

# Reconf\_KMT, Uma Ferramenta Reconfigurável para a Simulação de Microprocessadores

Talles Henrique de Medeiros, Carlos Augusto P. S. Martins  
PUC Minas - Pontifícia Universidade Católica de Minas Gerais  
LSDC - Laboratório de Sistemas Digitais e Computacionais  
Av. Dom José Gaspar, 500 Coração Eucarístico  
Belo Horizonte – MG – Brasil  
talles@pucmg.br, capsm@pucminas.br

## Resumo

*Neste trabalho apresenta-se uma ferramenta de simulação reconfigurável de microprocessadores com o objetivo principal de ser uma ferramenta para auxílio ao ensino de arquitetura de processadores em cursos de Ciência da Computação e afins. Este simulador apresenta o conceito de reconfigurabilidade como grande vantagem para o desenvolvimento de um simulador para uma arquitetura de processador sem a necessidade de sua programação por completo. O simulador é totalmente implementado em Java permitindo sua utilização em diversas plataformas hardware/software. O simulador reconfigurável é o grande resultado deste trabalho sendo o mesmo validado através da comparação da simulação com um outro simulado de funcionamento reconhecidamente comprovado.*

## 1. Introdução

Os microprocessadores estão presentes nos mais diversos equipamentos imaginados, vão desde os supercomputadores até os mais comuns equipamentos eletrônicos domésticos como por exemplo videocassetes, forno microondas, etc. São hoje objetos de estudo de grande importância e temas centrais de diversas pesquisas em todo mundo. Isso pode ser percebido através do histórico da evolução das arquiteturas e das tecnologias embutidas nos microprocessadores ao longos nos anos [2].

Baseando-se no fato do grande universo de aplicação dos microprocessadores, estudar e pesquisar a sua estrutura interna é de enorme importância tanto para poder explorar da maneira mais otimizada possível seus recursos quanto para o domínio da tecnologia que pode ser útil para a continuidade do desenvolvimento científico da arquitetura dos processadores.

Diante da importância dos estudos dos microprocessadores, a técnica de simulação ganha uma

importância considerável no contexto de ensino e aprendizado nas universidades. A definição de simulação pode ser entendida como a técnica da resolver problemas seguindo as variações, ao longo do tempo, ocorridas num modelo dinâmico do sistema. Com a simulação pretende-se abordar dois grandes aspectos que são a modelagem e a experimentação, e alcançar outros dois grandes objetivos que são a compreensão e prescrição do comportamento do sistema simulado.

Um dos grandes desafios na simulação é exatamente saber quando seu uso é indicado. Pode-se justificar o uso da simulação em processos quando o custo para a construção do projeto é inviável, quando se deseja simplesmente efetuar um treinamento ou então no auxílio a compreensão e análise de sistemas existentes. É nesta última justificativa que este trabalho possui seu maior enfoque. Diante disto a simulação é atualmente a técnica adotada em diversas disciplinas dos cursos de computação, em especial a disciplina de Arquitetura de Computadores, para auxiliar no processo de ensino e aprendizado dos modelos estudados como por exemplo os processadores, o pipeline, a hierarquia de memória, etc. A grande vantagem da simulação neste caso é que dentro do contexto de uma disciplina do curso de graduação não é simples o uso de sistemas reais para verificação dos projetos dos alunos. Com o uso do mecanismo de simulação este problema da verificação é facilmente resolvido com a implementação de um simulador em software. Uma das ferramentas mais populares e eficientes utilizadas na simulação é o software de simulação que é capaz que representar o sistema real com o grau de detalhamento que for desejado.

### 1.1 Motivação

Todo o processo de implementação do simulador de um microprocessador é uma tarefa exaustiva, além disso existe o fato de que em boa parte das vezes o próprio projetista da arquitetura é o responsável pela implementação do simulador da arquitetura. O fato é que desenvolver um simulador por completo para cada

arquitetura que for especificada e projetada é um trabalho que de maneira alguma é o desejado. Até mesmo se fosse necessário alguma adaptação na arquitetura simulada, seria necessário a modificação do código fonte do simulador. Isto implicaria num esforço que iria além do escopo desejado do usuário do simulador. Neste caso o projetista, que não deveria e nem quer ser o responsável pela da implementação do software. Estes fatores acabam por complicar o processo de verificação de um processador, tomando um grande tempo que poderia ser utilizado para outras atividades de maior importância para o projetista do processador. Diante dos problemas que toda a implementação de um simulador pode apresentar iniciou-se então um estudo para que pudessemos encontrar um método que fosse eficiente o suficiente para poder simular com precisão o processador proposto e projetado, e que também fosse simples o bastante de modo que o projetista tivesse seu trabalho de implementação poupado. A idéia proposta para a solução deste problema é um simulador que pudesse ser capaz de ter seus parâmetros configurados sempre que uma simulação for executada. O modo descoberto para permitir a configuração dos parâmetros fazia parte dos desafios enfrentados na pesquisa. Isto deixa claro o objetivo principal deste trabalho que é a disponibilidade de um software reconfigurável para simulação de microprocessadores, que a princípio tem a finalidade de servir como ferramenta de auxílio no ensino da arquitetura dos microprocessadores [8].

## 2. Processadores

A história do desenvolvimento dos processadores não é antiga, nascem junto a criação do primeiro computador, o ENIAC (Eletronic Numerical Integrator and Calculator), construído por John Von Neumann em 1946. O processador do ENIAC era totalmente discreto, as suas unidades funcionais eram fisicamente separadas e sua comunicação era feita por fios soldados manualmente. Devido a criação dos circuitos integrados pôde-se então, em 1971, que a Intel Corporation desenvolvesse através de Ted Hoff, um processador que reunia todas as suas unidades funcionais em um único chip. Este ficou conhecido como o 4004, o primeiro microprocessador. Isto iniciou a produção em grande escala destes processadores, o que consequentemente culminou por reduzir drasticamente o seu custo. Em 1974, a própria Intel anunciou o 8080, de 8 bits e atingia 5Mhz de velocidade de *clock*, mas apesar de suas características não foi utilizado a ponto de ser um marco na história dos processadores. Em 1975, surgiu então o Zilog 80, que era um melhoramento do 8080. O Z-80 funcionava com um barramento de endereços memória de 16 bits e o

barramento de dados de 8 bits. Em 1978, surgiu o 8086, com barramento de 16 bits o que permitia desde logo controlar uma maior quantidade de memória, e com registradores de 16 bits, o que permitia fazer cálculos mais complexos, como por exemplo, operações com reais de ponto flutuante, muito mais facilmente. Em 1982 surgiu o 80286. As suas principais características eram o modo protegido, que permitia ultrapassar o problema de endereçamento de memória, e as velocidades de 8 e 12 MHz. Mas esta característica só alguns anos mais tarde viria a ser usada em larga escala, pelo que a sua compatibilidade com o 8086/88 e as suas velocidades de processamento foram os principais fatores que fizeram dele um *best seller*. O 80386 apareceu em 1985, inicialmente a velocidade de 16 MHz, e mais tarde de 20, 25, 33, 40, ..., e executava 5-6 MIPS e era 15 vezes mais potente que 8086. Foi o primeiro processador que trabalhava com 32 bits. Em seguida, em 1989 surgiu o 80486, a velocidades iniciais de 20, 33, 50 MHz, e executava 20 MIPS. Surgiu nele o conceito de pipeline, que permitia a execução de mais de uma instrução simultaneamente. Isso fez com que mesmo com um clock igual ao do 80386, ele conseguisse um desempenho superior. A especificação do 486 pressupunha também a existência de um coprocessador matemático (FPU) integrado, o que em versões menores veio a ser *desligado*. Continha também 8KB de *memória cache* quer para dados quer para instruções. Em 1993 surgiu o Intel Pentium [3]. Tem capacidade de endereçamento de 32 bits e barramento de 64 bits, tem duplo cache interno, unidades de cálculo para inteiros e reais, tecnologia superescalar e outros. Em seguida foram surgindo extensões deste processador com pequenos acréscimos como o Pentium, Pro, II, III e IV. Como opção de processadores de baixo custo a fabricante AMD também desenvolveu seus processadores K6-2, Duron, Athlon.

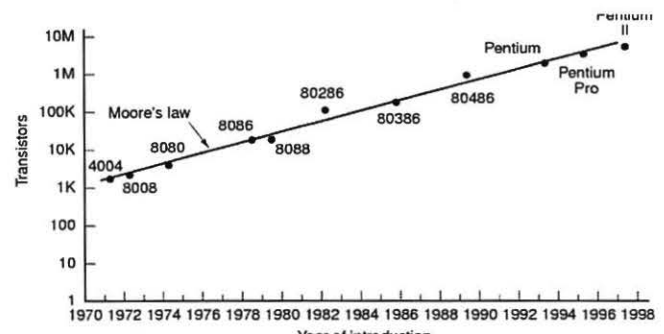


Figura 2.1 – Número de transistores da cada processador lançado

Como mostra a figura 2.1 o número de transistores foi crescendo muito a medida que cada novo microprocessador era lançado no mercado. Pode notar o

# PÁGINA FALTANDO

Esse trabalho foi disponibilizado online como parte do esforço de publicação de todo o histórico do evento, entretanto, a versão do trabalho recuperada não continha essa página.

# PÁGINA FALTANDO

Esse trabalho foi disponibilizado online como parte do esforço de publicação de todo o histórico do evento, entretanto, a versão do trabalho recuperada não continha essa página.

estiver o arquivo de reconfiguração que utiliza esta instrução para que o simulador a encontre no momento em que o simulador tentar instanciar a classe da instrução que foi reconhecida no arquivo. Deste modo, o usuário poderá definir para o simulador qualquer instrução que não faça parte das instruções padrão embutidas no simulador. Uma grande vantagem que se obteve ao utilizar a programação orientada a objeto no desenvolvimento do software foi a herança entre classes, principalmente quando se deseja implementar uma nova instrução que na verdade seja uma extensão da uma instrução mais simples já implementada anteriormente.

Após passada a mais importante etapa do simulador, o desenvolvimento do mecanismo de configuração, passa-se então a voltar a atenção as limitações que antes tiveram que ser impostas para alcançar os primeiros resultados. O fim desta etapa possibilitou o fechamento da primeira versão do *Reconf\_KMT* utilizada para os testes neste trabalho.

## 5. Resultados

Para apresentação da ferramenta desenvolvida serão exibidas algumas das telas que o simulador apresenta ao seu usuário. O exemplo que está sendo utilizado neste trabalho para apresentação do simulador e para sua verificação é um simulador desenvolvido na própria Universidade durante a disciplina de arquitetura de computadores. Este simulador é simples e rápido de ser implementado pela nova ferramenta e já foi testado o suficiente de modo que seus resultados podem ser comparados com os resultados apresentados pelo simulador reconfigurável.

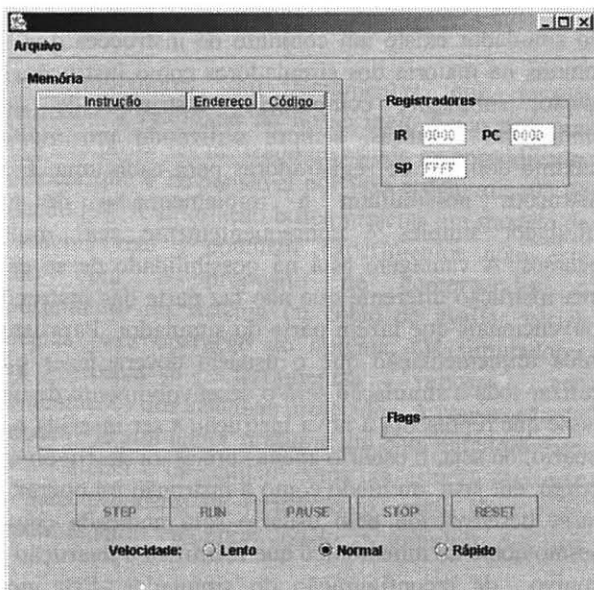


Figura 2 – Simulador com sua tela principal.

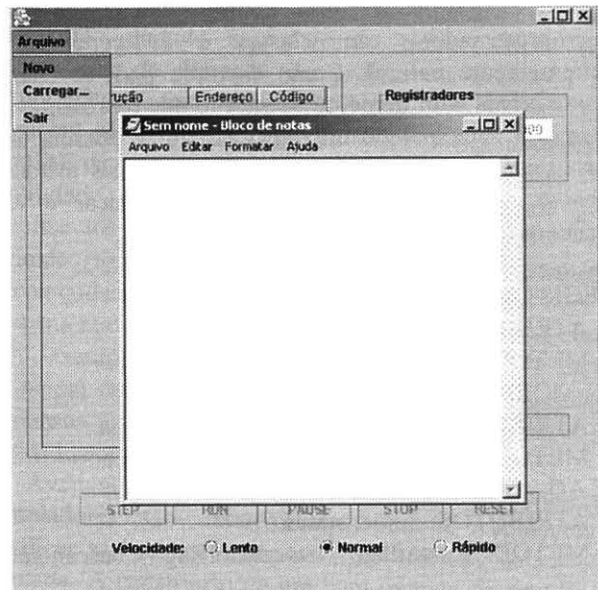


Figura 3 – Área de edição do programa no assembly da arquitetura desejada..

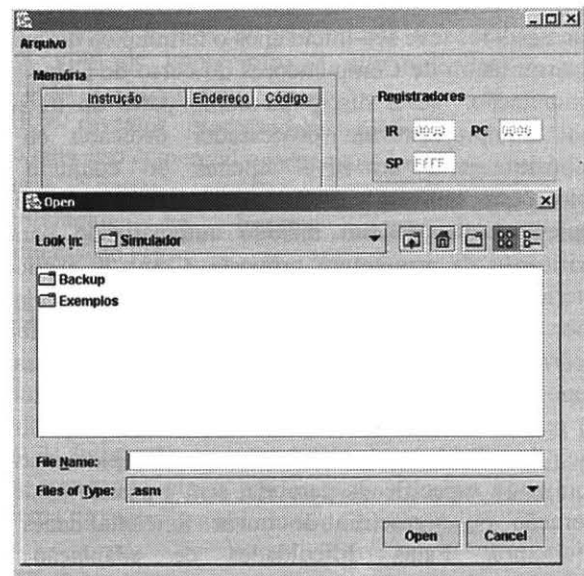


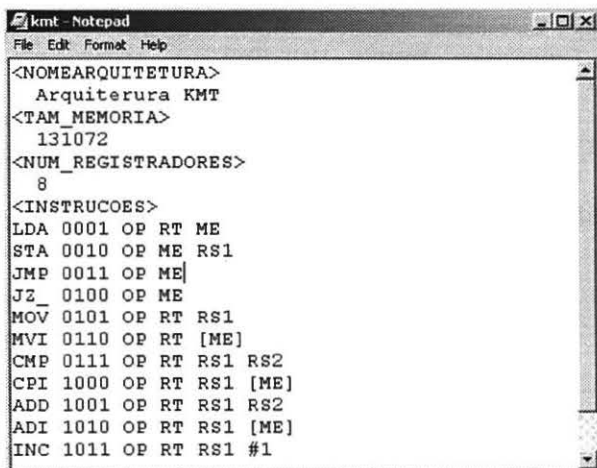
Figura 4 – Componente Swing padrão de Java para abrir um arquivo.

### 5.1 Validação do Simulador

Para se garantir a confiança dos resultados apresentados pelo simulador precisava-se avaliar comparativamente com algum outro simulador. Para isto utilizou-se o KMT, um simples simulador de 16 bits com memória principal de 128Kbytes e 15 instruções do tipo LOAD – STORE [7]. Um pequeno arquivo de teste utilizado no KMT foi semelhante ao utilizado pelo *Reconf\_KMT* para que o comportamento fosse analisado passo a passo. A necessidade de algumas mudanças dos



arquivos era necessária pela convenção utilizada nos nomes das instruções que o simulador reconfigurável interpreta no arquivo do programa assembly.



```

kmt - Notepad
File Edit Format Help
<NOMEARQUITETURA>
Arquiterura KMT
<TAM_MEMORIA>
131072
<NUM_REGISTRADORES>
8
<INSTRUCOES>
LDA 0001 OP RT ME
STA 0010 OP ME RS1
JMP 0011 OP ME
JZ_ 0100 OP ME
MOV 0101 OP RT RS1
MVI 0110 OP RT [ME]
CMP 0111 OP RT RS1 RS2
CPI 1000 OP RT RS1 [ME]
ADD 1001 OP RT RS1 RS2
ADI 1010 OP RT RS1 [ME]
INC 1011 OP RT RS1 #1

```

Figura 5 – Arquivo de configuração da arquitetura do KMT

O principal resultado alcançado durante toda a pesquisa é na verdade o próprio software de simulação, o *Reconf\_KMT*, desenvolvido para simular, de cada vez, em um único software simulador diversos processadores através do uso de um arquivo de configuração da arquitetura do microprocessador com seu conjunto de instruções. O que definiu a exatidão dos resultados obtidos foi a validação do simulador reconfigurável com o simulador normal de uma determina arquitetura de microprocessador, como foi explicada no sub – item 5.1.

O uso do *Reconf\_KMT* trouxe para o usuário um ganho de tempo considerável para implementação do seu simulador, uma vez que com esta ferramenta todo esforço fica concentrado somente na criação do arquivo de reconfiguração. Uma comparação do uso da ferramenta com o método normal pôde ser feita e verificou – se o quanto se ganhou de tempo já que o desenvolvimento do simulador por completo consumiu mais de um dia para ser implementado e testado enquanto que com a ferramenta reconfigurável é possível ver o resultado da simulação após alguns minutos ou horas gastos somente com o desenvolvimento do arquivo de reconfiguração. Esta agilidade obtida com o *Reconf\_KMT* se deve também a facilidade que é o desenvolvimento do arquivo com os parâmetros de configuração da arquitetura do processador. Desenvolver um arquivo de reconfiguração não é grande problema para o usuário uma vez que sua estrutura é extremamente simples e que pode ser reutilizada através de alguns arquivos exemplos que já existem. Este software traz para o ensino em computação uma poderosa ferramenta de auxílio ao aprendizado de arquitetura de microprocessadores para alunos de graduação ou até mesmo de pós – graduação.

## 6. Conclusão e Trabalhos Futuros

Para se chegar a conclusão final deste trabalho, voltamos a ressaltar que o objetivo principal deste trabalho era o de conseguir um software capaz de simular qualquer processador sem a necessidade de desenvolver um simulador para cada processador. Em segundo plano, conseguir um mecanismo simples para o usuário reconfigurar o simulador a medida que for necessário. Com isso pode – se considerar que ambos objetivos foram alcançados com sucesso. Diante dos resultados apresentados pela simulação do processador KMT no simulador reconfigurável e verificando a validade dos resultados obtidos pôde – se comprovar a eficácia do método implementado para a reconfiguração da ferramenta. Como se trata de um trabalho complexo e com pouquíssimas referências sobre outras pesquisas semelhantes, o grau de dificuldade para seu desenvolvimento em um projeto de iniciação científica, contexto no qual foi desenvolvido, não pode ser deixado de lado e por isso o software ainda pode e deverá apresentar versões mais flexíveis quanto a reconfigurabilidade do conjunto de instruções e também a possibilidade do próprio usuário pode definir a interface do seu simulador. Assim sendo um trabalho com tão poucos semelhantes, os resultados alcançados com a ferramenta e seu mecanismo de configuração podem representar uma importante contribuição no auxílio ao desenvolvimento de simuladores no ensino de microprocessadores.

## 7. Agradecimentos

Agradeço a PROPPG da PUC Minas pelo financiamento desta pesquisa e aos amigos do LSDC pela colaboração em algumas etapas deste projeto.

## 8. Referências

- [1] Dwight, S., Glasco, D., Flynn M., “Absv2.0: A Sparc Simulator”, Technical Report – Stanford University, April 1998.
- [2] Anceau, François, “Past, Present And Future Of Microprocessor”, IX IFIP – International Conference On Very Large Scale Integration, Gramado - RS - Brasil, 26-29 de Agosto, 1997.
- [3] Antonakos, J. L., “The Pentium Microprocessor”, Prentice Hall, 1997.
- [4] Arnold, Ken E Gosling, Jones, “Programando em Java”, Makron Books, 1997.
- [5] Hennessy, John L. and Patterson, David A., “Computer Architecture A Quantitative Approach”, Morgan Kaufmann, 1996.

- [6] Mangan, M. A. S., Vargas, P.K., Azzolin, D., "Técnicas Para Desenvolvimento De Aplicações Orientadas A Objetos Utilizando Linguagem Java", III Simpósio Brasileiro De Linguagens De Programação, Porto Alegre - RS, 1999.
- [7] Medeiros, Talles H., Corrêa, João B. T., Martins, Carlos Augusto P. S., "Projeto e Desenvolvimento de um Software de Simulação de um ASIP no nível ISA", Anais WSCAD 2001 págs. 146 - 149, Pirenópolis - GO - Brasil, Setembro 2001.
- [8] Medeiros, Talles H., Martins, Carlos A. P. S., "Proposta e implementação de um simulador reconfigurável de microprocessadores", Projeto PROBIC P2001/112, LSDC - PUC Minas, Belo Horizonte - MG - Brasil.
- [9] Soares, Luiz Fernando G., "Modelagem e Simulação Discreta de Sistemas", Editora Campus, 1992.
- [10] Tabak, Daniel, "Advanced Microprocessor", McGraw – Hill, 1995.
- [11] Tabak, Daniel, "Advanced Microprocessor", McGraw – Hill, 1995.
- [12] Tanenbaum, Andrew S., "Structured Computer Organization", Prentice - Hall, 1999.
- [13] Tocci, Ronald J., "Sistemas Digitais, Princípios e Aplicações", Prentice - Hall do Brasil, 1994.
- [14] Yourdon, E., Argila, Carl, "Análise a Projeto Orientados a Objetos", Makron Books, 1999.
- [15] CPU Info Center: <http://bwrc.eecs.berkeley.edu/CIC>