLSR
SMMUL
SMMULR
SMUAD
SMUADX
SMULBB
SMULBT
SMULTB
SMULTT
SMULWB
SMULWT
SMUSD
SMUSDX
SSAT16
SSAX
SSUB16
SSUB8
SXTAB
SXTAB16
SXTAH
UADD16
UADD8
UASX
UHADD16
UHSUB8
UMAAL
UQADD16
UQADD8
UQASX
UQSAX
UQSUB16
UQSUB8
USAD8
USADA8
USAT16
USAX
USUB16
USUB8
UXTAB
UXTAB16
UXTAH
UXTB16
ADC
ADD
ADR
AND
ASR
B
BFC
BFI
BIC
CBNZ
CBZ
CDP
CDP2
CLREX
CLZ
CMN
CMP
DBG
EOR
LDC
LDC2
LDMIA
LDMDB
LDR
LDRB
LDRBT
LDRD
LDREX
LDREXB
LDREXH
LDRH
LDRHT
LDRSB
LDRSBT
LDRSH
LDRT
LSL
LSR
MCR
MCR2
MCRR
MCRR2
MLA
MLS
MOV
MOVT
MRC
MRC2
MRRC
MRRC2
MUL
MVN
NOP
ORN
ORR
PLD
PLI
POP
PUSH
RBIT
REV
REV16
REVSH
ROR
RRX
RSB
SBC
SBFX
SDIV
SEV
SMLAL
SMULL
SSAT
STC
STC2
STMIA
STMDB
STR
STRB
STRBT
STRD
STREX
STREXB
STREXH
STRH
STRHT
STRT
SUB
SXTB
SXTH
TBB
TBH
TEQ
TST
UBFX
UDF
UDIV
UMLAL
UMULL
USAT
UXTB
UXTH
WFE
WFI
YIELD
IT
ADC
ADD
ADR
AND
ASR
B
BIC
BKPT
BL
BLX
BX
CBN
CMP
CPS
DMB
EOR
DSB
ISB
LDMIA
LDR
LDRB
LDRH
LDRSB
LDRSH
LSL
LSR
MOV
MRS
MSR
MUL
MVN
NOP
ORR
POP
PUSH
REV
REV16
REVSH
ROR
RSB
SBC
SEV
STMIA
STR
STRB
STRH
SUB
SVC
SXTB
SXTH
TST
UDF
UXTB
UXTH
WFE
WFI
YIELD
Cortex-M0+
Cortex-M3
Cortex-M4
Cortex-M7
Cortex-M4
FPU
Cortex-M7
FPU

ARM

Cortex-M Adds:

- Simpler register set.
- More interrupts and exceptions. NVIC
- More breakpoints (to 6)
- 4 Watchpoints.
- SYSTICK timer for RTOSs.
- Fixed address space.
- Bit manipulation.
- Upgrade and downgrade paths.





ARM

Cortex-M3 Exceptions

Exception handling order is defined by programmable priority

- Reset, Non Maskable Interrupt (NMI) and Hard Fault have predefined pre-emption.
- NVIC catches exceptions and pre-empts current task based on priority
- Program Counter set to exception address in vector table which directs to handler code

Fault Mode & Start-up Handlers

System Handlers

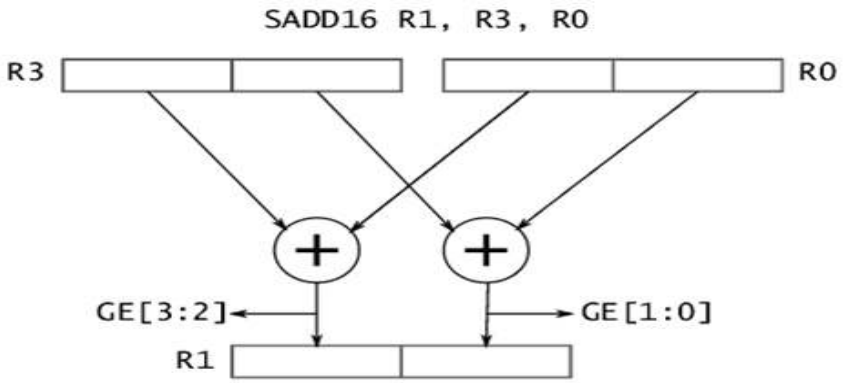
Custom Handlers

Exception	Name	Priority	Descriptions
1	Reset	-3 (Highest)	Reset
2	NMI	-2	Non-Maskable Interrupt
3	Hard Fault	-1	Default fault if other hander not implemented
4	MemManage Fault	Programmable	MPU violation or access to illegal locations
5	Bus Fault	Programmable	Fault if AHB interface receives error
6	Usage Fault	Programmable	Exceptions due to program errors
14	PendSV	Programmable	Pendable SerVice request for System Device
16	Interrupt #0	Programmable	External Interrupt #0
...	...	...	...
...	...	...	...
...	...	...	...
255	Interrupt #239	Programmable	External Interrupt #239

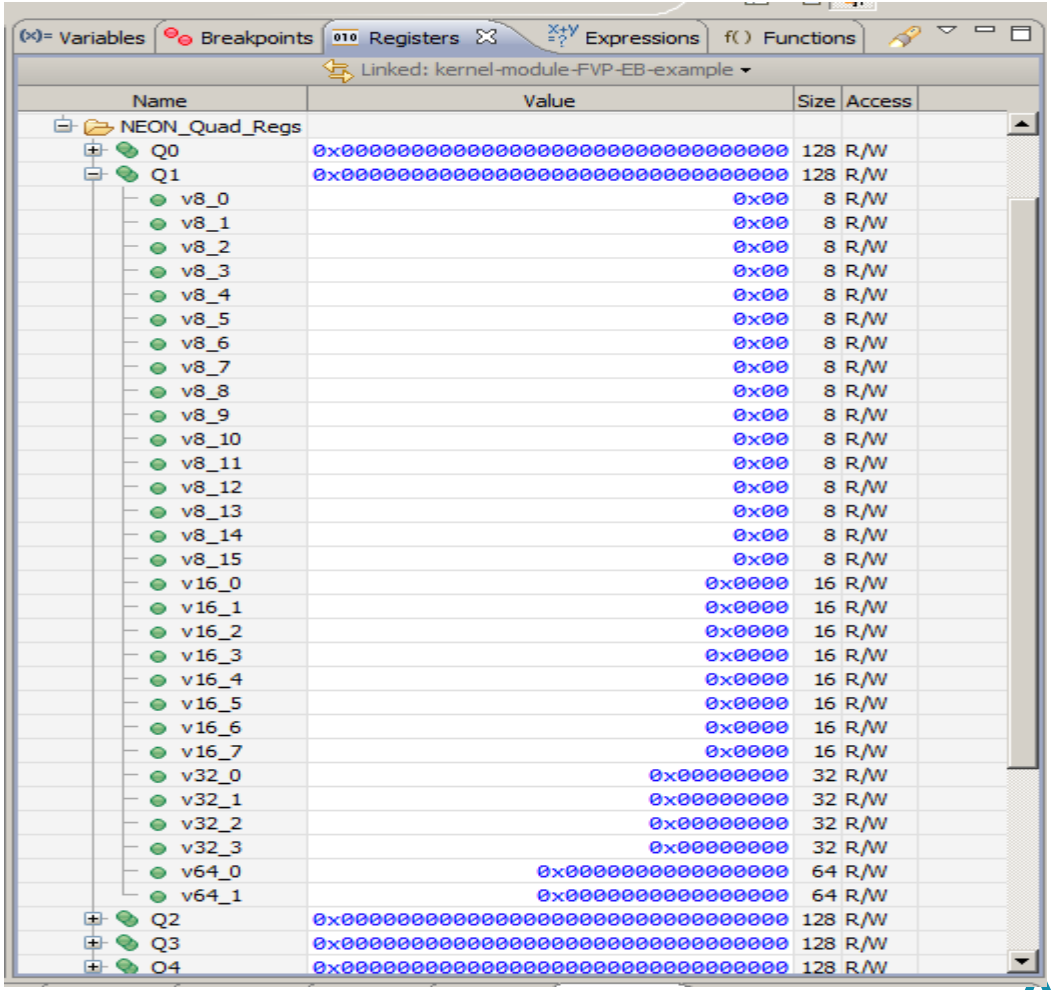
SIMD – Single Instruction Multiple Data

- **SIMD is a set of instructions that can operate on multiple data sets contained in a register.**
- Registers used are the 32 bit general purpose R in ARM.
- Some of the SIMD instructions start with a "S" (signed) or a "U" (unsigned) and with a suffix denoting the size of the operand (116). An example is UADD116.
- Here is the SADD16 instruction:

Two sets of 16 bit operands are taken from 32 bit registers R3 and R0 and are added together (signed) and the 16 bit result stored in the 32 bit register R1:

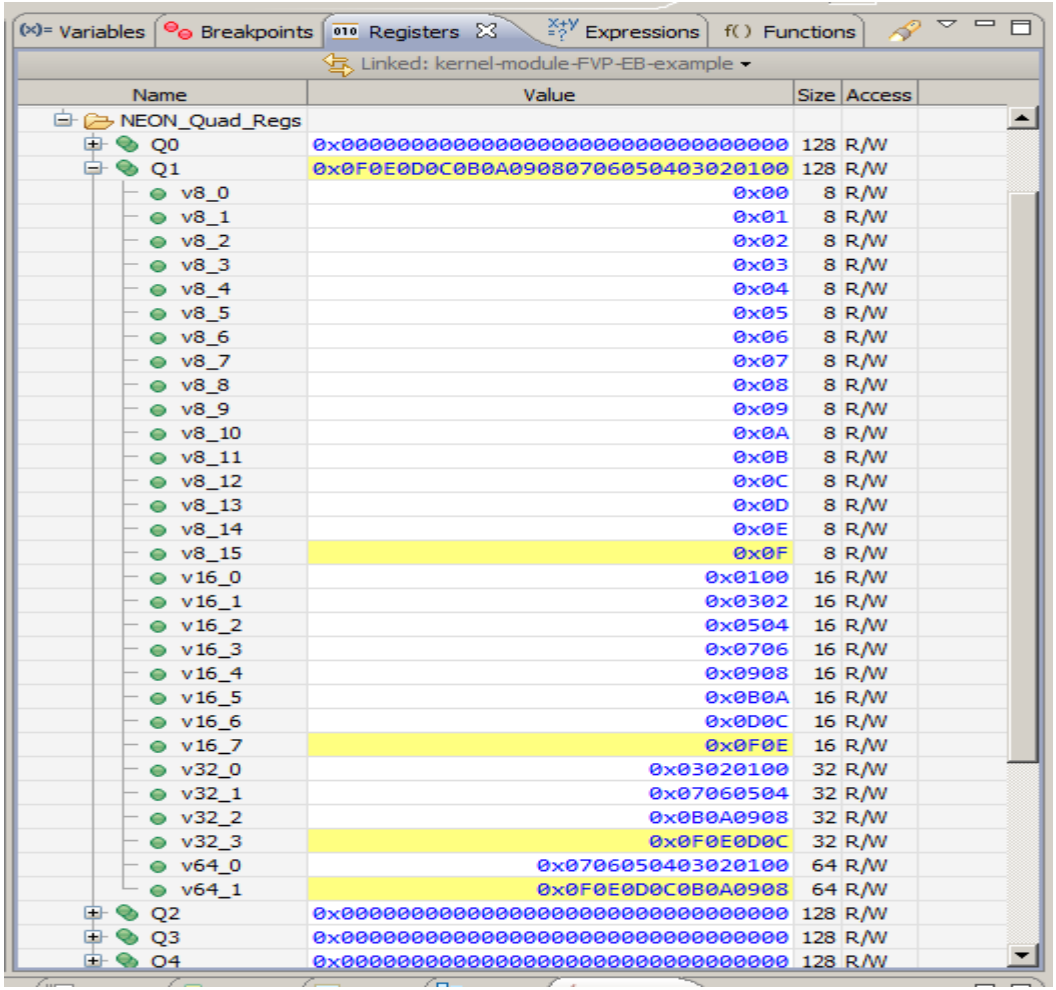


NEON™:



Name	Value	Size	Access
NEON_Quad_Regs			
Q0	0x00000000000000000000000000000000	128	R/W
Q1	0x00000000000000000000000000000000	128	R/W
v8_0	0x00	8	R/W
v8_1	0x00	8	R/W
v8_2	0x00	8	R/W
v8_3	0x00	8	R/W
v8_4	0x00	8	R/W
v8_5	0x00	8	R/W
v8_6	0x00	8	R/W
v8_7	0x00	8	R/W
v8_8	0x00	8	R/W
v8_9	0x00	8	R/W
v8_10	0x00	8	R/W
v8_11	0x00	8	R/W
v8_12	0x00	8	R/W
v8_13	0x00	8	R/W
v8_14	0x00	8	R/W
v8_15	0x00	8	R/W
v16_0	0x0000	16	R/W
v16_1	0x0000	16	R/W
v16_2	0x0000	16	R/W
v16_3	0x0000	16	R/W
v16_4	0x0000	16	R/W
v16_5	0x0000	16	R/W
v16_6	0x0000	16	R/W
v16_7	0x0000	16	R/W
v32_0	0x00000000	32	R/W
v32_1	0x00000000	32	R/W
v32_2	0x00000000	32	R/W
v32_3	0x00000000	32	R/W
v64_0	0x0000000000000000	64	R/W
v64_1	0x0000000000000000	64	R/W
Q2	0x00000000000000000000000000000000	128	R/W
Q3	0x00000000000000000000000000000000	128	R/W
Q4	0x00000000000000000000000000000000	128	R/W

NEON:



Name	Value	Size	Access
NEON_Quad_Regs			
Q0	0x00000000000000000000000000000000	128	R/W
Q1	0x0F0E0D0C0B0A09080706050403020100	128	R/W
v8_0	0x00	8	R/W
v8_1	0x01	8	R/W
v8_2	0x02	8	R/W
v8_3	0x03	8	R/W
v8_4	0x04	8	R/W
v8_5	0x05	8	R/W
v8_6	0x06	8	R/W
v8_7	0x07	8	R/W
v8_8	0x08	8	R/W
v8_9	0x09	8	R/W
v8_10	0x0A	8	R/W
v8_11	0x0B	8	R/W
v8_12	0x0C	8	R/W
v8_13	0x0D	8	R/W
v8_14	0x0E	8	R/W
v8_15	0x0F	8	R/W
v16_0	0x0100	16	R/W
v16_1	0x0302	16	R/W
v16_2	0x0504	16	R/W
v16_3	0x0706	16	R/W
v16_4	0x0908	16	R/W
v16_5	0x0B0A	16	R/W
v16_6	0x0D0C	16	R/W
v16_7	0x0F0E	16	R/W
v32_0	0x03020100	32	R/W
v32_1	0x07060504	32	R/W
v32_2	0x0B0A0908	32	R/W
v32_3	0x0F0E0D0C	32	R/W
v64_0	0x0706050403020100	64	R/W
v64_1	0x0F0E0D0C0B0A0908	64	R/W
Q2	0x00000000000000000000000000000000	128	R/W
Q3	0x00000000000000000000000000000000	128	R/W
Q4	0x00000000000000000000000000000000	128	R/W

ARM

NEON: is also called Advanced SIMD:

- **NEON is both a) another set of more instructions that operate on b) 32 special 64 bit registers.**
- NEON works on a 128-bit data path. So on the Cortex-A9, it normally uses two of the 64-bit NEON registers for each of the operands and puts the result back in one or two of them.
- NEON share many instructions with a FPU…and each have their own instructions.
- NEON instructions begin with a "V".
- VADD, VABS, VCGE, VCGT, VEOR, VQADD…
- See ARM appnote DHT0002A "Introducing Neon".
- Cortex-A8, A9, A15, A64, some ARM11 devices.

How to program SIMD and NEON:

- Programs are written using assembly language.
- You can also use Intrinsics.
- The compiler can also use automatic vectorization on C or C++ code. (you have to tell the compiler it can do this)
- NEON libraries: This is a good way to do this to avoid writing in assembly. OpenMax and Ne10 are two. The executable code will use SIMD and NEON whenever possible.
- See DHT0004A for more information.
- At this time, all ARM Cortex-A series processors have NEON.
- SIMD and NEON used in Video encode/decode, 2D/3D graphics, audio processing…
- Cortex-A8, A9, A15, A64, some ARM11 devices.

big.LITTLE™

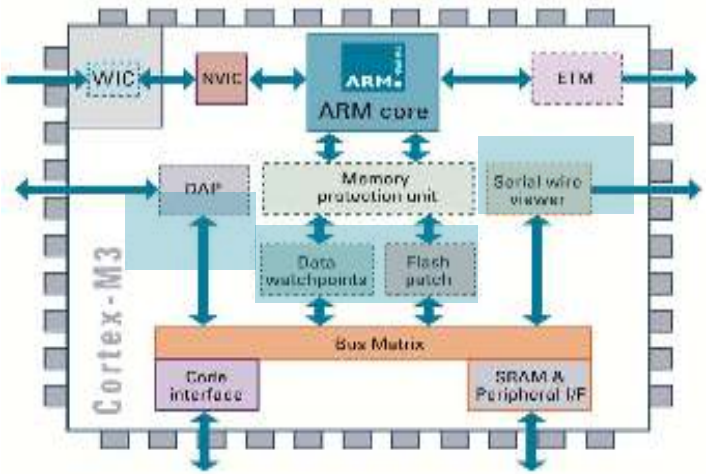
- Two or more processers – share workload
- Processors not always the same.
- Especially helpful for power savings.
- Cortex-A15 and Cortex-A7 are the original.
- Freescale: A5 + M4
- NXP LPC4300 M4 + M0
- Samsung announced a 8 ARM core 5 Octa Processor
- Plenty of others now and more coming.

ARM 64 Bit Architecture

- The latest from ARM….
- Instructions 32 bits.
- 31 64 bit registers + PC + SP
- Two execution states: AArch32 and AArch64
- 48 bit addresses to outside world on first chips.
- Microsoft: Windows 8 and RT will work on ARM
- NVIDIA demonstrated Windows 8 on Tegra 3 at CES.
- www.arm.com/files/downloads/ARMv8_Architecture.pdf
- http://www.linaro.org/engineering/armv8

CoreSight Debug & Trace

- **CoreSight debug technology delivers enhanced debugging modes and features**
- **Serial Wire Debug (SWD) Mode**
  - 2 wire interface instead of 4 or 5. Most have JTAG too.
- **Serial Wire Viewer (SWV)**
  - Data R/W, Exceptions, PC Samples
- **Enhanced Trace Macrocell (ETM)**
  - Adds all the program counters.
  - Provides Code Coverage, Timings, Performance Analysis, crash reports.
- **Program Trace Macrocell (PTM)**
  - Cortex-A9, A15. All program counters.
- **Embedded Trace Buffer: small 4 – 8K trace buffer.**



ARM

Operating Systems

- ***Linux***
- Windows, WinCE
- Android
- Micrium, ExpressLogic, Quadros, QNX and so on.
- Big differentiator between processors is the MMU.
- Linaro – an open source community supported by ARM.
- Keil RTX has BSD or Apache 2.0 license now – means this is free.
- CMSIS-Core – header files and startup files
- CMSIS-DSP – DSP libraries
- CMSIS-RTOS – for the RTOS market.
- CMSIS-Packs – distribution of processor files and examples.

ARM

World's No. 1 Embedded Ecosystem

Thank you…