

ARM NEON

Pocket Supercomputers



Leonardo Giovanni Prati - 8300079

Gabriela Rodrigues do Prado Rossales - 6605109

Arm Holdings

- Fundada em 27/11/1990 em Cambridge, Inglaterra
- Desenvolve:
 - Microprocessadores
 - *Software*
 - Sistemas Embarcados

ARM NEON

- Tecnologia que utiliza um versão estendida da arquitetura SIMD
- Usada no Arm Cortex-A e Cortex-R52
- Introduzida ao Armv7-A e Armv7-R
- Atualmente também é usada no Armv8-A e Armv8-R
- Objetivo: melhorar a experiência multimídia do usuário

ARM NEON

- Acelerar:
 - Codificação/decodificação e processamento de áudio e vídeo
 - Interface de usuário
 - Gráficos 2D/3D ou jogos
 - Reconhecimento de facial e de voz
 - Visão computacional
 - *Deep learning*

ARM: Advanced RISC Machines

- Família de Arquiteturas RISC
- Conjunto de instruções reduzido
 - Menos Transistores
 - Menor custo
 - Menor consumo de energia
 - Menor aquecimento

ARM: Advanced RISC Machines

- Ótimo para Dispositivos *Mobile* ^[9]
- Ótimo para Sistemas Embarcados
- deteve 90% do valor do Mercado *Mobile* em 2016^[11]

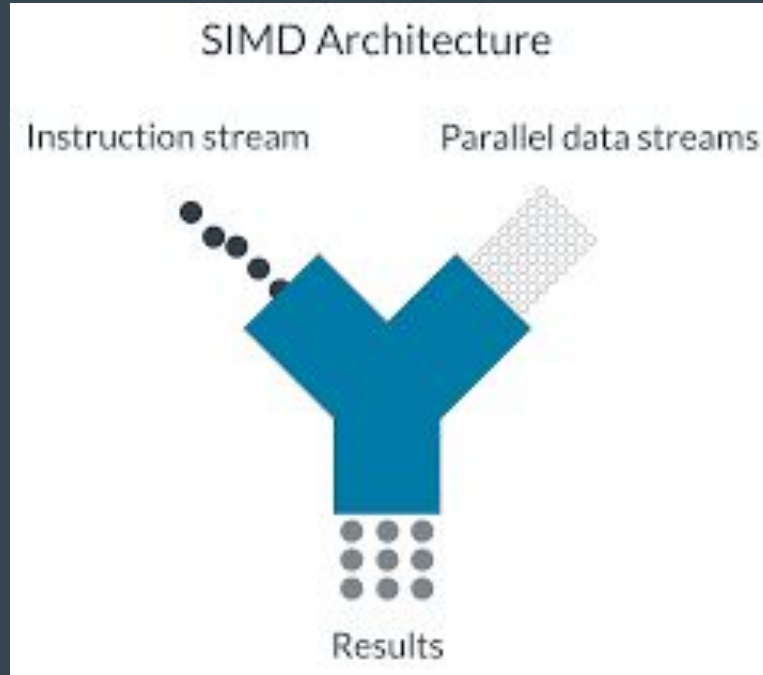
ARM: Advanced RISC Machines

- *Hardware:*
 - Coprocessador SIMD
- ISAE(*Instruction Set Architecture Extension*)
- *Software:*
 - Otimização via compilador
 - Bibliotecas *open source*
- Conjunto chamado de MPE (*Media Processing Engine*)

Hardware: SIMD

- **Arquitetura SIMD** (*Single instruction, multiple data*)
 - **Múltiplo processamento**
 - **Somente uma instrução**

Hardware: SIMD

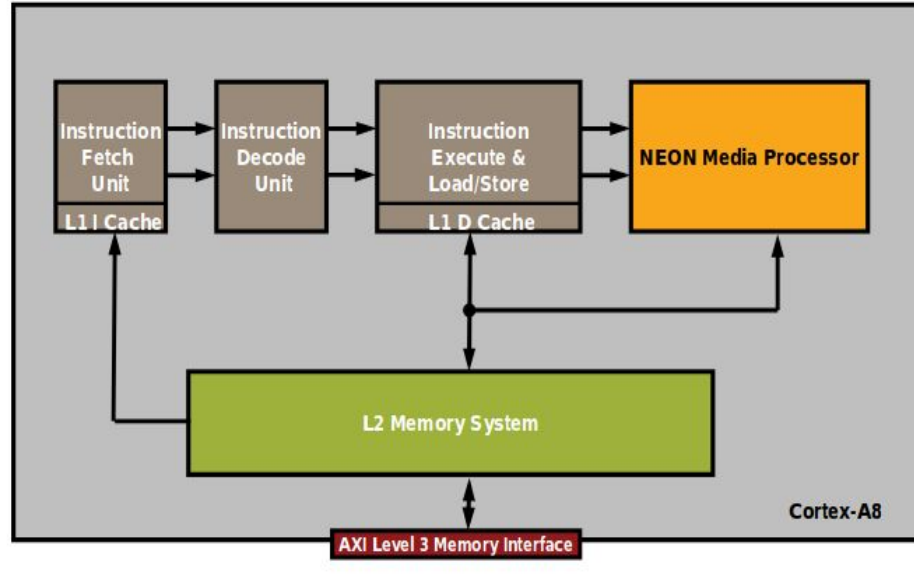


Hardware: Coprocessador

- Estende o processador principal
- Possui decodificação de instrução própria
- Possui *pipeline* próprio

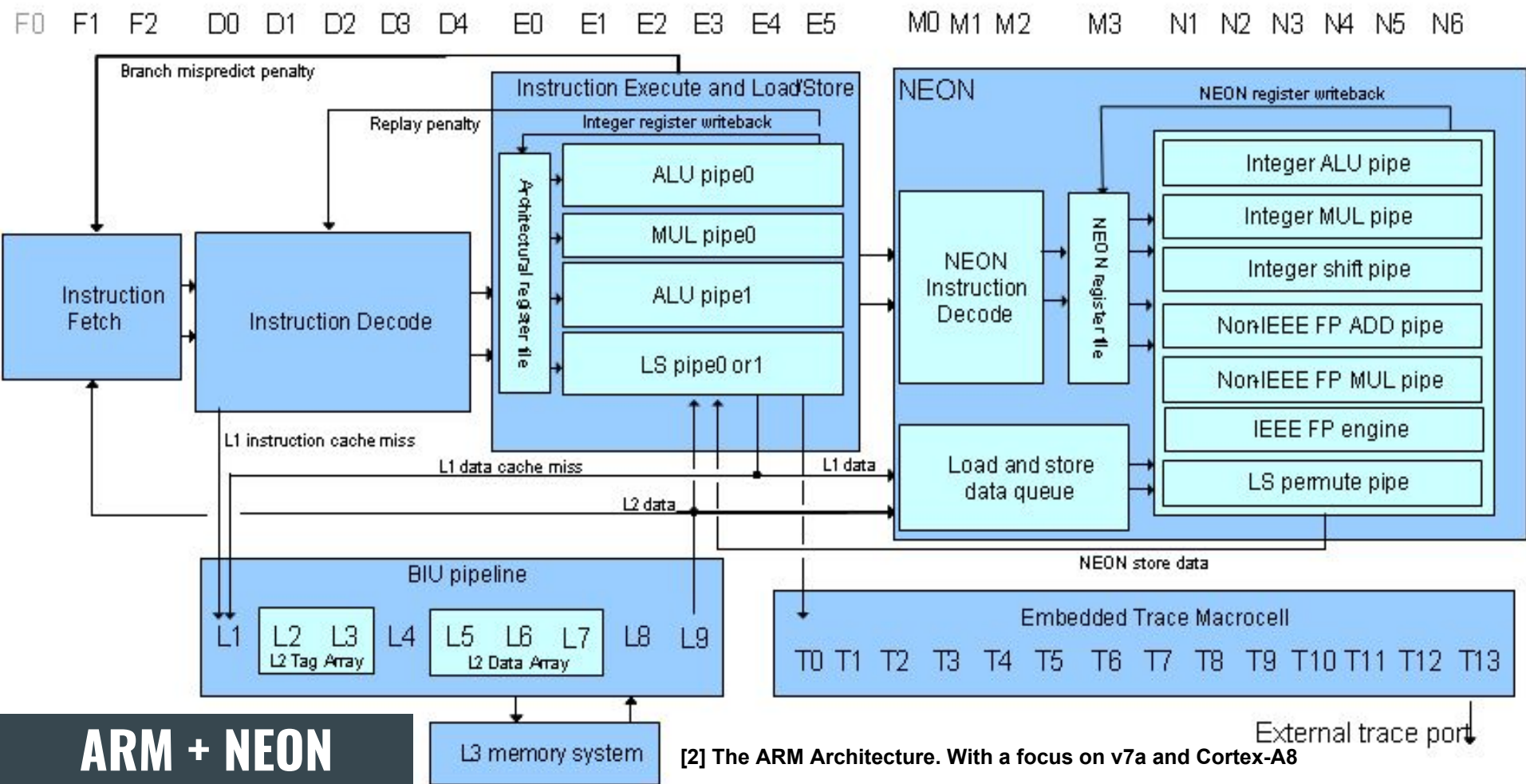
Hardware: Coprocessador

Cortex-A8 Block Diagram



13-Stage Integer Pipeline

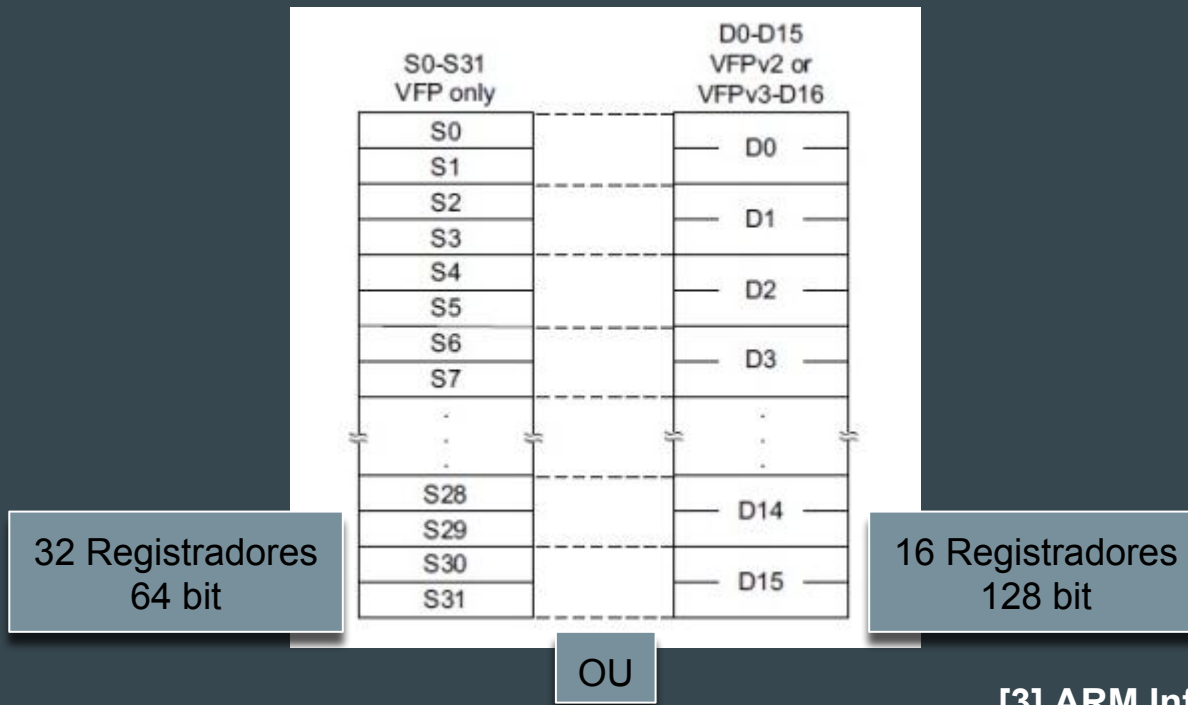
10-Stage NEON Pipeline



Hardware: Processador Vetorial

- Um banco de registradores
- Múltiplas interpretações

Hardware: Processador Vetorial



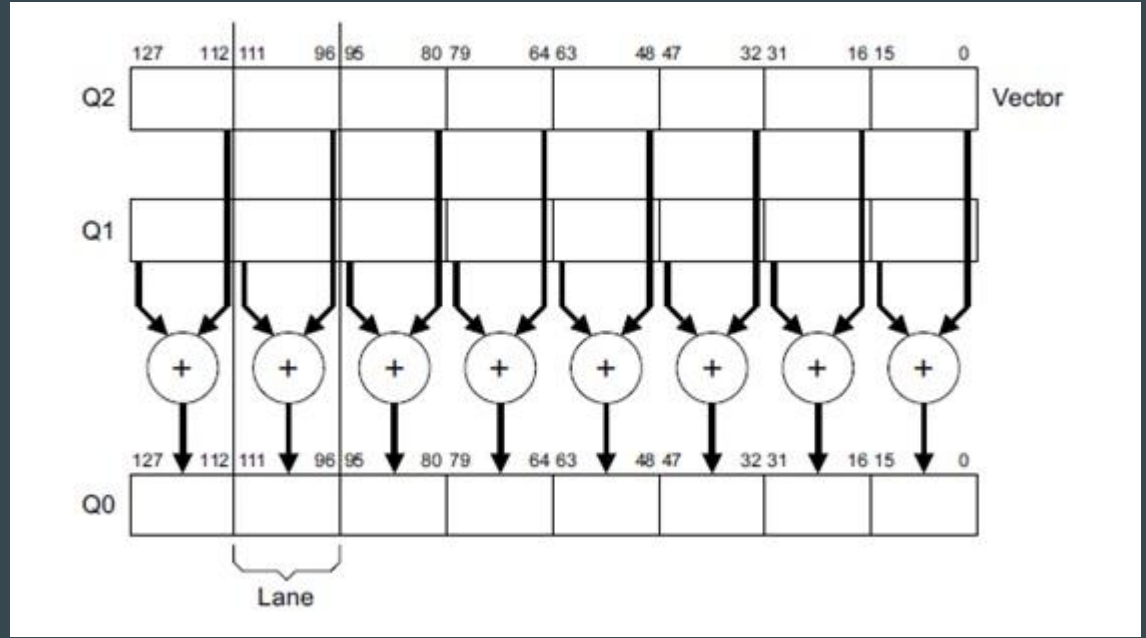
ISAE

- *Instruction Set Architecture Extension*
- Interpretação do registrador é dada pela instrução
- *Vector Data Types:*
 - <type><size>x<lanes>
- *int 32x2*
 - vetor de 64 bits
 - lanes
 - 32 bits por lane

ISAE

- **Biblioteca C:**
(intrinsic)

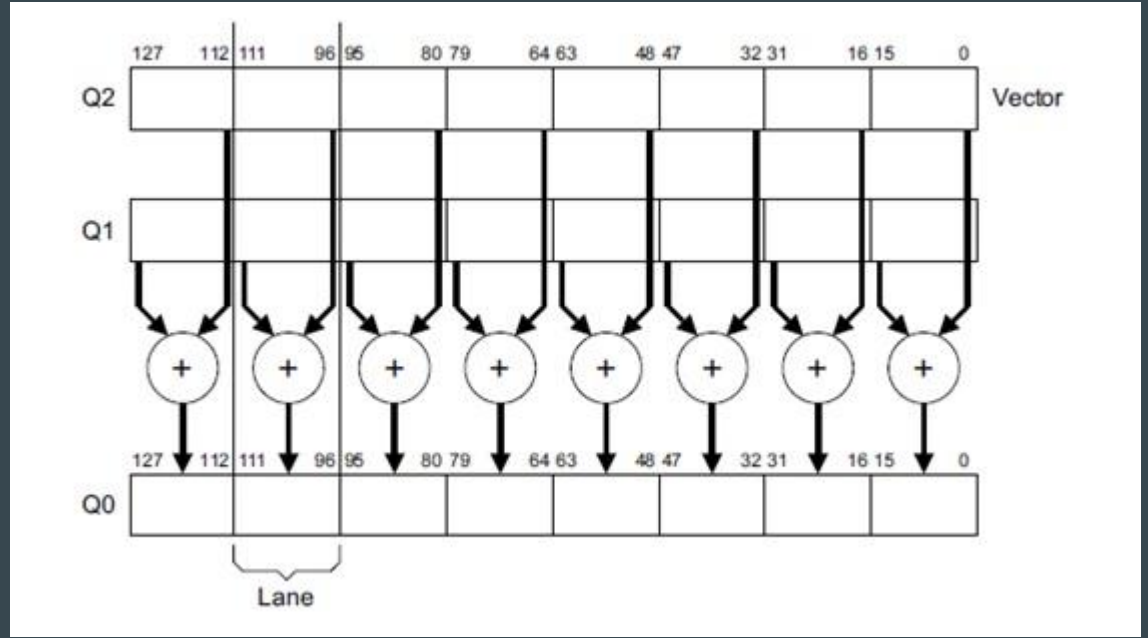
`uint8x16_t vaddq_u8`
(`uint8x16_t a`,
`uint8x16_t b`)



ISAE

- Em linguagem Assembly:

ADD Q0.16B, Q1.16B, Q2.16B



ISAE

- Conjunto de instruções bastante estendido
- Tipos:
 - int
 - float
 - poly(polinomial)
- Tamanhos:
 - 2x64b
 - 2x32b
 - 4x32b
 - 4x16b
 - 8x16b
 - 8x8b
 - 16x8b

Software

- Apesar do tamanho, é regular e bem aceito no mercado
- Compilador C gera código vetorizado
 - Menos eficiente que algoritmos vetorizados
- Bibliotecas *open source*
 - Renderização, *transcode*, processamento de imagens, IA, etc.

Vantagens

- Instruções focadas na área de multimídia
- Bibliotecas *open source*
- Suporta uma grande variedade de *codecs* usados para aplicações na internet

Vantagens

- Aplicações com uso intenso de matrizes e vetores obtém speed up.
 - Jogos
 - Áudio/Vídeo ^[10]
 - Processamento de sinais
 - Simulações

Exemplo

- Player de vídeo embarcado para automóveis
 - Utilização de CPU sem otimização NEON:
 - CPU1: 66%
 - CPU2: 81%
 - FPS @ 720p: 28

Exemplo

- Player de vídeo embarcado para automóveis
 - Utilização de CPU COM otimização NEON:
 - CPU1: 99%
 - CPU2: 99%
 - FPS @ 720p: 42

Exemplo

- Player de vídeo embarcado para automóveis

Speed Up: x1.5

Baixíssimo custo energético! Viabiliza projetos com recursos restritos

Exemplo

- Display dos Smartphones
 - 2011: ARM Cortex A8 (NEON e VFP)

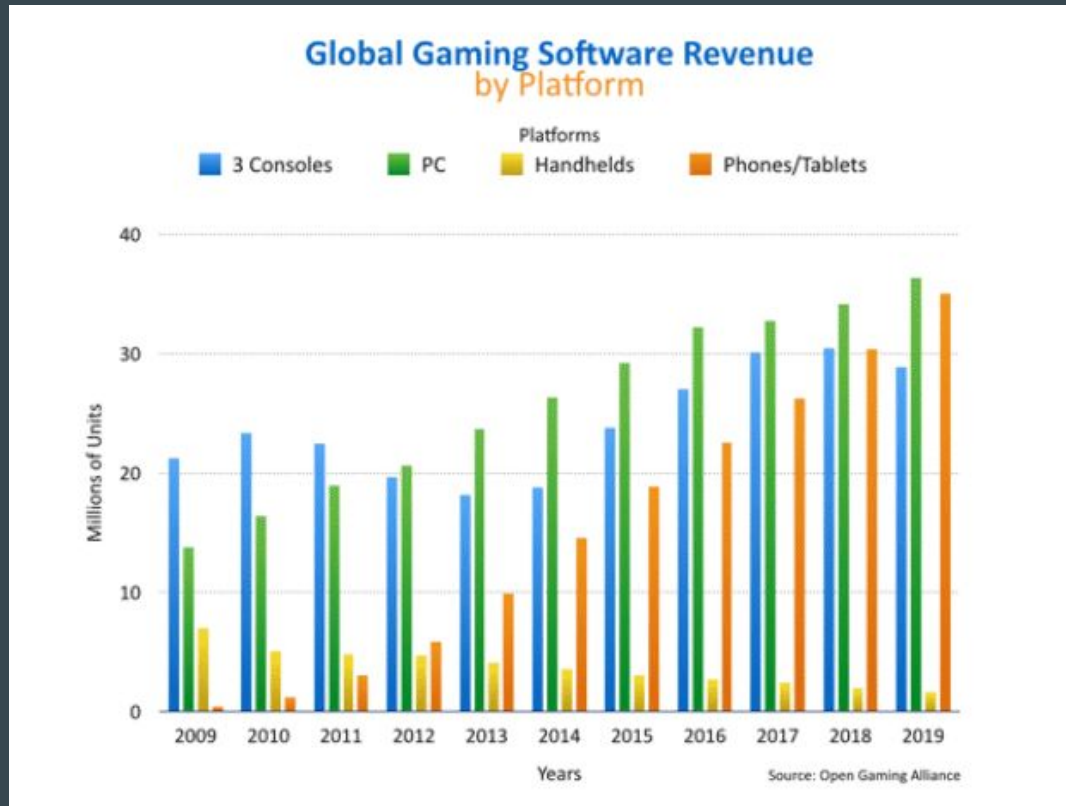
Exemplo: Evolução da Resolução

<https://www.gocompare.com/home-insurance/evolution-of-screen-resolution/>

Exemplo

- Mercado de jogos mobile
 - 2011: ARM Cortex A8 (NEON e VFP)
 - 2014 - Atualmente: ARM Mali GPU

Exemplo: Mercado de jogos online



Desvantagens

- Programas vetorizados são complexos de desenvolver
- Interpretação flexível dos registradores implica em alinhar dados
- Transferência NEON -> ARM lenta
 - Perda de ciclos

Conclusão

- **Tecnologia certa**
 - SIMD, anos 60
- **Momento certo**
 - trend multimídia de 2010
- **Mercado certo**
 - mobile, recursos limitados

Referências

- [1] https://www.arm.com/company/-/media/arm-com/company/Investors/Quarterly%20Results%20-%20PDFs/Arm_SB_Q1_2017_Roadshow_Slides_Final.pdf
- [2] https://www.arm.com/files/pdf/ARM_Arch_A8.pdf
- [3] <http://infocenter.arm.com/help/topic/com.arm.doc.dht0002a/ch01s03s02.html>
- [4] <https://www.embedded.com/design/mcus-processors-and-socs/4457386/2/Designing-a-low-cost--low-power-multicore-ARM-based-AV-player>
- [5] <https://dazeinfo.com/2015/03/12/pc-gaming-market-estimated-grow-35-billion-2018-report/>
- [6] NEON intrinsics: <https://developer.arm.com/technologies/neon/intrinsics>
- [7] ARM Compiler Toolchain Reference:
https://static.docs.arm.com/dui0491/f/DUI0491F_arm_compiler_reference.pdf?_ga=2.148623671.1635372059.1542664566-650097429.1540348225#G15.1144247
- [8] NEON Tutorial: https://people.xiph.org/~tterribe/daala/neon_tutorial.pdf

Referências

[9] The Performance Analysis of ARM NEON Technology for Mobile Platforms:

<http://delivery.acm.org/10.1145/2110000/2103401/p104-jang.pdf>

[10] Packet Ray Tracing with the ARM NEON Architecture - Gustaf Waldemarson:

<http://sam.cs.lth.se/ExjobGetFile?id=683>

[11] github ARM-software compute library: <https://github.com/ARM-software/ComputeLibrary>

[12] Optimizing Mobile Deep Learning on ARM GPU with TVM:

<https://tvm.ai/2018/01/16/opt-mali-gpu.html>

[13] (Comparação ARM x Intel) Use of SIMD Vector Operations to Accelerate Application Code

Performance on Low-Powered ARM and Intel Platforms: <https://core.ac.uk/download/pdf/143891591.pdf>