

UNIVERSIDADE DO VALE DO RIO DOS SINOS
CURSO CIÊNCIAS DA COMPUTAÇÃO

PEDRO HENRIQUE ACCORSI

TRABALHO SOBRE ARQUITETURA ARM

São Leopoldo
2019

Sumário

1. INTRODUÇÃO.....	3
2. HISTÓRIA.....	3
3. CARACTERÍSTICAS PRINCIPAIS.....	3
4. ORGANIZAÇÃO DA ARQUITETURA ARM, FAMÍLIAS E VERSÕES.....	4
5. FAMÍLIA CORTEX.....	5
6. ARMv8, for ARMv8-A ARCHITECTURE PROFILE.....	7
6.1 CORTEX-A53.....	7
6.2 HIERARQUIA DE MEMÓRIA NO CORTEX-A53.....	7
6.2.1 CACHE L1.....	8
6.2.2 CACHE L2.....	8
6.3 PIPELINE NO CORTEX-A53.....	8
6.4 CONSUMO E ÁREA NO CORTEX-A53.....	9
7. CONCLUSÃO.....	10
8. REFERÊNCIAS.....	10

1. Introdução

O presente trabalho consiste em uma breve overview na arquitetura de processadores ARM, abordando temas como sua história, aplicações, detalhes técnicos e comparações de mercado.

2. História

Em 1980, a BBC iniciou o projeto *Computer Literacy Project*, que consistia em trazer computadores à todas as salas de aula das universidades inglesas. Nisso, a BBC precisava de empresas que ajudassem no desenvolvimento de computadores relativamente baratos e de performance razoável. Uma das empresas envolvidas foi a Acorn Computers, que desenvolveu um protótipo inicial para demonstrar à BBC, denominado Proton. O protótipo cumpriu a especificação da BBC, e já em 1981 a Acorn Computers apresentou o BBC Micro, seu produto, tornando-se um grande sucesso.

Em 1983, reconhecendo a necessidade de computadores mais poderosos, a Acorn formou a divisão Advanced Research and Development. O grupo foi estabelecido na tentativa de desenvolver um processador RISC próprio. O resultado inicial do projeto foi o primeiro processador ARM. Inicialmente a sigla se regeria a Acorn RISC Machine, mas mais tarde foi “alterada” para Advanced RISC Machine.

Nisso, no início dos anos 90, a empresa ARM LTDA foi aberta, dando início formal ao que conhecemos hoje. Importante ressaltar, entretanto, que a empresa não produz processadores, apenas os projeta. Suas vendas consistem em certificações/permissões de uso de outras empresas, para que as mesmas implementem suas próprias aplicações a partir dos processadores da ARM.

3. Características principais

Os processadores ARM, como o próprio nome diz, utilizam como base a arquitetura RISC, acrônimo de *Reduced Instruction Set Computer*, em

português, "Computador com um conjunto reduzido de instruções". Isso significa que a filosofia por trás do desenvolvimento da arquitetura consiste em organizar o hardware de maneira que o set de instruções seja simples e preciso, a ponto de ser executado diretamente pelo mesmo, sem haver necessidade de "microcódigo".

Em outras palavras, pode-se entender que tal arquitetura possui um conjunto simples de instruções, porém as executa de maneira extremamente consistente e rápida. Essas instruções, não por coincidência, são as mais comuns e amplamente utilizadas na maioria das aplicações, deixando de lado instruções mais complexas que são raramente usadas.

Além da arquitetura RISC, os processadores ARM também utilizam a filosofia de arquitetura LOAD-STORE, isso é, dados só são manipulados quando estão armazenados nos registradores internos do processador, não havendo acesso direto entre processador e memória principal.

Outro detalhe importante é que a arquitetura ARM aceita, dependendo da família, até três tipos diferentes de Instruções: ARM32, THUMB16 e THUMB32, e, também dependendo da família, possui pipelines que vão de 3 até 15 estágios, tanto in-order quanto out-of-order.

Por último, uma das maiores razões da hegemonia ARM: baixo consumo de energia e alto poder de processamento; atualmente, as arquiteturas ARM se mostram quase imbatíveis nesse aspecto, um dos mais importantes, explicando porque a grande maioria das aplicações reais hoje em dia possui um (ou mais) processadores ARM em seu core.

4. Organização da arquitetura ARM, famílias e versões

Os processadores e microcontroladores construídos com a arquitetura ARM são identificados conforme a versão da arquitetura adotada, o perfil e suas variantes.

Até o momento já foram definidas 8 versões de arquitetura ARM, sendo atualmente em uso apenas 5, identificadas pelo Prefixo ARMv, sendo elas ARMv4, ARMv5, ARMv6, ARMv7 e ARMv8.

Cada uma das versões possui algumas famílias de processadores que as "implementam", possuindo cada uma suas respectivas otimizações em

detrimento uma ou outra aplicação em específico. Abaixo pode-se visualizar uma tabela exemplificando algumas versões, com suas famílias e respectivas aplicações.

Example ARM component	Architecture Generation	Example Application	Approximate date of introduction
ARM1	ARMv1	Acorn Computer in internal testing	1985
ARM2	ARMv2	Acorn Archimedes (Macintosh-era PC)	1987
ARM6	ARMv3	Apple Newton MessagePad 100 series	1994
ARM7TDMI	ARMv4	Game Boy Advance, Nintendo DS*, iPod	2001
ARM9E	ARMv5	Nintendo DS*, Nokia N-Gage, Airport Extreme N basestation	2004
ARM11	ARMv6	iPhone, iPhone 3G, iPod touch	2007
Cortex-A8	ARMv7	Palm Pre, iPhone 3GS	2009

* Nintendo DS incorporates both processors for different uses.

Fonte: anandtech, 2015.

5. Família Cortex

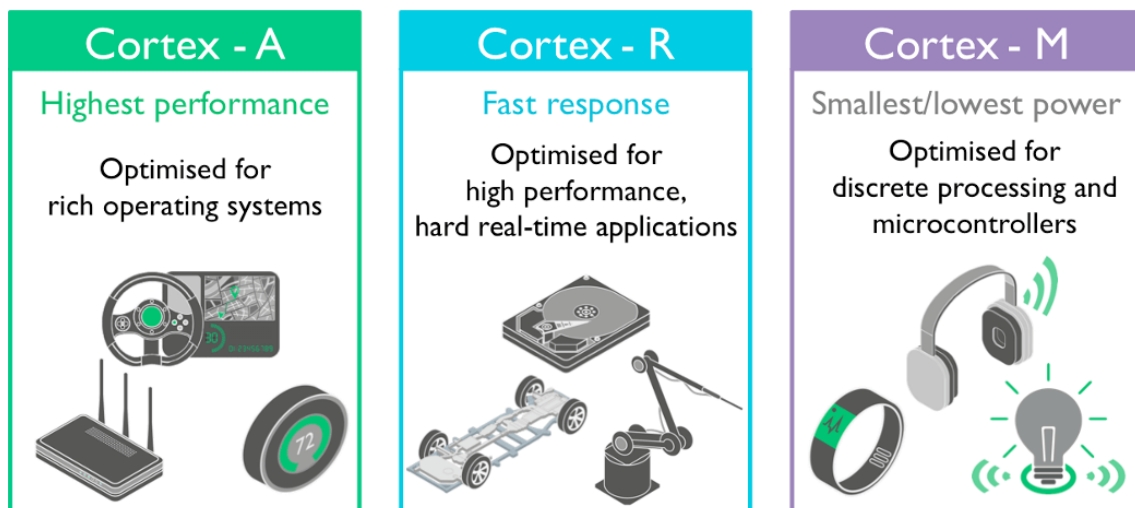
Considerando as versões mais atuais, ARMv7 e ARMv8, temos a família córtex como carro chefe, com 3 perfis de uso definidos, ARMv7/8-A, ARMv7/8-R e ARMv7/8-M.

Hierarquicamente temos, então, o seguinte fluxo: versão de arquitetura ARMv7/8, família Cortex, perfil A/R/M.

Cada um dos três perfis tem um foco de otimização diferente, de maneira que exista um processador mais recomendado para cada diferente aplicação que se possa imaginar. O perfil A, por exemplo, possui foco em alta performance, tendo uso em videogames, celulares top de linha, notebooks e desktop gamers, por exemplo; no perfil M, tem-se o foco de resposta rápida e precisa, possuindo aplicações no ramo da medicina e automobilismo, por exemplo, onde precisões de altíssimo grau são obrigatórias. Por fim, existe o perfil R, que tem foco em

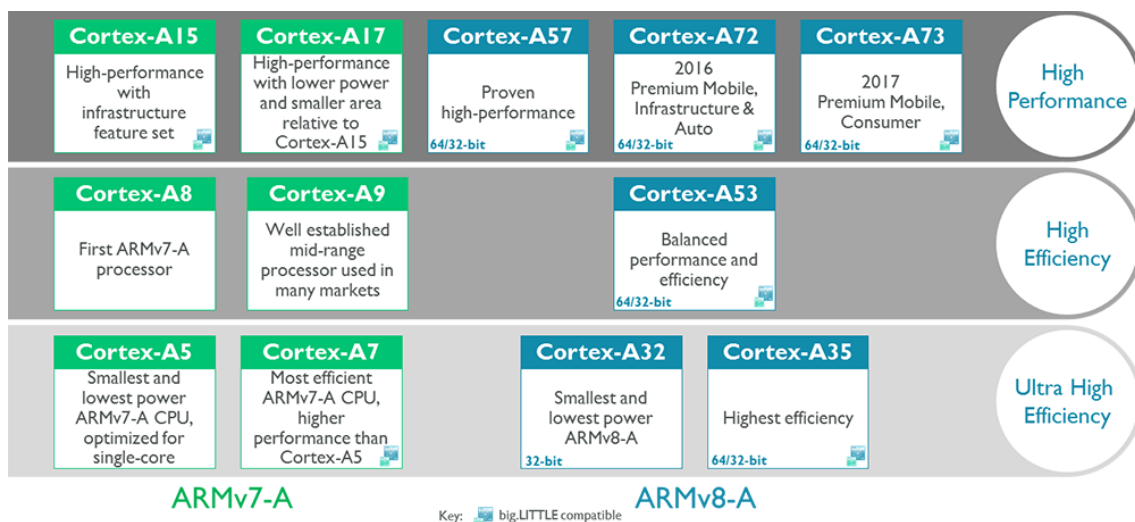
dispositivos médios que precisam de um micro controlador sem exigências enormes de performance ou de precisão, sendo encontrado em headphones e lâmpadas inteligentes, por exemplo.

Abaixo, temos uma imagem que exemplifica os perfis descritos anteriormente.



Fonte: arm community, 2016.

Dadas as três famílias, o presente trabalho tomará como foto o perfil A, com foco em performance. Esse perfil é implementado por diversos processadores da linha, como os listados abaixo.



Fonte: hexus, 2016

6. ARMv8, for ARMv8-A architecture profile

Como descrito, a versão ARMv8, da família Cortex, perfil A, é implementada por diversos processadores diferentes, cada qual tendo sua própria arquitetura, porém mantendo o essencial da família.

Nisso, os dados apresentados abaixo serão focados na linha Cortex® -A53.

6.1 Cortex® -A53

O ARM Cortex-A53 é uma das duas primeiras microarquiteturas que implementam o conjunto de instruções ARMv8-A de 64 bits. Algumas de suas características seriam um pipeline in order superescalar, com dual issue simétrico, capaz de emitir duas instruções. Mais eficiente em termos de energia, que a micro-arquitetura Cortex-A57, por exemplo, porém menos performático.

6.2 Hierarquia de memória no Cortex® -A53

O modelo A53 conta com três níveis de cache, sendo o terceiro deles opcional, possuindo a seguinte organização:

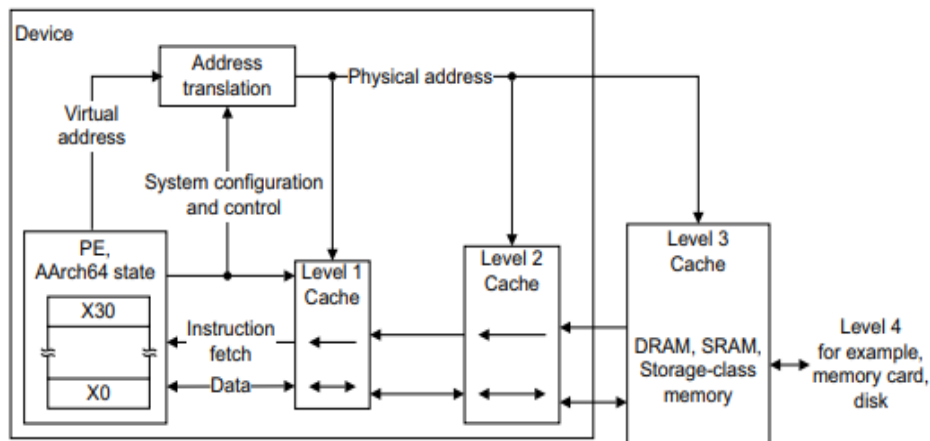


Figure B2-1 Multiple levels of cache in a memory hierarchy

Fonte: Arm® Cortex® -A53 MPCore Processor Revision: r0p4 Technical Reference Manual

6.2.1 Cache L1

O nível L1 é dividido em duas memórias, uma para Instruções e a segunda para Dados, possuindo tamanhos configuráveis de 8KB, 16KB, 32KB ou 64KB.

O sistema de Instruções possui as seguintes características, entre muitas outras:

- Linha de cache de 64 bytes.
- 2-way set associative.

Já o sistema de dados possui as seguintes características, entre muitas outras:

- Linha de cache de 64 bytes.
- 4-way set associati

6.2.2 Cache L2

O nível L2, diferente do L1, não possui dois sistemas separados para Dados e Instruções; entretanto, também possui tamanho configurável de 128KB, 256KB, 512KB, 1MB e 2MB.

Algumas características importantes são as seguintes:

- Comprimento de linha fixa de 64 bytes.
- Cache fisicamente indexado e “tagged”
- 16-way set associative

6.3 Pipeline no Cortex® -A53

Esse perfil possui um pipeline de 8 estágios superescalar com a característica “symetric dual issue”, isso é, possui “dois datapaths” que podem executar instruções por separado, com mesma taxa de latência e throughput; além disso, seu pipeline é in order, as instruções são processadas “por ordem de chegada”, não havendo otimizações em casos de não dependência de instruções subsequentes, por exemplo.

Abaixo segue uma representação do pipeline de 8 estágios utilizado no córtex-A53.

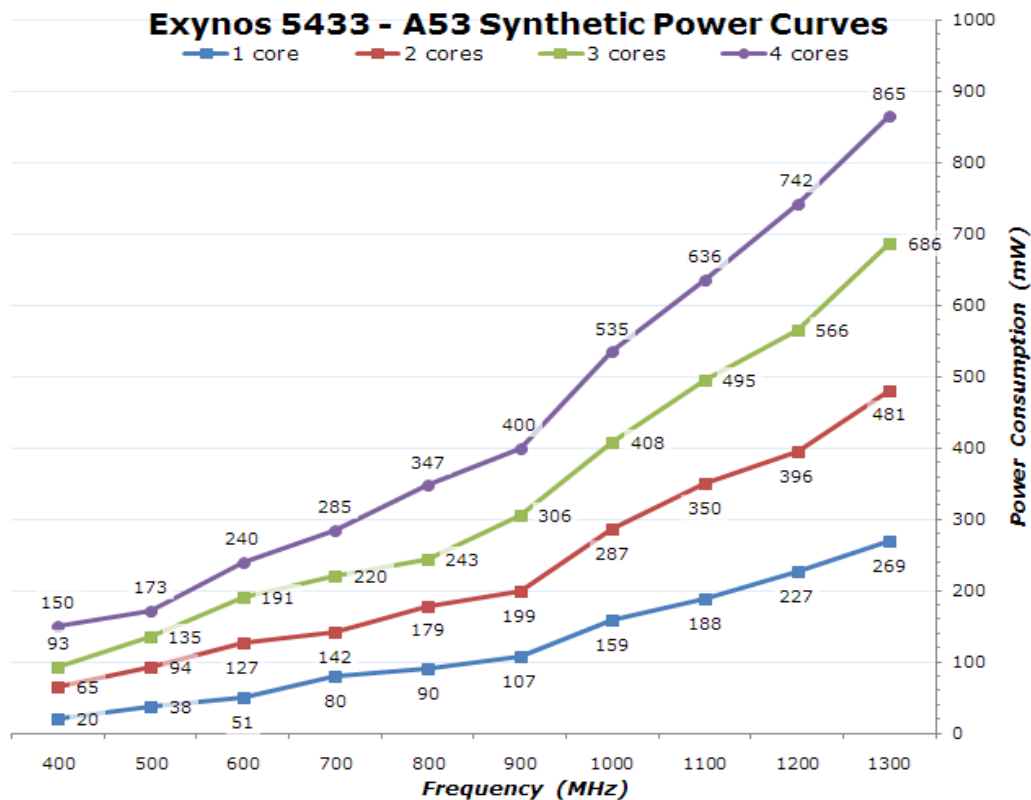
Fetch	Decode 1	Decode 2	Decode 3	Operand	Execute 1	Execute 2	Retire
Fetch instruction	Decode instruction			Operand	Execute instruction		Retire

Fonte: Comparison between Intel Atom and ARM Cortex-A54

6.4 Consumo e área

O consumo de um A53 é bastante interessante, chegando a 865mW executando tarefas com seus quatro núcleos a 1300MHz, passando 100MHz da frequência recomendada pela ARM. Não podemos chamar de overclock, mas claramente é uma maneira de “forçar” o chip ao seu limite, no qual apresentou números razoáveis, indo de encontro com a proposta para a arquitetura.

É interessante analisar, de acordo com o gráfico, o “pico” de consumo a partir da faixa de 1000MHz, talvez contradizendo a recomendação da desenvolvedora de manter os núcleos a 1200MHz.



Fonte: anandtech, 2015.

Já em relação a área, cada núcleo tem em torno de 0.7mm² de área, e o cluster final tem cerca de 4.58mm².

7. Conclusão

Ao final da elaboração do trabalho, pode-se quase afirmar que a ARM LTDA é a dona do mercado de arquiteturas de processadores. Seu portfólio enorme de famílias e microarquiteturas diferentes, bem como sua “mágica” de baixo consumo e alta performance, ainda estão por ser batidas por alguma outra desenvolvedora.

Seus processadores possuem uma engenharia minuciosa por trás, levando em consideração os mais pequenos detalhes na hora de implementar algo novo.

Além disso, é incrível a liberdade que a ARM dá à seus usuários de configurarem seu chip quase que “a gosto”, ativando ou desativando (quase) todas as funcionalidades, o que resulta em mais liberdade ao usuário (usuário, no caso, fabricante que desenvolve aplicações com ARM) e autonomia ao chip, que não precisa alimentar blocos específicos que se encontram desativados.

8. Referências

History – ARM. Disponível em: <<https://bit.ly/2R5qd0l>> Acesso em: 11 de junho de 2019.

SMITH, Ryan. **ARM Announces ARMv8-M Instruction Set For Microcontrollers – TrustZone**

Comes to Cortex-M. Disponível em: <<https://bit.ly/2XCbdtN>> Acesso em: 11 de junho de 2019.

DAVE, Kinjal. **Introducing Cortex-A32: ARM’s smallest, lowest power ARMv8-A processor.** Disponível em: <<https://bit.ly/2WzOWzW>> Acesso em: 11 de junho de 2019.

Lionel Belnet Blog: Introducing Cortex-A73. Disponível em <<https://bit.ly/2X6LTOQ>> acesso em: 11 de junho de 2019.

HUMRICK, Matt. **Exploring DynamIQ and ARM’s New CPUs? Cortex-A75, Cortex-A55.** Disponível em: <<https://bit.ly/2KH0jik>> Acesso em: 11 de junho de 2019.

ARM® Architecture Reference Manual ARMv8, for ARMv8-A architecture profile. Disponível em <<https://bit.ly/2KEBcg5>> Acesso em: 11 de junho de 2019.