

ARQUITETURA ARM11 – (ADVANCED RISC MACHINES)

Rodolfo William de Oliveira, rodolfowoliveira@hotmail.com

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Bragança Paulista.

RESUMO. *A intenção deste trabalho é esclarecer e abordar a história da arquitetura, princípio de funcionamento, as principais características e suas aplicações. Visando uma abordagem clara e simples, o trabalho tem como objetivo simplificar a arquitetura e ajudar a compreender o poder desta ferramenta que ainda é pouco usada em meios acadêmicos, por ser de difícil acesso e ter um acervo restrito.*

Palavras chaves: Processadores. Arquitetura ARM. ARM11. Embarcados. RISC.

1. INTRODUÇÃO

O profissional necessita de reciclagem de conhecimento e a atualização nas diversas ramificações do desenvolvimento tecnológico, seja em aparelhos residenciais, industriais, médicos entre muitos outros, sempre procurando melhorar todos os aspectos. Uma dessas áreas de desenvolvimento é o processamento de dados, o mundo presenciou muita evolução tecnológica, o processador é uma delas, sempre tentando melhorar desempenho, reduzir tamanho, diminuir custo, garantir aplicabilidade, bem como baixo consumo de energia, esses objetivos foram sendo melhorados com o decorrer dos tempos. Hoje o processador com arquitetura ARM ganha mercado, possuindo um alto desempenho que tende a sempre aumentar junto com a redução do tamanho e custo. A aplicabilidade se encontra e tende a somente aumentar e seu baixo consumo de energia permite com que o processador com arquitetura ARM se sobressaia em relação a muitos outros no mercado, fazendo deste modelo de processador um dos mais procurados quando falamos de produtos que utilizam de bateria interna, por exemplo, os Tablets, Smartphones entre outros. Porém a divulgação desta arquitetura ainda é pequena, este trabalho é o primeiro a ser escrito no Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Bragança Paulista, esperamos que com a divulgação trabalhos futuros sejam elaborados.

2. REVISÃO BIBLIOGRÁFICA

2.1. História

De acordo com (GOMES, 2005) a arquitetura ARM (*Advanced RISC Machines*) com uma tradução direta significa “Maquina RISC Avançada”, começou com um projeto em 1983 na Arcon Computers de Cambridge, Inglaterra, para desenvolver um processador que fosse similar ao já usado MOS Technology 6502. O desenvolvimento da primeira versão foi terminado em 1985, chamado ARM1. Esta arquitetura não chegou ao mercado e a Arcon continuou no desenvolvimento de novos projetos, terminando no ano seguinte a segunda versão, chamada de ARM2.

Além do 6502, a arquitetura ARM tem uma grande herança do processador Berkeley RISC1 afirma (GOMES, 2005), como instruções de tamanho fixo (32 *bits*), formato das instruções e a arquitetura “*load-store*”. Apesar disso, a Arcon optou por ter instruções multiciclo, ao contrário do RISC1. Em sua época, o ARM2 era o processador de 32 “*bits*” mais simples no mercado, com apenas 30.000 transistores, mas ainda assim superava o desempenho de muitos processadores (RISC ou CISC) mais complexos, como o 286 da Intel.

O desenvolvimento continuou, e o nome original (*Arcon RISC Machine*) deixado de lado quando a ARM Ltda. foi criada e assumiu as patentes e o desenvolvimento da arquitetura ARM. Atualmente a arquitetura já conta com sua 11ª versão, mas as versões antigas ainda são usadas e desenvolvidas, já que o uso de cada família é voltado para um nicho de mercado.

Evolução dos processadores ARM:

1. **ARM1 – 1985** – Protótipo que não chegou ao mercado;
2. **ARM2 – 1986** – Possuía um barramento de dados de 32 *bits*, porém só usava 26 *bits*, deixando os seis *bits* restantes para as *flags* de estado. Foi considerado o microprocessador de 32 *bits* mais simples, com 30.000 transistores.
3. **ARM3 - 1989** - 4kb de cache, melhorou a performance. Ao longo dos anos 90 a ARM torna-se líder no mercado de processadores embarcados, pois oferece alto desempenho com baixo consumo de energia.
4. **ARM – 1990** – “*Advanced RISC Machines Ltda.*” – Fundada pelo grupo “*Apple Computer + Acorn Computer + VLSI Technology*”.
5. **ARM6 – 1991** - Processador 32 *bits* tinha somente 35 mil transistores. A Apple usou em seus primeiros PDAs processadores ARM610.
6. **ARM7 – 1994** - Usa a metade da energia usada no ARM6, ele tem 50% a 100% mais performance.
7. **ARM7 - 1994** - *Pipeline* de três estágios. Processador de 32 *bits* que combina tamanho reduzido, pouca

energia e alta performance, (usado em *Game Boy Advance*, *Ipod*).

8. **StrongARM** – 1995 - Série mais rápida que os ARM anteriores, têm limitações de softwares com suas versões anteriores, possui diferentes tamanhos para memória e instruções.
9. **ARM8** – 1996 - *Pipeline* de cinco estágios, banda de memória duplicada. Aumento na taxa de clock, redução na CPU.
10. **ARM9TDMI** - 2000 - *Pipeline* de cinco estágios também, usado em calculadoras HP. Reduziu o espaço do programa executável em 35%.
11. **ARM10E** - 2002 - *Pipeline* de seis estágios cache variando entre 32k e 16k, dependendo do modelo excelente combinação de performance pouco consumo de energia.
12. **Xscale** - 2006 - Linha produzida pela Intel, baseado na arquitetura *StrongARM*, usam menos energia porque trabalham em velocidades inferiores do que os CPUs desktop. Trabalham entre 133MHz e 400MHz, comparado a 2GHz aos CPUs usados em desktop. *Pipeline* de sete estágios.
13. **ARM11** – 2008 - 532-665MHz, usado em comunicadores como *Nokia e90*, *Apple Iphone*.
14. **Cortex** - 2010 em diante - Atualmente o processador mais poderoso da ARM, de 600Mhz até mais de 1GHz, inclui treze estágios de *pipeline*, como estado de espera programáveis.

Abaixo os principais fabricantes que atuam no mercado:

- *Apple Inc.*
- *Intel*
- *LG*
- *Microsoft*
- *Nintendo*
- *Nvidia*
- *Sony*
- *Qualcomm*
- *Samsung*
- *Yamaha* entre outros.

Atualmente mais de dois bilhões de processadores ARMs são fabricados por ano, a ARM Ltda. desenvolve os “cores” (núcleos) dos processadores ARM, licencia seus “*IP cores a design houses*”, fábricas de semicondutores, por exemplo, desenvolve também outras tecnologias de apoio aos seus processadores como, por exemplo:

- Ferramentas de software;
- Placas de desenvolvimento;
- Hardwares para debug;
- Softwares aplicativos;
- Barramentos;
- Periféricos;

2.2. Principais Características

O ARM tem uma arquitetura que em geral é baseado no modelo RISC 32 *bits* que é utilizada em larga escala em varias aplicações. Foi projetada para ser programada utilizando de linguagem Assembly e contem diversas características que fazem com que a arquitetura seja eficiente.

Estas características são, por exemplo, a combinação de um hardware simples com um conjunto de instruções reduzidas, permitindo que alcance uma alta eficiência no quesito consumo de energia e tamanho reduzido.



Figura 1. – Processador ARM.
Fonte: Adaptado de ARM, 2014.

Características diferenciais que o processador ARM e o Processador x86 possuem:

Item	Processador ARM11	Processador x86
1	Processador de 32 Bits;	Processador de 64 Bits;
2	16 Registradores de uso geral;	8 Registradores de uso geral;
3	Possibilidade de utilização de conjunto de instruções extensíveis com co-processadores;	Possibilidade de utilização de conjunto de instruções extensíveis com co-processadores;
4	Instruções básicas de três endereços;	Instruções básicas de três endereços;
5	Possibilidade de executar instruções de 16 bits usando arquitetura de programação Thumb;	Maiores possibilidades de instruções a serem executadas, usando varias arquiteturas de programação;
6	Baixo consumo de energia;	Alto consumo de energia;
7	Tamanho do núcleo reduzido;	Tamanho do núcleo pouco reduzido;
8	Até 16 co-processadores lógicos;	Até 4 co-processadores;

Tabela 1. – Diferenças entre Processadores ARM e Processadores x86.

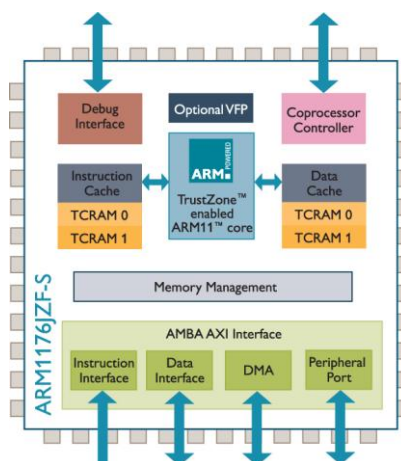


Figura 2. – Arquitetura do processador ARM11 em diagrama de blocos.

Fonte: Adaptado de ARM11, 2014.

2.3. Princípio de Funcionamento

A arquitetura ARM foi desenvolvida para possibilitar implementações muito compactas, sem deixar de lado o alto desempenho. Isso é possível pela simplicidade dos processadores ARM. Importante lembrar que implementações pequenas implicam em baixo consumo de energia, o que torna esses processadores interessantes para aplicações móveis.

O ARM é tipicamente um RISC (*Reduced Instruction Set Computer*). Algumas características interessantes das instruções ARM:

- Conjunto grande e uniforme de registradores;
- Arquitetura de *LOAD/STORE*. Operações de processamento de dados não operam diretamente com o conteúdo da memória, somente com o conteúdo de registradores;
- Modos de endereçamento simples, com todos os endereços de *load/store* sendo determinados a partir dos registradores ou pelos campos da instrução;
- Uniformidade e tamanho fixo dos campos das instruções para simplificar a decodificação de instruções;
- Controle sobre a ULA (Unidade Lógica Aritmética) e sobre o *shifter* (deslocador) em todas as instruções de processamento de dados;
- Auto incremento e decremento dos endereços das instruções;
- Instruções de múltiplos *loads/stores* para maximizar a performance;
- Execução condicional da maioria das instruções.

A arquitetura possui 31 registradores de propósito gerais todos de 32 *bits*. Em qualquer momento apenas 16, dos 31, registradores são visíveis. Os registradores restantes são usados em operações de exceção, quando o processador entra em um de seus modos especiais de operação e substitui alguns dos 16 registradores comuns por registradores específicos do modo. Dos 16 registradores, dois tem papel especial:

1. *Link Register* (R14) - possui o endereço da próxima instrução a ser executada após um *Branch and Link* (BL), que é uma instrução usada na chamada de sub-rotinas. Excluindo essa situação, o R14 pode ser utilizado normalmente pelo programador.
2. *Program Counter* (R15) - possui o endereço da próxima instrução a ser executada pelo processador. Sempre possui o valor do endereço da instrução atual mais oito *bytes*.

É comum o uso do registrador R13 para fins de *Stack Pointer*, todos os outros 13 podem ser usados para propósito geral. A partir da versão três o ARM passou a separar em registradores independentes o seu status.

Existem dois registradores responsáveis por essa função, o *Current Program Status Register* (CPSR) e o *Saved Program Status Register* (SPSR), ambos de 32 *bits*. O primeiro possui 4 *bits* condicionais de *flag* (*Negative*, *Zero*, *Carry* e *Overflow*), 2 *bits* de controle de interrupção, 5 para controle do modo atual do processador e 1 que indica o tipo de instrução que está sendo executada (ARM ou Thumb). O segundo registrador de status guarda o conteúdo do primeiro durante uma exceção, para possibilitar um retorno seguro do contexto anterior da CPU, existe um registrador SPSR para cada modo de exceção do processador.

A arquitetura ARM suporta até 7 modos de operações, apresentados a seguir:

1. *User*: execução normal de programas. Possui restrições de acesso a registradores;
2. *FIQ* (*Fast Interrupt*): suporta a transferência rápida de dados;
3. *IRQ* (*Interrupt*): usado para manipulação de interrupções de propósito geral;
4. *Supervisor*: é um modo protegido para o sistema operacional;
5. *Abort*: implementa memória virtual e/ou proteção de memória;
6. *Undefined*: suporta emulação em software de co-processadores;
7. *System*: executa tarefas privilegiadas do sistema operacional, existente a partir da versão quatro do ARM.

2.4. Ferramentas de Desenvolvimento

Para que possamos entender todos os processos de funcionamento de um processador ARM, temos vários tipos de ferramentas de desenvolvimento, por exemplo:

1. Placa de desenvolvimento;
2. Módulo de desenvolvimento;
3. IDE/Compilador;
4. Cabo USB/Serial;
5. Depurador JTAG;
6. Livros e Manuais;



Figura 3. – Placa de Desenvolvimento *FriendlyARM Tiny6410*.
Fonte: Adaptado de TINY6410, 2014.

O kit de desenvolvimento oferece muitas possibilidades de experimentos, contendo vários periféricos. Sendo possível a utilização de:

1. Interface *TouchScreen*;
2. Interface Ethernet 100M;
3. Interface Serial RS232;
4. Interface Mini USB 2.0;
5. Interface USB;
6. Interface de Áudio IN/OUT;
7. Interface de Microfone;
8. Interface TV-OUT;
9. Interface *SDcard*;
10. Receptor Infravermelho;
11. Porta Serial TTL;
12. Interface para Câmera CMOS;
13. Conjunto de I/Os GPIO;
14. Interface ADC/DAC;
15. Interface *SDWiFi*;
16. Interface JTAG;
17. *Buzzer* controlado por PWM;
18. Memória I2C-EEPROM;
19. RTC – *Real Time Clock* com Bateria.
20. LEDs;
21. Botões;

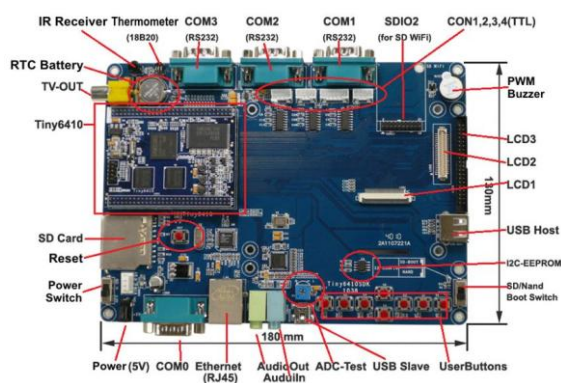


Figura 4. – Periféricos do kit *FriendlyARM Tiny6410*.
Fonte: Adaptado de TINY6410, 2014.

Com as ferramentas necessárias em mãos podemos dar início ao processo de aprendizado, o kit *FriendlyARM Tiny6410* tem muitos recursos e possibilidades de utilização, a cada parte de testes e utilização podemos analisar o quanto a ferramenta é poderosa, ao compararmos com outros segmentos no mercado. Muitas empresas tentam acompanhar a evolução destes processadores, consequentemente o progresso tecnológico acelerado resulta em muitas versões pouco usadas e diferenças gritantes entre as famílias que chegaram ao mercado.

2.5. Comparativo entre arquitetura x86 e ARM

Com a demanda de utilização de processadores com alta eficiência é natural o desejo de comparação, principalmente com dispositivos que a propaganda e o marketing consagram. Diante deste questionamento entre qual processador apresenta melhor desempenho: um processador Intel I7 ou um ARM11. Para analisar foi especificado um teste com operações básicas utilizando banco de dados. Abaixo podemos analisar as diferenças de tempo de resposta entre o processador ARM1176JZF-S e o processador Intel I7 terceira geração, os teste foram feitos por (MOTYCZKA, 2014) utilizando banco de dados *PostgreSQL* e *CouchBD*.

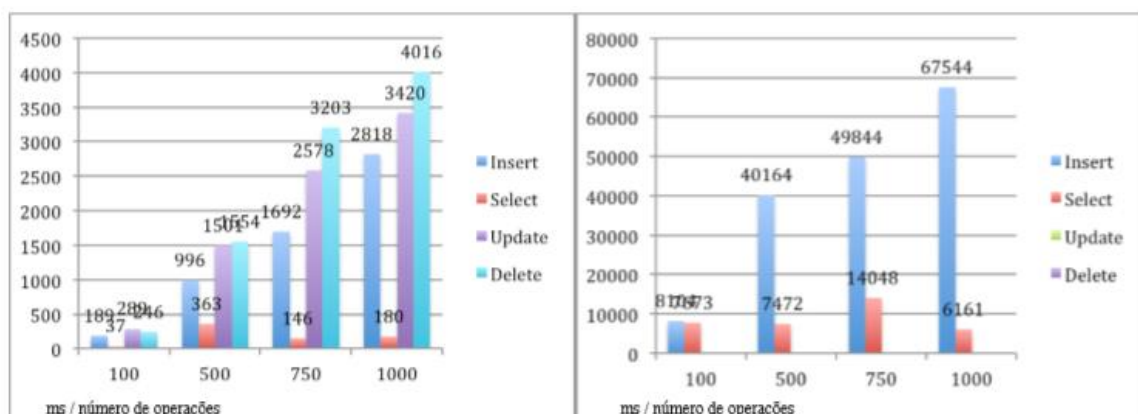


Figura 5. – Tempo de resposta das operações realizadas no banco de dados *CouchBD* na arquitetura x86 “Intel I7” (à esquerda), e ARM1176JZF-S (à direita).
Fonte: Adaptado de MOTYCZKA, 2014.

A figura 5 mostra o tempo medio (ms) de cada operação realizada, a tendência é que quanto maior a quantidade de iterações, maior é o tempo de execução. Abaixo a figura 6 demonstra a utilização do banco de dados *PostgreSQL*.

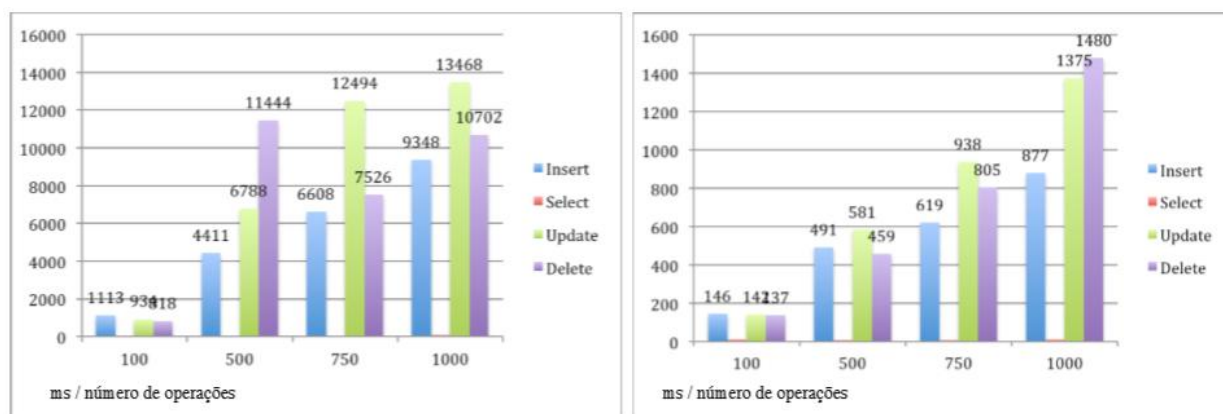


Figura 6 – Tempo de resposta das operações realizadas no banco de dados *PostgreSQL* na arquitetura x86 “Intel I7” (à esquerda), e ARM1176JZF-S (à direita).
Fonte: Adaptado de MOTYCZKA, 2014.

Analisando os resultados obtidos pelos gráficos do (MOTYCZKA, 2014), é possível afirmar que o banco de dados *PostgreSQL* possui melhores respostas nas duas arquiteturas. Porem analisando o consumo de energia o ARM ganha desparadamente utilizando *PostgreSQL* como atesta a figura 7 abaixo.

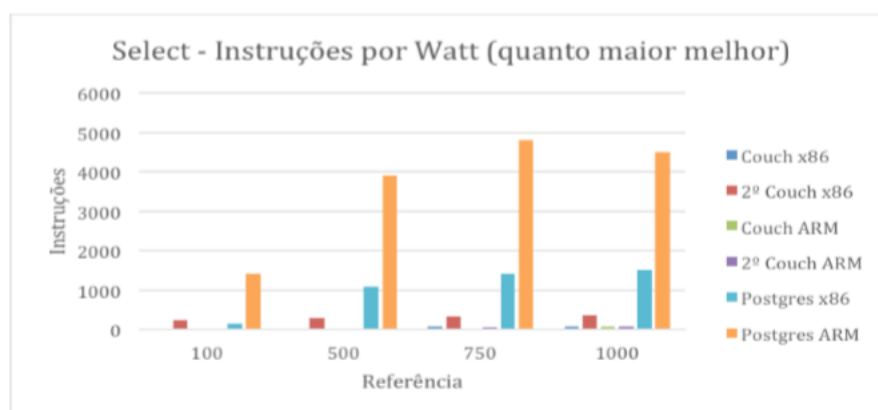


Figura 7. – Análise de consultas por Watt.
Fonte: Adaptado de MOTYCZKA, 2014.

3. ESTUDO DE CASO

Com suas características de alto processamento e baixo consumo de energia a arquitetura ARM demonstra uma possível aplicação para várias empresas que necessitam melhorar seus produtos, a maioria dos produtos que estas empresas desenvolvem utilizam da arquitetura x86, por ser de fácil acesso e ter um baixo custo. A necessidade de procurar um outro processador surgiu da demanda de que os produtos fabricados atualmente necessitam de permanecer em operação por uma maior quantidade de tempo quando estão operando com baterias. O consumo de energia da arquitetura x86 é alto comparado com o consumo de energia da arquitetura ARM, restando somente um quesito para garantir que o ARM11 possa ser uma possibilidade de utilização nos produtos fabricados atualmente. Além de ter um consumo de energia extremamente baixo o processador deve ter uma capacidade de processamento parecida ou superior que a atual solução com a arquitetura x86. Para podermos quantificar e avaliar seu desempenho foi especificado um teste que simula um sistema real em operação. O teste tem como princípio comparar um sistema completo com software e hardware que opera com estabilidade na arquitetura x86.

O hardware e o software foram adaptados para funcionarem na arquitetura ARM, com isso o teste foi colocado em operação. O teste durou 3 horas e demonstrou um desempenho semelhante ao desempenho da solução com a arquitetura x86. Segue abaixo uma tabela com os resultados obtidos no teste e nas análises com o processador ARM11 e os resultados e análises obtidos como de costume da arquitetura x86.

Item	Arquitetura ARM11 TINY6410	Arquitetura x86 Intel Atom - mini ITX
Consumo em Watts	3W	15W
Consumo de Memória RAM	80%	50%
Consumo de CPU	88%	68%
Custo	Alto	Baixo

Tabela 2. – Comparativo de desempenho da Arquitetura ARM11 e Arquitetura x86.
Fonte: Adaptado de ARM, 2014.

4. CONCLUSÃO

Concluimos que a arquitetura ARM é uma ferramenta poderosa, possibilita uma vasta área de utilização e sua utilização no mercado esta crescendo de tempos em tempos. Para que um produto seja desenvolvido é estritamente necessário a análise de o que realmente é necessário utilizar pois, a arquitetura proporciona um alto poder de processamento e baixo consumo de energia, estas duas características são as de maior relevância, com a utilização do kit de desenvolvimento garantimos e afirmamos o que algumas literaturas já indagam. Podemos acrescentar que ferramentas de desenvolvimento com arquitetura ARM não são fáceis de se encontrar e que poucas entidades de ensino a colocam em suas grades curriculares. Este tipo de ferramenta deve ser mais explorada em ambientes educacionais visando disseminar como esta ferramenta pode ser utilizada. Também concluimos no estudo de caso que o processador ARM11 pode ser utilizado, o processador teve consumos altos no teste e mesmo assim atende as necessidades atuais.

Ressaltamos que o ambiente de testes como hardware e software tiveram que sofrer adaptações para que cada arquitetura pudesse operar e com isso constatamos que cada detalhe específico de cada família contribuiu para que seus desempenhos fossem parecidos, com o resultado do teste na tabela 2, podemos afirmar que a arquitetura ARM11 não supera a x86 em alguns quesitos como consumo de CPU, consumo de memória RAM e custo. Porém seu baixo consumo é muito atrativo resultando com que os outros itens tivessem menor peso na análise final, pois o objetivo do estudo de caso efetuado foi de encontrar uma solução que tenha alto desempenho de processamento e um baixo consumo de energia, sendo os dois itens atendidos.

5. AGRADECIMENTOS

Agradeço primeiramente à Deus por ter me guiado até o presente momento, agradeço a minha família que me incentivou a chegar até esta última etapa, agradeço aos amigos e colegas da graduação que acompanharam todas as etapas difíceis e fáceis da minha vida. Agradeço a todos os funcionários desta instituição que fizeram com que fosse possível concluir minha graduação e a produção deste trabalho.

6. REFERÊNCIAS

MOTYCZKAL, Leonardo B., Um Comparativo de Consumo Elétrico entre Arquiteturas x86 e ARM em Servidores de Bancos de Dados, Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUI) – Ijuí – RS – Brasil, 2014.

RODRIGUES, Guilherme, Smartphones e Suas Tecnologias, São Carlos, 2009.

RENAULT, Luís, Grupo de Robótica Experimental do Instituto Militar de Engenharia, <http://ead.ime.eb.br/course/view.php?id=156>, acessado em 04/06/2014.

PEREIRA, F. Tecnologia ARM microcontroladores de 32 bits. 1º ed. Érica: São Paulo 2007.

SOUZA, D. R. ARM7 - o poder dos 32 bits, teoria e prática. 1º ed. Érica: São Paulo 2006. <http://www.olimex.com/dev/index.html> - desenvolvedor de placas e ferramentas para microcontroladores. Acesso em 02/04/2014.

SILVESTRE, Erich, Estudo Sobre Processador ARM7, Universidade Federal de Santa Catarina, Florianópolis, 2007.

GOMES, Pedro H., A Arquitetura ARM, UNICAMP, 2005.

TECMUNDO, Por que os processadores ARM podem mudar o rumo dos dispositivos eletrônicos?, Acesso em 23/04/2014, <<http://www.tecmundo.com.br/qualcomm/7708-por-que-os-processadores-arm-podem-mudar-o-rumo-dos-dispositivos-eletronicos-.htm>>.

SAMSUNG, *User's Manual RISC Microprocessor*, S3C6410X, 2008.

TINY6410, User Manual, 2013.

ROCHA, Daniel, Historia da Arquitetura ARM, Acesso em 18/04/2014, <<http://pt.slideshare.net/LuisFernandoMarques/histria-da-arquitetura-arm-novo#>>.

7. NOTA DE RESPONSABILIDADE

O autor é o único responsável pelo conteúdo deste artigo.