Luiz Gustavo de Souza Carvalho

Métricas e Análise de Desempenho da Unidade Central de Processamento

Unipac
Barbacena/MG
2003

Luiz Gustavo de Souza Carvalho

Métricas e Análise de Desempenho da Unidade Central de Processamento

Monografia apresentada como exigência parcial para a obtenção do Título de Bacharelado em Ciência da Computação à Banca Examinadora da Universidade Presidente Antônio Carlos, Campus Magnus, elaborada sob a orientação do Professor Eduardo Bhering.

Unipac
Barbacena/MG
2003

Este trabalho de conclusão de curso foi julgado adequado à obtenção do grau de Licenciado em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciência da Computação da Universidade Presidente Antônio Carlos.

Barbacena, 4 de dezembro de 2003.

_____

Prof. Eduardo Macedo Bhering

_____

Prof. Emerson Rodrigo Alves Tavares

_____

Prof. Eliseu César Miguel

Agradeço a Deus por me amparar plenamente durante todo o tempo, hoje e sempre, fortalecendo as minhas expectativas e meus sonhos, para que, no final, tudo dê certo!

Aos meus pais, que viabilizaram a maior riqueza que hoje possuo; o conhecimento.

Aos professores, que ampliaram nossas visões e nos ajudaram durante toda caminhada.

Ao meu amigo, Alessandro Martin Dinalli, que sempre me auxiliou nas mais diversas questões.

A todas as pessoas que me acompanharam até o presente momento, amigos e irmãos, e aquelas que ainda hão de vir...

A meu orientador, pela disposição a qual orientou o meu estudo.

Luiz Gustavo S. Carvalho

O tempo revela a verdade

Sêneca
Ensaios sobre a moralidade, 22 a.D.

LISTA DE GRÁFICOS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

3DNow!	Multi-Media Extensions (AMD)
AGP	Accelerated Graphics Port
AMD	Advanced Micro Devices, Inc.
BIOS	Basic Input / Output System
CISC	Complex-Instruction-Set-Computing.
CPI	Clock cycles per Instruction.
CPU	Central Processing Unit.
GFLOPS	Billions of Floating Point Operations Per Second (GigaFlops)
MFLOPS	Millions of Floating Point Operations per Second.
MHz	MegaHertz
MIPS	Millions of Instruction per Second.
MOPS	Millions of Operations Per Second
MMX	Multi-Media Extensions
MWIPS	Millions of I instruction Whetstone per Second.
PC	Personal Computer.
RISC	Reduced-Instruction-Set-Computing
SMD	Surface Mount Device
SPEC	System Performance Evaluation Cooperative.
SSE	Streaming SIMD Extensions
SSE2	Streaming SIMD Extensions 2

RESUMO

O objetivo do presente estudo está focado em analisar a Unidade Central de Processamento de duas fabricantes de processadores, AMD e INTEL, medindo o desempenho dos mesmos quanto a cálculos aritméticos. Foram analisados os principais processadores do mercado e usados dois softwares benchmarks: o *Sisoftware* SANDRA 2004, que é um programa capaz de efetuar esses testes nos microcomputadores da linha IBM-PC e um outro programa específico para testar a UCP foi desenvolvido, e é chamado de WinBenchGLC. Os resultados mostram que a arquitetura do ATHLON é mais eficiente tanto em cálculos aritméticos e manipulação de *strings* quanto em operações que envolvem ponto flutuante.

SUMÁRIO

1 Introdução

Os arquitetos de computadores geralmente alteram o projeto de determinada arquitetura, muitas vezes, levando em conta o fator desempenho. Dependendo do projeto a ser construído ele poderá aumentar a performance em determinados pontos estratégicos, se apoiando em programas que lhe auxiliem nesta tarefa. Para esta comunidade é interessante o uso de programas que possam medir (quantificar) o desempenho de determinados elementos do sistema e de testar habilidades do processador para cálculos de pontos flutuantes e inteiros. Isso possibilita balancear os custos de um projeto, obtendo uma relação custo / benefício equilibrada.

Outro importante aspecto é o consumidor final de computadores. É importante que na escolha de determinado microcomputador, o usuário faça uma comparação entre os disponíveis no mercado levando em conta, principalmente, o parâmetro desempenho. Entretanto, na maioria das vezes esta comparação é feita com base em dados gerais disponibilizados pelos fabricantes e em informações fornecidas pelos revendedores — informações estas que na maioria das vezes são superficiais e que nem sempre consideram a necessidade real do consumidor.

O objetivo deste trabalho é realizar um estudo de métodos de análise de desempenho da UCP (Unidade Central de Processamento) e suas métricas, evitando conclusões distorcidas, centradas em informações subjetivas do consumidor. Um programa benchmark foi desenvolvido para testar as principais habilidades do processador (UCP) quanto à aritmética de inteiros e ponto flutuante e é chamado de WinBenchGLC. Os resultados serão reportados se baseando no tempo de execução dos módulos e será usado um software de domínio público: o SisSoftware Sandra 2004. Duas arquiteturas de processamento atualmente muito usadas foram testadas: AMD Athlon XP e Intel Pentium 4, sendo que o Athlon ganhou nos testes de performance.

Os métodos para avaliação de desempenho constituem uma área de pesquisa importante no desenvolvimento de arquitetura de computadores. Por outro lado, o crescimento acelerado da tecnologia e o aumento da complexidade dos sistemas computacionais fazem da avaliação de desempenho uma questão não muito simples de ser resolvida [1].

Uma forma de avaliar o desempenho de um sistema computacional é através de ferramentas capazes de quantificar em dados o desempenho de um computador. Essas ferramentas são conhecidas como benchmarks, e são programas desenvolvidos especialmente para trabalhar na medida de desempenho. Através de um benchmark, é possível determinar se um sistema em particular pode processar adequadamente uma carga real, se as características de custo e desempenho são adequadas para um determinado problema e quais as melhores opções de configuração de hardware e software, considerando se um sistema em particular [2].

Além disso, os benchmarks podem ser utilizados na avaliação de sistema já em operação. Neste caso deseja-se verificar se uma alteração em um determinado parâmetro do sistema, seja de hardware ou de software, resultará na melhoria do desempenho. É preciso que haja condições de submeter o benchmark ao sistema alterado antes de consolidar definitivamente a modificação. Pois existem situações

em que não é possível interromper a operação de um sistema, para isso, uma solução é disponibilizar um outro sistema no qual possam ser efetuados os testes antes de alterar o sistema em operação. Toda e qualquer atividade envolvida no processo computacional pode e deve ser medida e avaliada, a fim de que se possa ter a certeza que ela é adequada à aplicação para a qual foi projetada e que corresponde às expectativas em termos de eficiência e confiabilidade.

Existem várias medidas para quantificar o desempenho de um sistema computacional, as métricas utilizadas são relacionadas a aspectos históricos envolvidos. Até o final dos anos 80, as medidas dominantes de desempenho de computadores eram MIPS (milhões de instruções por segundo) e MFLOPS (milhões de operações de ponto-flutuante por segundo). Essas medidas tiveram origem na percepção do conjunto de instruções que representava a carga de trabalho de um sistema. Entretanto, medidas baseadas no conjunto de instruções revelam possuir um critério não muito adequado para a comparação de desempenho, devido às diferenças existentes nos conjuntos de instruções dos diversos processadores. Além do mais, o aspecto de desempenho em relação ao conjunto de instruções representam um critério do uso de benchmark voltado somente as operações que envolvem UCP. A evolução das famílias RISC (reduced-instruction-set-computing) e CISC (complex-instruction-setcomputing)[1] mostram que medidas de MIPS e MFLOPS são inadequadas para indicar com precisão às métricas de desempenho [1].

Um benchmark é constituído por cargas naturais e cargas artificiais. Cargas, no contexto, são módulos que executam alguma computação. Elas podem ser naturais ou artificiais. As cargas naturais correspondem exatamente à carga real, onde estas são observadas nos sistemas em operação normal utilizando um determinado aplicativo real. Já as cargas artificiais podem ser de duas espécies: as

[1] RISC são arquiteturas com o conjunto de instruções reduzidas e CISC o conjunto de instruções são complexas.

cargas sintéticas, projetadas e implementadas sem nenhum compromisso em utilizar cargas reais, e as cargas híbridas, que se caracterizam em possuir partes de uma carga real.

De um modo geral, as cargas naturais necessitam de períodos de medição muito longos para que se possa garantir um alto grau de representatividade, o que não ocorre com as cargas artificiais, pois são projetadas adequadamente apara analisar um certo aspecto do sistema (análise de UCP, periféricos etc.) permitindo aplicações repetitivas e controladas. Isso constitui uma das razões básicas pela qual as cargas artificiais são geralmente preferidas. Entre as cargas artificiais, as híbridas são, aparentemente, de obtenção menos dispendiosa que as sintéticas. Estas, entretanto, constituem os chamados benchmarks sintéticos, são mais flexíveis e, uma vez cuidadosamente ajustadas e validadas, permitem conseguir uma melhor representatividade em um período de testes mais curto.

Todavia, os profissionais da computação sempre quiseram estabelecer critérios de avaliação de um sistema. Dois benchmarks comuns para medir o desempenho de computadores são: os do tipo kernel e os sintéticos, usados como representações de carga de trabalho reais para computadores. Programas kernel são segmentos de código extraídos de aplicações reais e altamente representativas (inversão de matrizes e busca em árvores, por exemplo), enquanto programas sintéticos são programas gerados artificialmente para tentar retratar as características de um grande conjunto de programas. Os testes mais comuns de benchmark baseados em kernel ou programas sintéticos incluíam os testes Dhrystone, Whetstone, Linpack e Livermore Loops, que são benchmarks comuns. MIPS foi derivado do Dhrystone e o MFLOP foi largamente influenciado pelo Linpack, apesar da contribuição do Livermore Loops [3].

Um outro conjunto de benchmark’s muito conhecido chama-se System Performance Evaluation Corporative (SPEC). O SPEC é uma cooperativa formada por grandes fabricantes de computadores que buscam desenvolver, manter e apoiar

um conjunto padronizado de benchmarks que possam ser aplicados a mais nova geração de computadores de alto desempenho. O SPEC não é exatamente o nome de um benchmark. Na verdade, trata-se de um consórcio de vinte e duas empresas que têm como objetivo prover a indústria com um padrão realístico para medir o desempenho de sistemas de computadores avançados [2]. Os benchmarks desenvolvidos por esta cooperativa de empresas são programas reais (o que os classifica como benchmarks de aplicação), que visam medir o desempenho da maior parte possível de componentes do sistema, principalmente o processador e a arquitetura de memória.

O que se tem de mais recente no conceito de benchmark é o acoplamento de um hardware ao processador, monitorando o seu desempenho. Os dados coletados por esse hardware fornecem informações de desempenho das aplicações, do sistema operacional e do processador. Os dados podem guiar os esforços de obtenção de desempenho fornecendo informações que ajudam os programadores a melhorar os algoritmos utilizados pelas aplicações, o sistema operacional e as seqüências de código que implementam esses algoritmos [1].

Este trabalho de conclusão de curso, objetiva especificamente analisar processadores dos fabricantes Intel e AMD, buscando identificar aquele que possui melhor desempenho em termos da performance da UCP, medido sobre o tempo de execução, sobre alguns módulos implementados em um pequeno software WinBenchGLC. Assim como utilizar um software benchmark de terceiros para auxiliar nos testes. Para alcançar esses objetivos, foi realizado um estudo que consiste de três fases. A divisão é necessária para uma análise detalhada dos principais programas que realizam testes de desempenho, e consistem em:

a) Escolha de programas capazes de efetuar todos os testes de desempenho nos microcomputadores da linha IBM-PC. Desenvolvimento de um pequeno software benchmark para analise e teste de desempenho de processadores quanto à capacidade de cálculos com inteiro e ponto flutuante.

b) Realização dos testes de desempenho considerando-se várias configurações;

c) Apresentação dos resultados do desempenho de cada processador.

O trabalho foi estruturado em seis capítulos — o primeiro representado por esta introdução, apresenta os objetivos principais, estado da arte em que se encontra a área. O segundo capítulo apresenta os fundamentos teóricos, necessários para a compreensão do trabalho e discute os aspectos gerais envolvidos na análise de desempenho de UCPs. O terceiro capítulo focaliza os principais e mais comuns benchmarks encontrados na literatura computacional, dando maior ênfase ao benchmark SPEC. O quarto capítulo, por sua vez, traz uma aplicação dos testes de benchmark realizados nos computadores Pentium 4 e Athlon XP, a partir de certos parâmetros de comparação, da escolha da configuração, da confecção de um programa WinBenchGLC e apresentação dos resultados obtidos. A conclusão constitui o quinto capítulo e aborda as principais constatações resultantes dos testes, comentando a perspectiva de trabalhos futuros na área. O sexto capítulo descreve a bibliografia consultada.

2 Fundamentos Teóricos

2.1 Métricas de Desempenho

A medida geral de desempenho de um sistema de computação depende fundamentalmente da capacidade e velocidade de seus diferentes componentes, da rapidez com que estes componentes se comunicam entre si e do grau de compatibilidade que possa existir entre eles. Um exemplo é a velocidade da UCP de um sistema ser muito maior que a da memória principal causando um comprometimento no desempenho global.

Considerando a existência de tantos fatores que influenciam o desempenho de um sistema de computação, desenvolveram-se diversas métricas. O desempenho dos processadores, em geral, é medido em termos da sua velocidade de trabalho. Como seu trabalho é executar instruções, criaram-se as unidades chamadas MIPS (milhões de instruções por segundo) e MFLOPS (milhões de operações de ponto flutuante por segundo), esta última uma medida típica de estações de trabalho e de supercomputadores, pois estes costumam trabalhar com cálculos matemáticos. Já quando se trata de recuperação ou escrita de informações na memória, o tempo de acesso é uma unidade de medida mais apropriada,

estando relacionada à velocidade de cada componente e a do canal de interligação entre a UCP e a memória [2].

Tempo de resposta ou tempo de execução é uma medida ligada ao desempenho mais global do sistema. Trata-se do período de tempo gasto entre o instante em que o usuário iniciou uma solicitação ou interrupção e o instante em que o sistema apresentou ao usuário a sua resposta ou atendeu à sua solicitação. A medida padrão é o segundo (s), mas neste trabalho foi adotado o milésimo de segundo, para uma maior precisão. Como exemplo, pode-se citar o intervalo de tempo entre a solicitação de um saldo de conta em um terminal bancário e a apresentação, no vídeo, da resposta, o saldo da conta.

Uma outra unidade de medida de desempenho é a vazão ("throughput"), que define a quantidade de ações ou transações que podem ser realizadas por um sistema na unidade de tempo, por exemplo, pode-se mencionar a quantidade de atualizações que são realizadas em um sistema de controle de estoque de uma empresa.

Quando se faz referência à velocidade com que um determinado dispositivo de entrada ou de saída transfere ou recebe dados da UCP, é utilizada uma unidade que mede a taxa de transferência que o canal de ligação pode suportar, isto é, a quantidade de bits por segundo que podem trafegar por aquele canal [5].

2.1.1 MIPS

No contexto de medidas de desempenho de UCP, MIPS significa Milhões de Instruções por Segundo. A métrica MIPS de uma UCP se refere ao número de instruções de código de máquina que um processador pode executar em um segundo. Infelizmente, usar este número como a única maneira de medir desempenho de processador é algo completamente ineficiente, já que dois

processadores individuais nunca usam o mesmo tipo de instruções, métodos de execução etc. Por exemplo, em um processador, uma simples instrução pode realizar várias tarefas quando executadas como se observa na família CISC, enquanto em outro processador uma simples instrução pode fazer muito pouco, mas é mais rápida de ser executada como ocorre na família RISC. Além disso, instruções diferentes no mesmo processador quase sempre significam cargas muito diferentes de trabalho (por exemplo, uma simples instrução aritmética pode levar apenas um ciclo de relógio (relógio) para ser completada, enquanto fazer algo como uma divisão de ponto flutuante ou uma operação de raiz quadrada pode levar de 20 a 50 ciclos de relógio).

Os especialistas que desenvolvem processadores e as pessoas interessadas em saber como eles funcionam quase nunca utilizam a taxa de MIPS quando discutem desempenho. O número de MIPS de uma máquina é geralmente muito alto em decorrência da maneira como trabalham os processadores, porém, na verdade, o número de MIPS geralmente diz muito pouco sobre como o processador realmente funciona. Em um processador hipotético com uma taxa de MIPS menor que a de outro pode, na realidade, ser um processador melhor no aspecto de velocidade, uma vez que suas instruções teriam condições, por exemplo, de realizar mais trabalho por ciclo de relógio do que um outro com uma taxa maior [2]. Vejamos uma ilustração: imagine um processador de 32 bits rodando a 400 MHz, ele pode possuir uma taxa de 400 MIPS. Já um processador de 64 bits rodando a 200 MHz pode ter uma taxa de 200 MIPS (assumindo um design simples em cada caso). Se a tarefa a ser realizada pelos processadores envolvesse processamento de ponto-flutuante de 64 bits (processamento de áudio, por exemplo), o processador de 32 bits levaria muito mais ciclos de relógio para completar uma simples multiplicação de ponto-flutuante de 64 bits, já que os seus registradores têm comprimento de palavra de 32 bits apenas. A UCP de 32 bits levaria pelo menos o dobro do tempo para realizar tal operação. Então, para operações de 64bits, o processador de 32 bits seria muito mais lento do que o de 64 bits. Em compensação, se a tarefa envolvesse operações de 32 bits, se os registradores de 64 bits, na UCP de 64 bits,

pudessem ser tratados como dois registradores de 32 bits, a UCP de 32 bits seria muito mais rápida. Tudo dependeria dos requerimentos de processamento.

A situação na vida real é, entretanto, muito mais complicada, pois UCPs reais raramente realizam tarefas uma de cada vez em apenas um ciclo de relógio. Operações aritméticas simples podem levar apenas um ciclo de relógio, uma multiplicação de inteiros pode levar dois ciclos, uma multiplicação de ponto-flutuante pode levar cinco ciclos, e assim por diante. Além do mais, algumas UCPs foram projetadas para fazer mais do que uma operação do mesmo tipo ao mesmo tempo, por exemplo, quando possuem mais de uma unidade de processamento [5].

As UCPs modernas (entre elas as séries R10000 e PA8000) na maioria das vezes possuem duas ou mais unidades de processamento aritmético para inteiros, duas ou mais unidades de processamento de ponto-flutuante e pelo menos uma unidade de load/store (instruções de máquina). Às vezes, elas também podem ter unidades especiais para acelerar, por exemplo, cálculos de raiz quadrada. Atualmente, há tecnologias tais como a MMX, da Intel que foi desenvolvida para permitir que um registrador aritmético de 64 bits seja tratado como múltiplos registradores de 32 bits, 16 bits ou 8 bits. Existe ainda a MDMX (da MIPS Technologies Inc.) que faz o mesmo, mas é mais poderosa devido ao fato de também permitir a mesma divisão do registrador para ser feita com registradores de ponto-flutuante [2].

Essas novas idéias permitem que um número maior de cálculos seja realizado simultaneamente em comparação a arquiteturas mais velhas. Um exemplo: o sombreamento Gouraud, que envolve operações de ponto-flutuante de 32 bits, usando um registrador de 64 bits de ponto-flutuante como dois registradores de 32 bits “separados” resulta na melhor das hipóteses, no dobro da capacidade de processamento da UCP.

A métrica MIPS pode ser definida como [2]:

$$MIPS = \frac{taxa\_clock}{CPI \times 10^6}$$

Onde CPI (Ciclos de Clock por Instrução) é a media do tempo gasto por todas as instruções executadas pelo programa e a taxa de relógio é a média de ciclos de relógio por segundo de um microcomputador expressado em Mega Hertz (1 MHz sendo igual a 1 milhão de ciclos por segundo).

A fórmula mostra que, quanto maior a taxa de relógio, maior o desempenho da máquina. E, quanto menor for o CPI, ou melhor, ciclos de relógio por segundo (o valor mínimo é 1), maior será o desempenho medido em MIPS.

O MIPS tem a grande vantagem de ser fácil de entender e bastante intuitivo quando se analisa a relação dos fatores envolvidos na fórmula. Entretanto existem alguns problemas quando se utiliza MIPS para comparar o desempenho de duas arquiteturas diferentes. O primeiro problema é que MIPS especifica a taxa de execução de instruções de forma independente do conjunto de instruções. Não é razoável comparar máquinas com diferentes conjuntos de instruções usando MIPS, uma vez que o número de instruções de máquina vai ser diferente para cada caso. Segundo, os resultados obtidos com o uso do MIPS variam entre programas no mesmo computador impedindo, portanto, que determinada máquina tenha um único valor MIPS [2].

2.1.2 MFLOPS

É definido, de uma forma geral, como o número de palavras de tamanho total da palavra resultante de operações de multiplicação e de ponto flutuante que podem ser realizadas por segundo. Obviamente, operações de adição e subtração de ponto

flutuante levam menos tempo, sendo a divisão a mais lenta de todas. As UCPs mais antigas levam muitos ciclos de relógio para completar uma operação de ponto flutuante e, mesmo considerando-se freqüências altas de relógio, a taxa de operações de ponto flutuante pode ser baixa. Como no caso do 486DX4 de 100 MHz que tem taxa de 6 MFLOPS, ao compará-lo ao R4400 de 200 MHz, que tem taxa de 35 MFLOPS, percebe-se que, para processadores mais antigos, a velocidade de relógio não é uma indicação clara de taxa de MFLOPS.

Com o advento de novas arquiteturas o problema tornou-se ainda mais complexo, pois UCPs como a R10000 podem realizar duas operações de ponto-flutuante a cada ciclo de relógio, conseguindo com isso uma taxa de 400 MFLOPS a 200 MHz. O R8000 é ainda mais complexo, pois possui duas unidades de execução de ponto flutuante, cada uma capaz de realizar duas operações deste tipo por ciclo de relógio a uma taxa de 360 MFLOPS a 900 MHz, dez vezes mais rápido que o P90 da Intel.

Novamente, a tarefa a ser executada é um aspecto importante. Uma UCP de 64 bits capaz de fazer 400 MFLOPS pode ser algo significativo, mas, se a tarefa a ser realizada precisa apenas de processamento de 32 bits, então boa parte da capacidade da UCP está sendo desperdiçada. Algumas UCPs tais como a R5000 tratam desse problema visando principalmente os mercados consumidores que não precisam de operações de processamento de ponto-flutuante de 64 bits. Projetos futuros, como o MDMX, resolverão o problema do desperdício, mas também tornarão a medição do desempenho de UCPs ainda mais difícil.

Um ponto deve ser analisado: a velocidade de acesso à memória. Uma UCP rápida é um aspecto importante quando se refere ao desempenho de pico etc, mas, na realidade, a obtenção do melhor desempenho possível para uma UCP depende também da taxa em que ela pode acessar dados dos vários tipos de memória (caches L1 e L2 e RAM principal). Em um sistema, uma UCP rápida com pouca

memória ou com tempo de acesso lento, não será capaz de obter desempenho semelhante ao desempenho de pico teórico.

Conclui-se então que é difícil comparar duas arquiteturas tendo como base o desempenho em MFLOP, já que o conjunto de operações de ponto flutuante das duas arquiteturas pode não ser coincidente. Assim, em caso de dúvida sobre qual métrica utilizar, uma boa estratégia é adotar a do tempo de execução [2].

2.2 Considerações na Análise de Desempenho da UCP.

Este capítulo tem por objetivo apresentar as principais estruturas que influenciam diretamente no desempenho da UCP.

Dentre estas estruturas não se pode deixar de citar as principais características de arquitetura que influenciam diretamente no desempenho, que são: cache, pipeline, arquitetura Superescalar, barramento externo e finalmente quantidade de registradores.

2.2.1 Cache

É pouco provável, um microprocessador hoje em dia sem cache de memória, um sistema que utiliza uma pequena quantidade de memória estática como intermediária no acesso à lenta memória RAM, embora o seu funcionamento varie de acordo com o método organizacional empregado pelo controlador de cache, a finalidade é a mesma: aumentar o desempenho do microcomputador, fazendo com que o estado de espera (wait stats) não seja necessário [5].

É inquestionável a importância do cache de memória em microcomputadores modernos, tanto que hoje em dia, tem-se dois tipos de cache de memória: cache L1 (interno, presente dentro do processador) e o cache L2 (também interno, atualmente

presente dentro dos processadores mais novos). Nos testes do último capítulo, as memórias caches foram tratadas como uma “caixa preta”, não considerando diretamente a sua arquitetura interna e seus detalhes de gerenciamento.

2.2.2 Cache L1

Atualmente todos os processadores trabalham com um esquema chamado de multiplicação de relógio. Dessa forma, a freqüência de operação interna só é utilizada na execução de instruções que não dependam da memória RAM, tais como: os cálculos, aritméticos e lógicos, por exemplo. No caso da instrução necessitar buscar um dado na memória, a velocidade do processador cairá para mesma velocidade do barramento local, onde, ainda por cima, terá de utilizar os estados de espera no acesso à memória RAM.

O controlador de cache L1, que está embutido dentro do processador, carrega para a cache de memória interno dados que ela acredita que o processador necessitará durante os próximos pulsos de relógio. Dessa forma, em vez de acessar dados na memória RAM, que é lenta, o processador acessa uma cópia de tais dados no cache de memória L1. Como o processador é capaz de acessar o cache L1 em sua freqüência de operação interna, os dados são lidos quase que instantaneamente [1].

2.2.3 Cache L2

Assim como o cache L1, o cache L2 tenta antecipar os próximos endereços a serem lidos pelo processador, colocando no cache de memória os dados lá contidos. O acesso ao cache de memória é feito sem a utilização dos estados de espera e, com isso, o desempenho do processador é mantido [2].

Na maioria das vezes, o processador busca dados não da memória RAM, mas sim a cópia desses dados contida no cache de memória. Se precisasse ler a memória RAM, diminuiria o desempenho do micro por causa da utilização dos estados de espera ou por causa da latência da memória (tempo demorado para a memória devolver dados solicitados).

2.2.4 Pipeline

O pipeline é uma técnica de implementação em que múltiplas instruções são sobrepostas na execução [5].

Existem três tipos de pipeline. São eles os pipelines de instrução, os aritméticos e os superescalares. Um pipeline de instrução típico possui as fases de busca, decodificação, busca de dados e “write back” (escrita dos resultados). Esses blocos são simples e geralmente são executados em apenas um ciclo de relógio, com exceção do bloco de execução que por ser mais complexo pode demorar vários ciclos de relógio para ser executado. Isso pode causar um desequilíbrio já que as fases possuem tempos de execução diferentes.

Para resolver esse problema, duas técnicas podem ser utilizadas. Uma delas consiste em subdividir a fase de execução em vários estágios e a outra consiste em alongar as fases mais curtas. [5]

Existem também pipelines utilizados para aumentar a velocidade das operações aritméticas. Esses pipelines aritméticos são projetados para executar funções fixas. Eles efetuam separadamente as operações em ponto fixo e ponto flutuante. Esse tipo de pipeline pode ter vários blocos dependendo da aplicação implementada. Todas as operações aritméticas (add, subtract, multiply, division,

squaring, rooting, logarithm, etc.) podem ser implementadas através de um adicionador básico e de operações de deslocamento [9].

Os pipelines estáticos são monofuncionais, pois executam apenas funções fixas, e os dinâmicos são multifuncionais, pois podem realizar mais de uma função. A diferença entre os dois é que o estático executa uma função de cada vez, e diferentes funções podem ser efetuadas em instantes diferentes, já o dinâmico pode realizar várias funções simultaneamente.

Os pipelines superescalares são basicamente um conjunto de pipelines funcionando em paralelo. Consiste em se aumentar o número de pipelines, ao invés de um, têm-se dois ou três pipelines em paralelo. A vantagem desse tipo de pipeline se dá pelo paralelismo real, com dois ou mais instruções sendo processadas em paralelo, com melhora significativa do desempenho já as desvantagens consistem na necessidade do código ser preparado, aumento de complexidade e problemas de dependências e desvios. Nos testes do último capítulo, o pipeline atuou de modo indireto, pois não se quantificou quanto este elemento influencia no desempenho da UCP, porém de uma forma indireta o pipeline reflete muito o desempenho, só não foi medido a porcentagem em que o pipeline atua no desempenho da UCP e nem características técnicas como predição de desvios e BTB dinâmica.

2.2.5 Arquitetura Superescalar

Da mesma forma que o pipeline, a arquitetura superescalar, é uma forma de paralelismo no nível das instruções, pois tem como objetivo aumentar o desempenho de um processador executando mais de uma instrução ao mesmo tempo. Esta arquitetura é composta basicamente de múltiplas unidades funcionais dentro de um único processador. De forma que, comparando a um suposto processador com apenas uma unidade funcional, esta tecnologia, aumenta

teoricamente n vezes a velocidade do processamento, devido à n unidades funcionais adicionadas à arquitetura para trabalhar em paralelo.

2.2.6 Barramento Externo

O barramento é um conjunto de vias onde são conectados vários dispositivos possibilitando a troca de informações entre eles. Logo quanto maior a largura do barramento, melhor será a transmissão de bits e em conseqüência disso, melhor será o desempenho total de um sistema.

2.2.7 Quantidade de Registradores

Os registradores são memórias auxiliares internas a UCP que possui maior velocidade de transferência e menor capacidade de armazenamento dentro do sistema computacional.

A capacidade de armazenamento dos registradores está na habilidade de guardar apenas um único dado, uma única instrução ou até mesmo um único endereço [5].

Dessa forma, a quantidade de bits de cada um é de 8 a 128 bits, dependendo do tipo de processador. Registradores de dados têm, em geral, tamanho definido pelo fabricante. Os registradores são memórias de semicondutores e, portanto para funcionarem precisam de energia elétrica, logo são memórias voláteis, sendo fabricados com tecnologia igual à dos demais circuitos da UCP, visto que eles se encontram inseridos no seu interior. No entanto, há diversos modelos de tecnologia de fabricação de semicondutores, uns com tempo de acesso maior que outros custos e capacidade de armazenamento, no mesmo espaço físico, diferente. Tecnologia bipolar e MOS “metal oxido semiconductor” são comuns na fabricação de registradores.

Pelo fato dos registradores armazenar dados que serão requisitados pela UCP, e possuírem o menor tempo de acesso, a quantidade de registradores influenciam no desempenho do processador, pois uma arquitetura que possui uma vasta quantidade de registradores pode armazenar vários dados e evita que a UCP fique ociosa durante o seu processamento [5].

As informações dadas no decorrer deste capítulo se fazem necessárias para o entendimento da justificativa mencionada no capítulo quatro em relação à análise dos resultados dos testes de desempenho.

3 Benchmarks

Este capítulo define o que vem a ser benchmark, além de apresentar os diferentes tipos e principais benchmarks existentes no mercado para a análise de desempenho da UCP.

Um teste de benchmark é o processo que consiste na execução de um determinado programa ou carga de trabalho em uma máquina ou sistemas específicos, medindo o desempenho resultante. Esta técnica permite uma avaliação do desempenho desta máquina para a carga de trabalho utilizada que, em termos práticos, pode ser um conjunto de programas executados em um computador. Esses benchmarks podem ser aplicações completas, sendo partes mais executadas de um programa que são os kernels, ou programas sintéticos [1].

Os programas benchmarks podem ser classificados em quatro grupos: sintéticos, kernel, algorítimo e aplicação.

Os benchmarks sintéticos são aqueles cujo código não faz nenhuma computação útil, não representa nenhuma aplicação real, somente exercita alguns componentes básicos do computador. Geralmente, tentam determinar uma

freqüência média de instruções típicas, comumente utilizadas, e recriá-las em um programa. Os mais conhecidos são o Whetstone e Dhrystone. Os do tipo Kernel são baseados no fato de que a maior parte da computação de um programa é concentrada em uma pequena parte de seu código. Esta pequena parte, chamada de núcleo (kernel), é extraída do programa e usada como benchmark. Ressalta-se que eles não servem para avaliar completamente o desempenho de uma máquina. São bastante interessantes por sua simplicidade e pequeno tamanho. Um bom exemplo deste tipo de programa é o Livermore Loops. Os benchmarks do tipo algoritmos são aqueles que possuem seus algoritmos bem definidos, geralmente implementações de métodos conhecidos em computação numérica como, por exemplo, os métodos de resolução de equações lineares (álgebra linear) que fazem parte do benchmark Linpack.

Finalmente, os benchmarks do tipo aplicação são programas completos, que resolvem problemas científicos bem definidos, entre eles o SPEC [14].

Um benchmark deve ser escolhido de tal forma a caracterizar uma carga de trabalho particular da qual se deseja medir o desempenho. Hoje em dia, está claro que o melhor benchmark é o emprego de programas reais que serão usados no dia a dia da máquina. Porém, aplicações reais não são usadas no caso de projetos de novos processadores. O uso de benchmarks sintético tem algumas vantagens já que são programas pequenos e mais fáceis de simular do que programas reais. Além disso, a geração de código executável de programas reais pode ainda nem estar disponível nos compiladores.

Mas para comparar o desempenho de um sistema computacional já no mercado, a melhor estratégia é a utilização de programas reais. Mais uma vez vale lembrar que o desempenho não é uma medida absoluta. Para dois programas diferentes P1 e P2, um sistema A pode ter um melhor desempenho para o primeiro programa (P1) e o sistema B, para o segundo programa (P2).

Na seção seguinte, serão conceituados os diferentes tipos de benchmarks, suas características e modo de funcionamento.

3.1 Benchmark Whetstone

É um programa escrito em linguagem C que permite testar o desempenho do processador em operações de ponto-flutuante. A versão original foi desenvolvida em 1976. O Whetstone faz parte de vários programas de benchmark atuais, cujo resultado indica o número de vezes por segundo em que o processador é capaz de executar o programa. O desempenho do processador neste teste é um bom indicativo do seu desempenho em jogos 3D e em aplicativos científicos, apesar de não ser necessariamente uma medida confiável para aplicativos reais [4].

O Whetstone foi o primeiro benchmark sintético mencionado na literatura com fins específicos de medida de desempenho. Sua primeira versão foi publicada na linguagem ALGOL 60 (apesar de ter sido mais utilizado em FORTRAN), em 1976, por H. J. Curnow e B. A. Wichmann, do Laboratório Nacional de Física na Inglaterra [2].

O Whetstone é um programa com poucas linhas de código (cerca de 500 linhas), composto de vários módulos. Cada módulo tem um tipo diferente, explora diferentes características da linguagem de programação e é executado várias vezes através de laços do tipo "FOR". [13]

Na verdade, existia uma linguagem chamada ALGOL Whetstone, que foi utilizado para coletar estatísticas sobre a distribuição das instruções Whetstone de um grande número de programas numéricos.

O próprio código do benchmark foi traduzido para esta linguagem intermediária (instruções Whetstone), de onde eram calculadas as freqüências das

diferentes instruções e comparadas com o código original. Alguns ajustes eram feitos através da atribuição de pesos (alterando-se os tamanhos dos laços), até que a distribuição das instruções do benchmark refletisse a freqüência média das instruções dos códigos analisados anteriormente.

Os resultados eram, então, medidos em termos de MWIPS (Mega Whetstone Instructions per Second - Milhões de Instruções Whetstone por Segundo) [2].

As vantagens deste benchmark são o seu tamanho reduzido e simplicidade do código, além de explorar bastante as operações em ponto-flutuante. Portanto, serve como comparativo para pequenas aplicações científicas em computadores de pequeno e médio porte.

Algumas vezes, para fins comerciais, o programa sofre pequenas alterações como retirada dos comandos de impressão, o que pode eliminar partes importantes do código. Para tentar resolver este tipo de problema, em 1988, uma nova versão em Pascal foi publicada [1].

3.2 Benchmark Dhrystone

O benchmark Dhrystone foi publicado pela primeira vez na linguagem ADA, em 1984. Agora a versão em linguagem C do Dhrystone onde sua aplicação principal é na análise da eficiência de combinações hardware/compilador em máquinas de pequeno e médio porte. No entanto o benchmark Dhrystone original ainda é utilizado para medir desempenho de processadores.

Originalmente, com o Dhrystone, pretendia-se criar um programa pequeno de benchmark que fosse representativo na programação de sistemas (no caso de operações aritméticas). O código do Dhrystone é dominado por aritmética simples, operações com string, decisões lógicas, e acessos de memória com intenção de

refletir as atividades da UCP nas aplicações de computação de propósito mais geral. O resultado do teste do Dhrystone é determinado através do cálculo do tempo médio que um processador leva para executar as muitas interações de um simples laço que, por sua vez, contém uma seqüência fixada de instruções que compõem o benchmark. Quando se menciona o Dhrystone, é usualmente caracterizado como “DMIPS” (Dhrystone Millions of Instruction per Second) [13].

O Dhrystone tem um número de atributos que o levaram a ser amplamente usado no passado para medir desempenho de processadores. Em primeiro lugar, o Dhrystone é compacto, tem alta disponibilidade no domínio público e é simples de usar.

O Dhrystone compara o desempenho do processador em análise ao de uma máquina de referência. Esta é a vantagem que se tem sobre a utilização direta do número de MIPS, já que usar uma máquina de referência efetivamente compensa as diferenças na complexidade das instruções, etc. Assim, comparar literalmente os números de MIPS de uma arquitetura RISC com os de uma CISC não é válido, como dito anteriormente neste trabalho [2].

No passado, a indústria adotou o VAX 11/780 como uma máquina de referência de 1 MIPS. O VAX 11/780 alcança 1757 D/S (Dhrystone por segundo). O resultado do benchmark Dhrystone é calculado medindo o número de D/S para o sistema, e dividindo este número por 1757 (da máquina de referência). Então, 80 MIPS “significam” 80 MIPS VAX Dhrystone”, o que, por sua vez, significa dizer que esta máquina é 80 vezes mais rápida que a máquina de referência VAX 11/780. Uma taxa de DMIPS/MHz leva esta normalização ainda mais adiante, permitindo uma comparação de desempenho de processador para taxas diferentes de relógio [2].

Entretanto, algumas das vantagens aparentes do Dhrystone também são fraquezas significativas deste benchmark. Os números do Dhrystone refletem na verdade o desempenho do compilador da linguagem C e suas bibliotecas,

provavelmente mais do que o desempenho do próprio processador. Além disso, seu projeto foi baseado na análise de vários outros programas, escritos em diferentes linguagens e por diferentes autores, porém, voltados à programação de sistemas (sistemas operacionais, compiladores, etc.). Esta é uma característica bastante relevante, pois, diferentes classes de aplicações enfatizam diferentes tipos de operações, como por exemplo, as aplicações numéricas que utilizam intensamente vetores e aritmética de ponto-flutuante; aplicações comerciais utilizam predominantemente atividades de entrada / saída; e programação de sistemas utiliza bastante ponteiros, sentenças "IF ", chamadas de procedimentos, além de conter menos laços e expressões numéricas mais simples [14].

3.3 Livermore Loops

O nome mais correto deste benchmark é Livermore Fortran Kernels, porém, ele é mais conhecido como "Livermore Loops". Ele se classifica como um benchmark do tipo kernel. O Livermore Loops é uma síntese dos núcleos de vários programas, cujo responsável foi Frank H. McMahon, do Laboratório Nacional de Lawrence Livermore, em 1970.

O benchmark é composto de vinte e quatro núcleos de programas, extraídos de códigos vindos de diferentes áreas científicas e escritos por diferentes autores. Este núcleo contém sentenças comumente utilizadas em FORTRAN. Os laços contidos neste benchmark utilizam aplicações com vetores, que são executados em três dimensões diferentes.

A ênfase deste benchmark está nas operações em ponto-flutuante, além da possibilidade de verificar a habilidade do compilador em gerar códigos eficientes. Este benchmark originou a unidade de medida MFLOP/s [13].

3.4 Linpack

O Linpack é um dos mais famosos benchmarks, utilizado inclusive nos testes das 500 máquinas mais rápidas existentes (Top500), por ser o que tem maior número de resultados reportados.

Originalmente, o Linpack era um pacote de sub-rotinas que tinham por finalidade resolver sistemas de equações lineares algébricas. Jack Dongarra, da Universidade do Tennessee (antigo Laboratório Nacional Argonne), publicou-o em 1976, sem intenção de torná-lo um benchmark, incluindo apenas como um apêndice no Guia do Usuário do Linpack, dados referentes ao desempenho de vinte e três computadores [14].

O Linpack encaixa-se na classificação de benchmark tipo algoritmo. Ele contém dois conjuntos de rotinas: um para decomposição de matrizes e outro para resolver o sistema de equações lineares baseados em decomposição.

Dentre as várias sub-rotinas contidas no pacote, a mais utilizada e a que consome a maior parte do tempo de execução do programa é a "saxpy". Ela contém somente 15 linhas de código, em linguagem de baixo nível e trabalha com vetores de apenas uma dimensão. As rotinas de mais alto nível chamam-na várias vezes e operam com vetores bi-dimensionais [13].

Este benchmark baseia-se em um subpacote de rotinas para operações básicas de álgebra linear, o BLAS (Basic Linear Algebra Subroutine - Subrotinas de Álgebra Linear Básica). A versão FORTRAN é chamada FORTRAN BLAS e a versão Assembly, Coded BLAS, mas já não é mais utilizada.

Existem versões diferentes para o Linpack cada versão diferencia-se no tamanho das matrizes (a mais utilizada é 100x100); na precisão, que pode ser dupla ou simples; e em relação aos tipos de laços (rolled/unrolled). Os laços unrolled são

incrementados de 4 em 4, e o corpo do laço contém sentenças para índices i, i+1, i+2 e i+3. Em algumas máquinas/compiladores o código executa mais rápido desta maneira. Nas máquinas vetoriais mais modernas, o compilador já faz isto de maneira mais otimizada e, portanto, elas executam a versão "rolled" (laços incrementados de 1 em 1) mais rapidamente.

Os resultados dos testes são reportados em MFLOPS (Millions of Floating Point Operations per Second), GFLOPS (Billions of Floating Point Operations per Second) ou até TFLOPS (Trillions of Floating Point Operations per Second) [2].

Sua aplicação é visível em máquinas que utilizam softwares para cálculos científicos e de engenharia, visto que as operações mais utilizadas nestes tipos de aplicações são em ponto-flutuante.

3.5 Benchmark SPEC

O mais conhecido entre os benchmarks para processadores é o SPEC (System Performance Evaluation Cooperative). Essas cooperativas americanas, criadas em 1989 por diversas empresas do ramo de fabricação de computadores, objetiva melhorar as métricas e as informações disponíveis sobre desempenho de processadores [2].

Apesar de serem muito úteis para as máquinas dessa geração de computadores, os esforços dos cooperados da SPEC não serão suficientes para atender às exigências da próxima geração de máquinas [6]. Em 1991, uma medida do throughput foi adicionada ao benchmark SPEC. Embora este programa seja mais apropriado para a avaliação da utilização de tempo compartilhado entre os diversos usuários do sistema, ligados a sistemas com um ou vários processadores. Foram adicionados outros benchmark, entre eles, principalmente, os que fazem uso do sistema operacional e de atividades de entrada e saída. Outra mudança realizada foi

à exclusão do programa matrix300, além da inserção de mais benchmarks no padrão SPEC.

A versão inicial do SPEC, chamada SPEC89, era composta por seis programas em ponto flutuante e apenas quatro programas de inteiros, geralmente voltados à computação de números em ponto-flutuante.

O ano de 1992 marcou a incorporação de um novo benchmark, denominado SPEC92, que apresentava novos programas, sendo que o matrix300 fora eliminado, e ainda calculava dois índices separados, o SPECint e SPECpf, respectivamente, para programas em ponto fixo e ponto flutuante. Alem disso, o SPEC92 incluiu uma medida denominada SPECbase que não permitia a utilização de compiladores com opções especificas. O SPECbase é a medida de desempenho que mais se aproxima da vivência em aplicações reais [2].

Em 1995, houve mais uma atualização do SPEC que apresentava novos programas, tanto inteiros quantos em ponto-flutuantes. Também foram eliminados outros tantos, ou por defeito do programa ou por não mais satisfazerem aos requisitos do desempenho desde a primeira versão [6].

A partir daí continuou a inclusão de novos conjuntos de benchmarks, além dos anteriores cujo alvo era o desempenho do processador. O benchmark SDM (System Development Multitasking) compreende dois programas que são versões sintéticas de carga de trabalho, típicos de ambiente de desenvolvimento, com edições, compilações, comandos de sistemas. O benchmark SFS (System-level File Server) é uma carga de trabalho sintética para teste de desempenho de um servidor de arquivos. Ambos os benchmark incluem componentes de entrada/saída e sistema operacional em vez de testar processadores.

A mais recente atualização do SPEC recebeu o nome de UCP2000 da SPEC que representa o conjunto de programas de benchmark mais cotado da SPEC. É

um benchmark padronizado para UCP, projetado para fornecer uma medida comparativa de desempenho nas áreas mais amplas e práticas do que aquelas que a tecnologia de hardware de hoje abrange.

Os benchmarks UCP2000 da SPEC foram desenvolvidos para aplicações reais de usuários. Esses benchmarks medem o desempenho do processador, da memória e do compilador do sistema testado.

O UCP2000 é dividido em dois conjuntos principais de benchmarks: o SPECint e o SPECfp. O SPECint mede o desempenho das operações aritméticas com inteiros do sistema testado. Processamento de palavras, compressão de arquivos, e-mail e desempenho de database compreendem esta categoria. O SPECfp mede o desempenho das operações de ponto-flutuante do sistema testado; jogos 3D e processamento de áudio são exemplos de aplicações de pontos-flutuantes [8].

Um aspecto importante da metodologia utilizada pela SPEC para desenvolver o UCP2000 é a percepção de que, com as complexas aplicações que se tem hoje, é quase impossível e, mais importante ainda, fútil separar o desempenho da UCP do sistema de memória. Por exemplo, as aplicações do SPECfp e as cargas de trabalho associadas a essas aplicações tendem a requerer grandes quantidades de memória em adição a um bom desempenho computacional com pontos-flutuantes. Isto não é um acidente, nem se deve a peculiaridades associadas às cargas de trabalho do SPECfp.

As maiorias das aplicações que utilizam processamentos de ponto-flutuante necessitam, por natureza, tanto um de bom processamento de pontos-flutuantes quanto de grandes quantidades de memória. É este equilíbrio delicado no comportamento das aplicações que os benchmarks UCP2000 procuram expor [2].

4 Athlon x Pentium

O objetivo deste capítulo é reportar o resultado de uma aplicação prática. O teste consistiu em uma comparação de desempenho feita entre dois computadores da linha IBM-PC, utilizando processadores da Intel e AMD, fabricados por indústrias concorrentes e amplamente utilizados no mercado atual de computadores pessoais. Os principais comparativos recaíram para dois concorrentes diretos que são: o Athlon XP 1800+, codinome Palomino e o Pentium 4 1800 Mhz, codinome Willamette. Outros resultados também foram reportados para que se pudesse fazer uma comparação mais precisa em relação à velocidade de processamento da UCP. Os dois processadores foram tratados como uma “caixa-preta”, ou seja, os testes foram realizados encapsulando as funções de processamento e simplesmente constatando a arquitetura que possui o menor tempo de execução perante os vários módulos desenvolvidos em um programa benchmark para este fim.

No experimento foram utilizados dois programas *benchmarks*: O SisSoftware Sandra 2004 que implementa o whetstone e o dhrystone (ver anexo A) e um pequeno software, o WinBenchGLC, desenvolvido especialmente para testar as habilidades da CPU no que se refere à aritmética de inteiros e ponto flutuante, além de outros módulos que realizam rotinas comumente usadas em diversas áreas de

aplicação científica. No Sandra 2004 o whetstone mede o desempenho do processador no que se refere a operações de pontos flutuantes, enquanto o dhrystone mede o desempenho no que se refere a operações com inteiros e manipulações de *strings*.

Ao se comparar duas arquiteturas diferentes é importante ressaltar que com a tecnologia presente a partir do Athlon XP, a AMD voltou a utilizar a nomenclatura PR (Performance Reference), que indica a velocidade do processador não através do seu clock real e sim através do desempenho do seu processador comparado a um processador equivalente da Intel [10]. Isso quer dizer que o Athlon XP 1800+ que trabalha a 1500Mhz ou 1533Mhz é o processador com desempenho equivalente a de um Pentium 4 de 1.8Ghz. A tabela 1, abaixo, faz um comparativo entre os processadores da AMD e Intel segundo o índice referência de performance (PR).

Processador	Velocidade atual (MHz)	Índice de Performance (PR)	Competidor
Athlon XP 3600+	2800=21x133	3600+	Pentium 4 3.6GHz
Athlon XP 3400+	2667=20x133	3400+	Pentium 4 3.4GHz
Athlon XP 3200+	2533=19x133	3200+	Pentium 4 3.2GHz
Athlon XP 3000+	2400=18x133	3000+	Pentium 4 3GHz
Athlon XP 2800+	2250=13.5x167	2800+	Pentium 4 2.8GHz
Athlon XP 2700+	2167=13x167	2700+	Pentium 4 2.8GHz
Athlon XP 2600+	2133=16x133	2600+	Pentium 4 2.6GHz
Athlon XP 2400+	2000=15x133	2400+	Pentium 4 2.4GHz
Athlon XP 2200+	1800=13.5x133	2200+	Pentium 4 2.2GHz
Athlon XP 2100+	1733=13x133	2100+	Pentium 4 2.2GHz
Athlon XP 2000+	1667=12.5x133	2000+	Pentium 4 2GHz
Athlon XP 1900+	1600=12x133	1900+	Pentium 4 1.9GHz
Athlon XP 1800+	1533=11.5x133 ou 1500=15x100	1800+	Pentium 4 1.8GHz
Athlon XP 1700+	1467=11x133	1700+	Pentium 4 1.7GHz
Athlon XP 1600+	1400=10.5x133	1600+	Pentium 4 1.6GHz
Athlon XP 1500+	1333=10x133 ou 1300=13x100	1500+	Pentium 4 1.5GHz

Processador	Velocidade atual (MHz)	Índice de Performance (PR)	Competidor
Athlon XP 1400+	1200=12x100	1400+	Pentium 4 1.4GHz ou Pentium III 1.4GHz
Athlon XP 1300+	1100=11x100	1300+	Pentium 4 1.3GHz

Tabela 1: Índices de Performance Reference (PR) da AMD para comparação com processadores Intel equivalentes [10].

4.1 Programa Desenvolvido: WinBenchGLC

Foi criado um pequeno software benchmark para poder medir o tempo de execução de funções que utilizam aritmética de inteiro e ponto flutuante. O programa é composto por nove módulos, tendo cada um uma função específica. Todos os tipos de variáveis foram utilizadas, desde o inteiro simples até o inteiro de 64 bits e as variáveis do tipo ponto flutuante: real, dupla, estendida e comp. Além da utilização de matrizes, funções trigonométricas e aritméticas. A métrica utilizada foi o tempo de execução medido em minutos:segundos:milésimos de segundo e depois convertido para milésimos de segundo, para uma maior precisão. O relatório gerado inclui a medida parcial das operações de inteiros e ponto flutuante. O tempo de cada função é reportado separado e o tempo total serve para a comparação dos resultados obtidos entre os diferentes processadores. O programa pode ser considerado um benchmark do tipo híbrido, sendo os dois primeiros módulos do tipo sintético, ou seja, não executam computação útil, só exercitam as habilidades da UCP com cálculos. Os módulos restantes podem ser considerados do tipo algoritmo, pois executam computação útil e testam as habilidades do processador com cálculos aritméticos de inteiro e ponto flutuante comumente usados na computação científica.

O desempenho global não é avaliado, pois a influência dos demais hardwares e periféricos não é considerada diretamente nos testes, uma vez que o foco das

operações está no processador (UCP). Assim a placa de vídeo e o disco rígido, por exemplo, tem uma influência mínima no processo de testes. Porém o sistema global, considerando a placa mãe, as memórias principais, a velocidade do barramento externo e outros fatores, afeta consideravelmente o resultado.

As características técnicas dos computadores em questão, como o nome, a família, o clock, a velocidade do barramento externo, foram reportados pelo CPUz (ver anexo B) em um arquivo html. A linguagem utilizada para produzir o WinBenchGLC foi o Borland Delphi versão 7 Enterprise e não foram feitas otimizações de compilação, isso para que os resultados fossem o mais fiel possível em ambas as plataformas: Intel e AMD. O sistema operacional utilizado para compilação com o Delphi foi o Microsoft Windows XP Workstation (Build 2600). O programa é compatível com IBM-PC e roda em sistemas operacionais Windows 98/ME/NT/2000/XP. Os testes foram feitos em diversas máquinas, desde um Pentium 133Mhz até um Athlon XP 2800+ operando a 2083Mhz.

Antes da realização do teste nas máquinas, foram observados determinados fatores que influenciam no tempo de execução. Os principais cuidados percebidos para que os resultados fossem o mais confiável possível foram:

- O sistema operacional estava recém iniciado;
- Foram desabilitados programas rodando em segundo plano como anti-vírus e similares;
- Somente a aplicação desenvolvida estava sendo executada;
- O processador ficou ocupado em cerca de 98% de seu tempo de processamento com o programa, durante sua execução;
- Os testes são repetidos três vezes e o menor tempo entre as três execuções é considerado;
- O WinBenchGLC tem uma prioridade alta de execução no sistema operacional e executa uma função por vez, sem concorrência entre os processos internos;

4.2 Módulo Aritmética de Inteiros

Neste módulo foram construídas quatro funções que exploram as operações básicas: soma, subtração, multiplicação e divisão de inteiros. Uma constante de controle foi usada para calibrar os laços dos testes de inteiros, essa constante foi iniciada na versão final valendo 1800000. Em cada iteração um vetor de 100 posições de inteiros é percorrido e preenchido com um número inteiro aleatório, logo após, na próxima instrução, uma variável soma do tipo inteiro 64 vai acumulando o valor da posição corrente do vetor. Este procedimento é aplicado para as quatro funções, separadamente, cada uma utilizando um operador diferente (+,-,*). Para divisão de inteiros foi utilizado o operador div. O tempo então foi medido, para cada função, do instante em que se entra no laço principal até executar todas as iterações controladas pela constante de controle. São reportados então quatro tempos distintos um para cada operador, mais o tempo parcial do módulo aritmética de inteiros.

4.3 Módulo Aritmética de Ponto Flutuante

Neste módulo as operações básicas foram usadas em quatro funções explorando os operadores de soma, subtração, multiplicação e divisão de ponto flutuante. Uma constante foi declarada para controlar o tamanho dos laços dos testes de ponto flutuante e foi escolhido o mesmo valor para a constante de controle de inteiros, 1800000. Essa escolha padroniza os testes, uma vez que os dois módulos, de inteiros e de ponto flutuante, trabalham com o mesmo número de iterações. Quanto maior o número dessa constante maior será o tempo de execução. Em cada iteração um vetor de 100 posições em ponto flutuante estendido (3.6 x 10^-4951...1.1 x 10^4932) é percorrido e preenchido com um número fracionário aleatório, seguindo pela próxima instrução em que uma variável soma do

tipo comp (-2^63+1 .. 2^63 -1) vai acumulando o valor da posição corrente do vetor, utilizando um dos quatro operadores básicos (+, -, /, *) até chegar na última iteração, que é a de 1800000. O processo se aplica para as quatro funções. O tempo então foi medido, para cada função, do instante em que se entra no laço principal até executar todas as iterações controladas pela constante de controle. São reportados então quatros tempos distintos, um para cada operador, mais o tempo parcial do módulo aritmética de ponto flutuante.

4.4 Módulos Adicionais

Foram implementados mais sete módulos que exploram as habilidades do processador utilizando algoritmos necessários nas mais diversas áreas da computação:

4.4.1 Imagem (Mediana 7x7)

Utiliza aritmética de inteiros e é um algoritmo clássico para manipulação de imagens. Ele percorre cada pixel de uma imagem 640x480 de 24bit de cor. O tempo é medido no instante da primeira iteração até o último pixel da imagem.

Introduzido por Tukey na década de 70, este filtro calcula o valor mediano dos elementos da máscara ordenados cujo resultado será aplicado sobre o pixel da imagem original. Possui uma tendência de “consolidar regiões de mesma intensidade”, mas apresenta um bom resultado visual suavizando discretamente a imagem. [12].

4.4.2 Números Primos

O algoritmo procura por números primos entre um intervalo numérico, contando quantos números primos existem neste intervalo. O intervalo escolhido foi

entre 1 e 13000000. Esta função utiliza exclusivamente cálculos com aritmética de inteiros. [15]

4.4.3 Números Complexos

Utiliza uma classe de números complexos. Este conjunto numérico possui duas partes do tipo ponto flutuante double (5.0 x 10^-324...1.7 x 10^308). Uma parte real e outra imaginária. Os números complexos são muito utilizados em aplicações científicas, como a construção de fractais, construção de circuitos monofásicos, circuito em fase tipo R-C, circuitos em série tipo R - L – C [18]. Foram declaradas três variáveis double. Essas variáveis são enviadas como parâmetros para funções cos, seno, arcocotangente e outras, especialmente escritas para manipular números complexos. Um laço com 220000 iterações executa as funções que são utilizadas para explorar ao máximo os cálculos com ponto flutuante.

4.4.4 Eliminação de Gauss

Este módulo implementa a resolução de equações lineares utilizando o método de Gauss, que é um método numérico para resolução de sistemas de equações lineares que se baseia na manipulação das equações do sistema linear original de modo a transformá-lo em um sistema equivalente cuja matriz dos coeficientes é triangular superior ou inferior. [16]

Este processo é de grande importância, pois resolve de maneira simples qualquer sistema de equação linear plausível de resolução, e sistemas de equações lineares estão presentes nos mais diversos seguimentos, tais como: na engenharia elétrica para resolução de circuitos elétricos, na engenharia mecânica em sistemas de vigas e em qualquer outro lugar onde um sistema de equações lineares possa ser caracterizado. Foi utilizada uma matriz 100x100 de valores reais (5.0 x 10^-324...1.7 x 10^308) e executado o método de resolução de Gauss que é composto

de quatro funções básicas. São resolvidos 530 sistemas lineares diferentes. Este teste utiliza a aritmética de ponto flutuante.

4.4.5 Inversão de Matriz

A inversão de matriz é muito utilizada em cálculos de engenharia civil e cálculos científicos. A matriz inversa de uma matriz X , é a matriz X^{-1} , tal que $X * X^{-1} = X^{-1} * X = I_n$, onde I_n é a matriz identidade de ordem n. [17]

Este módulo inverte uma matriz 100x100 de ponto flutuante estendido, 1500 vezes. Utiliza exclusivamente aritmética de ponto flutuante.

4.4.6 Fractal de Mandelbrot

Os fractais são objetos matemáticos que possuem uma estrutura extremamente detalhada, não importa o quão de perto você os olhe, ou quanto os amplie. Para o teste foi utilizado o conjunto de Mandelbrot, que representa um ninho extremamente detalhado de baías dentro de baías. A imagem é desenhada, utilizando milhões de cálculos em ponto flutuante estendido, plotando um pixel a cada iteração. Esta figura representa um conjunto que James Gleick chamou de “o mais complexo objeto da matemática” [19]. Utiliza aritmética de ponto flutuante.

4.5 Resultados

O tempo total dos nove módulos é reportado em um arquivo texto chamado relatório.txt. O tempo total reportado pode sofrer uma variação de mais ou menos 8%; este é o valor da margem de erro considerando diversas situações como a influência do sistema operacional, programas rodando em segundo plano, etc. Um outro arquivo denominado CpuInfo.html é criado pelo programa CPUZ para relatar as características técnicas do computador em questão.

A influência do sistema operacional não foi determinante nos testes e o programa teve uma variação de no máximo dois segundos entre os diversos sistemas operacionais testados.

4.5.1 Configuração dos computadores utilizados no teste

A tabela 2 mostra a configuração dos dois principais computadores avaliados nos testes. É importante notar na tabela 2 que ambas as máquinas possuem configurações equivalentes, placa mãe de alto desempenho e estavam executando o mesmo sistema operacional.

Componente/Periféricos	Athlon	Pentium 4
Nº de CPU's	1	1
Codinome	Palomino	Willamette
Especificação	AMD Athlon(tm) XP 1800+	Intel(R) Pentium(R) 4 CPU 1800MHz
Conjunto de instruções suportadas	MMX, Extended MMX, 3DNow!, Extended 3DNow!, SSE	MMX, SSE, SSE2
Velocidade de clock	1500 MHz	1800 MHz
Multiplicador de clock	15	18
Placa Mãe	Asus A7N266	Intel D850GB
Cache Primária L1	128 Kb	8Kb
Cache Secundária L2	256 Kb	256Kb
Velocidade da cache secundária L2	Total – 1500 Mhz	Total – 1800 Mhz
Memória	256 MB DDR 266 Mhz	256MB RDRam 400 Mhz
Sistema Operacional	Microsoft Windows XP Workstation (Build 2600)	Microsoft Windows XP Workstation (Build 2600)

Tabela 2: Configuração dos dois principais computadores utilizados no teste.

4.5.2 Resultados reportados pelo programa WinBenchGLC

Após a execução do programa WinBenchGLC nos computadores de teste, foram obtidos os seguintes resultados que estão reportados na tabela 3:

Tipo de Teste	Athlon	Pentium
	Tempo (m:s:ms)	
Inteiros		
Soma	00:02:244	00:05:063
Subtração	00:03:024	00:05:468
Multiplicação	00:05:017	00:08:594
Divisão	00:09:114	00:08:641
Parcial Inteiros	**00:19:399**	**00:27:766**
Ponto Flutuante		
Soma	00:06:038	00:07:312
Subtração	00:06:480	00:07:594
Multiplicação	00:07:050	00:08:609
Divisão	00:09:584	00:09:813
Parcial Ponto Flutuante	**00:29:152**	**00:33:328**
Imagem (média 7x7)	00:02:243	00:03:250
Números Primos	00:05:809	00:05:578
Números Complexos	00:06:289	00:09:578
Eliminação de Gauss	00:05:528	00:05:922
Inversão de Matriz PF	00:06:479	00:07:344
Fractal	00:01:812	00:02:015
Tempo Total	**01:16:711**	**01:34:781**

Tabela 3: Tempo de execução dos nove módulos do programa

Os testes de desempenho foram executados em microcomputadores de usuários com o intuito de equiparar as configurações destes microcomputadores. O utilitário Gerenciador de Tarefas do Windows XP foi usado para confirmar o tempo de processamento gasto na execução do programa BenchMark, que foi de 98%. Essa é a porcentagem em que o processador fica ocupado enquanto o programa é executado.

De acordo com a tabela 3, o processador Athlon XP 1800+ se mostrou mais rápido que seu concorrente Pentium 4 1800Mhz em aproximadamente 19%, isso considerando cálculos que envolvem aritmética de inteiros e ponto flutuante.

É importante observar também que o módulo de aritmética de inteiros foi aproximadamente 30% mais rápido no Athlon, podendo ser considerado uma diferença significante, pois a maioria de programas comerciais como Editores de Texto, Navegadores de Internet e outros utilizam quase que exclusivamente cálculos com inteiros. A unidade de Ponto Flutuante (FPU) se mostrou com um desempenho equilibrado, sendo o Athlon mais rápido em cerca de 13%, isso pode ser observado no gráfico 1:

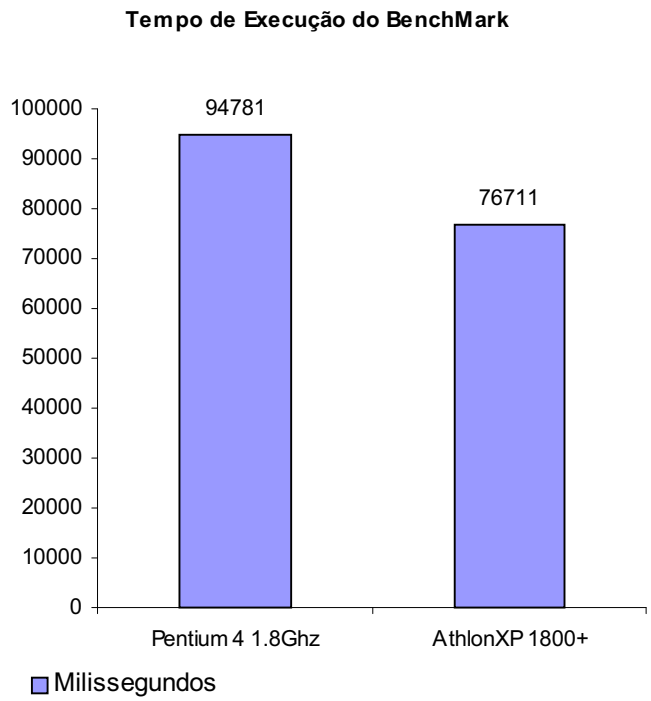

Gráfico 1: Tempo de execução dos computadores testados: Athlon e Pentium

Na tabela 4, foi reportado o tempo de vários computadores comparando vários processadores e o tempo de execução total dos módulos do programa WinBenchGLC.

Processador	Tempo (m:s:ms)	**Tempo (ms)**
AMD		
AMD Duron 800 Mhz	02:44:450	**164450**
AMD Athlon 1000 Mhz	02:14:020	**134020**
Athlon XP 1700+ 1467 Mhz	01:17:221	**77221**
Athlon XP 1800+ 1500 Mhz	01:16:711	**76711**
Athlon XP 2000+ 1667 Mhz	01:07:172	**67172**
Athlon XP 2100+ 1725 Mhz	01:06:516	**66516**
Athlon XP 2800+ 2083 Mhz	00:54:203	**54203**
Intel		
Pentium 133	22:56:100	**1376100**
Pentium 233MMX	11:09:170	**669170**
Pentium II 400	04:28:286	**268286**
Pentium III 550	03:22:241	**202241**
Pentium III 750	02:17:360	**137360**
Pentium 4 1600 Mhz	01:44:881	**104881**
Pentium 4 1800 Mhz	01:34:781	**94781**

Tabela 4: Resultados reportados pelo programa BenchMark para diversos processadores

A tabela 4 pode ser usada como referência para testes realizados na máquina do usuário sendo que foram tomados os mesmos cuidados em relação ao teste das máquinas principais, por isso os resultados apresentam uma precisão bem próxima, em tempo de execução, comparando sistemas similares.

O menor tempo foi alcançado por um Athlon XP 2800+, rodando a 2083Mhz com barramento externo de 333Mhz (2x166), cache L2 de 512Kb operando na velocidade do relógio e 512MB DDR-SDRAM de 333Mhz.

O gráfico 2 traz os resultados dos diversos testes realizados pelo programa WinBenchGLC, medidos em milésimos de segundos.

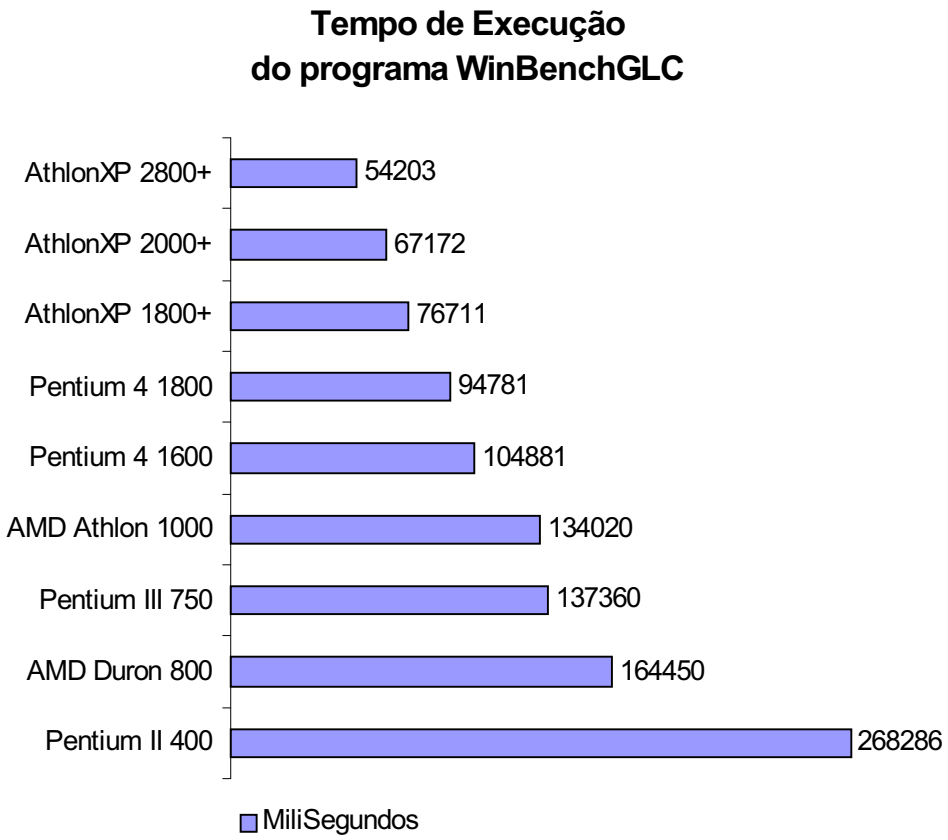

Gráfico 2: Diversos computadores testados pelo programa BenchMark

O desempenho dos processadores atuais podem ser medidos e depois comparados com os gráficos 2 e 3 e com a tabela 4. Para sistemas similares (mesmo processador) o resultado dos testes tem que se aproximar destes índices,

caso contrário, pode estar acontecendo algum problema de configuração ou algum gargalo no sistema.

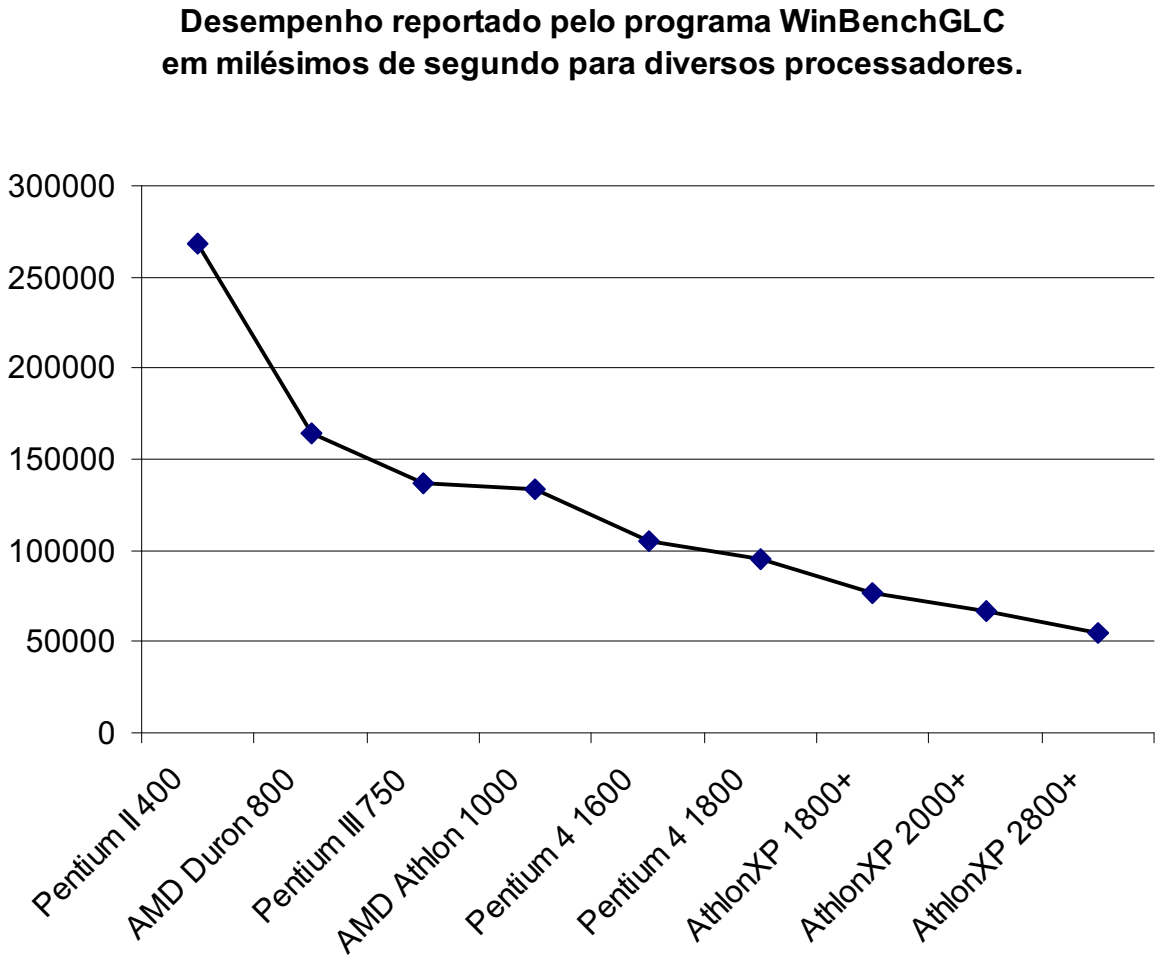

Gráfico 3 : Desempenho de diversos processadores testados pelo WinBenchGLC.

O gráfico 3, no estilo linhas, mostra o desempenho de diversos computadores testados pelo programa WinBenchGLC. O gráfico tem sua base em milisegundos.

De acordo com o gráfico, podemos observar que máquinas cujo valor dos testes forem abaixo de 50000 milisegundos são bastante rápidas.

4.5.3 Resultados reportados pelo Sissoftware Sandra 2004

Os resultados desta sessão foram reportados testando as duas máquinas em condições equivalentes, seguindo os mesmos cuidados com os testes anteriores. O gráfico mostra também o algoritmo Whetstone iSSE2 (ver anexo A) que é uma versão otimizada para operar com as novas instruções SSE2 presentes no processador Pentium 4. O gráfico 3 mostra o resultado desta comparação.

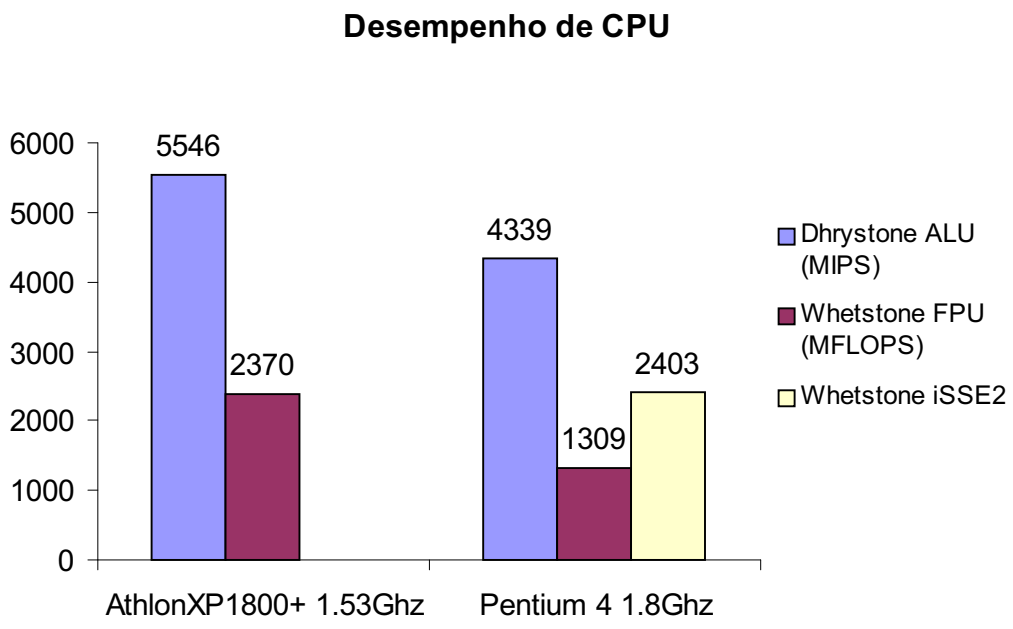

Gráfico 4: Comparando os dois processadores com os dois algoritmos clássicos: Dhrystone e Whetstone, adaptados para as atuais plataformas e sistemas operacionais.

O Athlon XP 1800+ foi aproximadamente 27% mais rápido que o Pentium 4 1.8Ghz nos testes do Sandra. Se considerarmos os testes com a tecnologia SSE2 presente no Pentium 4 essa vantagem da AMD cai para aproximadamente 14%.

Considerando cada algoritmo separado, o Athlon foi 21% mais rápido que o Pentium em cálculos que envolvem aritmética de inteiros e manipulação de strings, executados pelo algoritmo Dhrystone. Já o Whetstone trabalha com uma grande carga de números operando em ponto flutuante. O Athlon ganhou do Pentium, neste teste, com uma vantagem de aproximadamente 44%. A versão otimizada do

algoritmo Whetstone iSSE2 se mostrou equilibrada em relação ao Whetstone executado no Athlon, com uma pequena vantagem para o Pentium.

O gráfico 5 mostra alguns resultados provenientes da base de dados do SisSoftware Sandra 2004, exceto o do Athlon XP 1800+ e do Pentium 4 1.8Ghz que foram testados em máquinas de usuários.

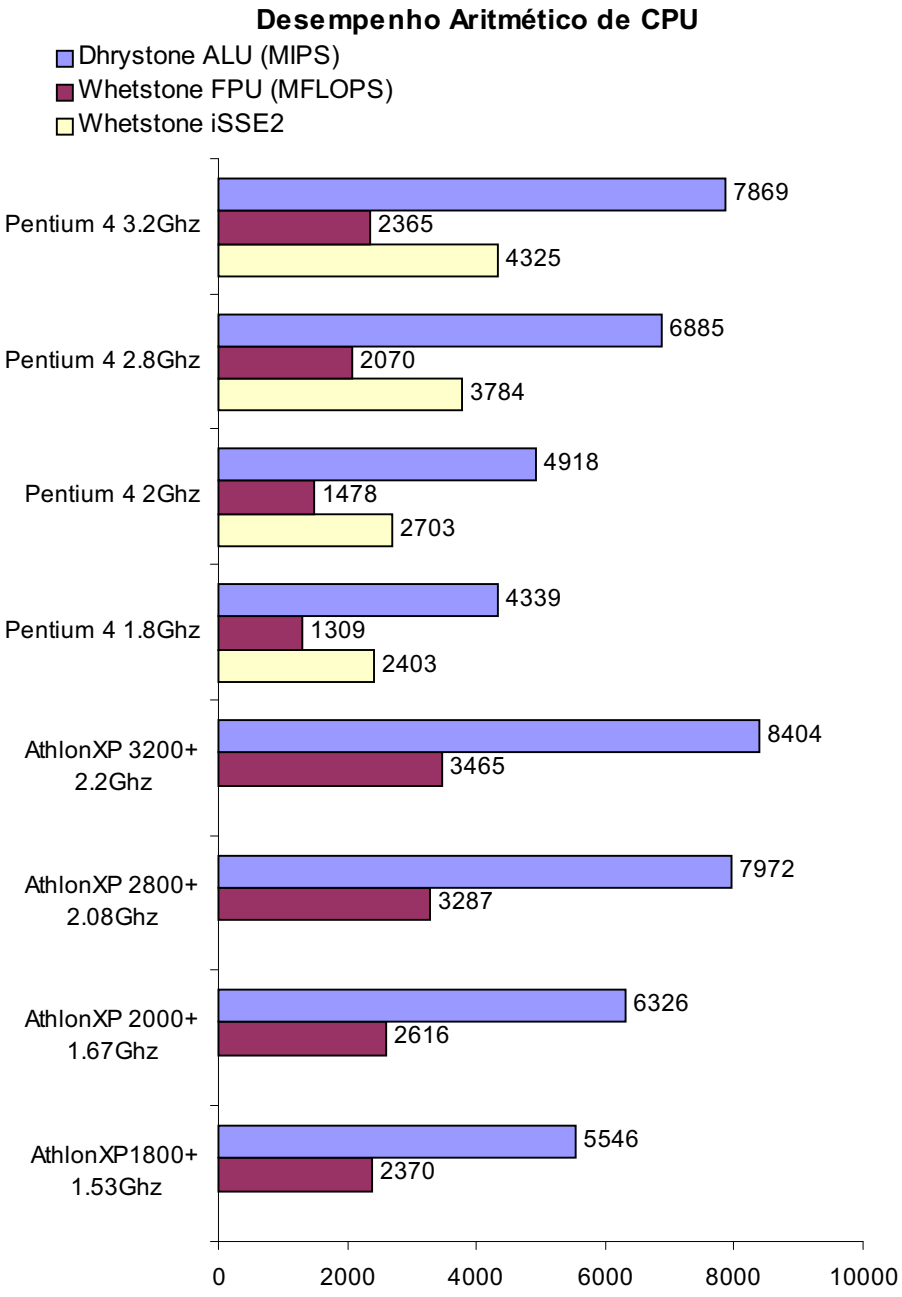

Gráfico 5: Comparativo entre diversos processadores quanto a capacidade de cálculos numéricos com aritmética de inteiros, aritmética de ponto flutuante e manipulação de strings. Testados pelo SisSoftware Sandra 2004.

4.5.4 Justificativas da análise proposta

O processador Athlon XP possui uma arquitetura superescalar de nove pipelines funcionando em paralelo, sendo que três são para instruções de ponto flutuante, três para instruções aritméticas e três para microinstruções, e todos esses pipelines possuem onze estágios. Possui também uma cache L1 de 128 KB sendo sessenta e quatro para dados e sessenta e quatro para instruções e uma cache L2 de 256 KB, e finalmente o seu barramento externo é de 266MHz (2x133Mhz).

O processador Pentium 4 possui um cache L1 de apenas 8 KB para dados, mas isso se justifica em duas inovações que compensam esta aparente deficiência. A primeira é que graças ao tamanho reduzido, o pequeno cache de dados tem um tempo de latência menor, ou seja, é mais rápido que o cache L1 encontrado no Pentium 3 e no Athlon XP. A segunda é que o Cache de instruções por sua vez foi substituído pelo *Execution Trace Cachê de 12 Kμops*, que ao invés de armazenar instruções, armazena diretamente micro operações, que são as instruções já decodificadas, prontas para serem processadas. Isto garante que o cache tenha apenas um ciclo de latência, ou seja, o processador não perde tempo algum ao utilizar um dado armazenado no *Trace Cache*.

Pelo fato do cache L1 ser pequeno e o cache L2 ser bem mais requisitado com 256 KB operando na mesma freqüência do processador, o seu barramento externo de 100MHz é multiplicado por 4 operando a 400 MHz. Em sua arquitetura também superescalar o Pentium 4 possui cinco pipelines sendo dois para intruções aritméticas, dois para geração de endereçamento e apenas um para ponto flutuante. Essas pipelines são a linha de produção das unidades de execução. Cada elemento da pipeline é responsável em executar certas microoperações. O Pentium 4 utiliza pipeline de 20 estágios que tem o dobro de um Pentium 3 (dez estágios) e praticamente o dobro da de um *Athlon* (onze estágios). O grande benefício disso é que a UCP pode operar com freqüências mais elevadas, pois os núcleos de

processamento tornam-se mais simples e possuem menos portas lógicas. Quanto mais compacto maior a velocidade e escalabilidade. O longo pipeline é uma das supostas razões que levou o Pentium 4 a perder para Athlon. Em termos de desempenho, é preciso que ele atinja freqüências muitos mais elevadas do que de um processador de pipeline mais curta, se quiser competir.

Outro problema dessa microarquitetura é um erro de prognóstico (técnica utilizada para preencher a pipeline antecipadamente) que será muito mais crítico, pois vários ciclos serão perdidos [9].

A realidade é que o Pentium 4 1.8GHz, com o núcleo Willamette, é a versão antiga do Pentium 4, no qual o desempenho é bem menor que o seu concorrente direto equivalente da AMD. A Intel, para corrigir a arquitetura e competir lado a lado da AMD, lançou o Pentium 4 com o núcleo *Northwood,* que possui uma cache L2 de 512Kb e opera com velocidades entre 1.4GHz a 2.6GHz, sendo o barramento externo de 100MHz multiplicado por 4, no total de 400MHz . O núcleo *Northwood "A*" opera com velocidades entre 2.26GHz a 3.6GHz, cache L2 de 512Kb e barramento externo de 533MHz a 800MHz. [11]

A partir dos processadores de 3,06 GHz o Pentium 4 vêm equipado com o recurso de multiprocessamento virtual, cujo nome técnico é "hyperthreading".

Neste cenário a AMD continua competindo com seu núcleo Throughbred B e mais recentemente com o núcleo Barton que atinge freqüências de até 2.25GHz e trabalha com um barramento externo de 333MHz.

Segundo pesquisas recentes sobre análise de desempenho, realizadas pelo Clube do Hardware [7], os processadores Athlon XP 2800+, da AMD, e o Pentium 4 2,8GHz, da Intel, possuem desempenho similar (ou seja, a nomenclatura PR usada pela AMD em seus processadores Athlon XP realmente indica o desempenho equivalente a um Pentium 4). Com isso, a escolha de um ou outro processador é

questão de gosto pessoal, sendo que o modelo da Intel mostrou ter uma aceitação maior ao overclock do que o modelo da AMD. Em termos tecnológicos, a estrutura interna do Athlon XP é melhor que a do Pentium 4. Apesar de terem desempenho iguais, o modelo da AMD roda a apenas 2,25 GHz, enquanto o da Intel roda a 2,80 GHz. [7].

A relação entre o custo / benefício é equilibrada em ambos os casos, sendo os processadores da AMD mais baratos. Atualmente o processador Pentium 4 de 3,06GHz está custando duas vezes mais que um Athlon XP 3000+, sendo uma opção inviável, uma vez que seu desempenho é ligeiramente maior que o do Athlon. O Pentium 4 de 2.8GHz está custando cerca de 13% a mais que o Athlon já citado. O Pentium possui uma banda bastante larga, em termos de barramento externo (as atuais versões estão em 800Mhz), o que garante transmissão de altas taxas de dados entre o processador e a memória principal. Já o processador da AMD trabalha com barramento externo de 333Mhz, o que limita a taxa de transferência de dados entre a memória e o processador, porém seu custo e os demais hardwares compatíveis (placas-mãe, cooler, memória) são bem mais viáveis.

O Pentium 4, com o núcleo Willamette já saiu de linha e possui um desempenho inferior a um Athlon XP equivalente. A linha da AMD Athlon XP 1700+ até 2600+ operando entre 1466MHz e 2083MHz oferece um custo / benefício bastante interessante, satisfazendo a maioria das necessidades do usuário. Já do Athlon XP 2700+ para cima, a escolha é uma questão pessoal e vai depender da necessidade em que vai ser empregado o microcomputador. Ambos os casos, para funcionarem em condições eficientes e eficazes necessitam de placa mãe de boa qualidade.

5 Conclusão

Ao longo deste trabalho foram apresentadas as técnicas e os componentes básicos para análise de desempenho de computadores, utilizando dois softwares: um programa público e um software de fácil utilização e execução, especialmente desenvolvido para medir o desempenho de computadores quanto à cálculos com inteiros e ponto flutuante. Além de focar os aspectos teóricos envolvidos no desempenho de computadores.

A análise dos resultados fornecidos pelos testes realizados durante o desenvolvimento do trabalho juntamente com o programa WinBenchGLC, fornece indicadores mais consistentes sobre o desempenho esperado do microcomputador testado. Estes indicadores permitem que um pesquisador se sinta mais à vontade em um processo de seleção e compra de equipamentos de acordo com as demandas do seu projeto de pesquisa.

Em relação aos testes realizados entre os computadores da marca PENTIUM 4 e ATHLON XP concluiu-se que a arquitetura do ATHLON mostrou ser mais eficiente tanto em cálculos aritméticos e manipulação de *strings* quanto em operações que envolvem ponto flutuante.

A expectativa deste trabalho, em futuras aplicações, envolve verificar avaliações mais concretas dos processadores, utilizando os demais tipos de benchmarks citados no decorrer do trabalho e analisando a performance na perspectiva do desempenho global, assim como disponibilizar o código fonte do programa para que sejam criadas futuras versões.

6 Bibliografias

• [1] PATTERSON, David. Hennessy, Jonh. *Computer Architecture – A Quantitative Approach, 3*a.ed., Morgan Kaufmann Pub, Inc, 2003.

• [2] PATTERSON, David. Hennessy, Jonh. *Computer Organization & Design: theHardware, Software.* 2a.ed. San Francisco, Morgan Kaufmann, Inc, 1997.

• [3] PATTERSON, David. Capturado OnLine em 23/07/2003, no site: http.cs.berkeley.edu/~patterson . Computer Architecture and Engineering.

• [4] JAIN, Raj. *The Art of Computer Systems Performance Analisis*. John Wiley & Songs, Inc, 1991.

• [5] MONTEIRO, Mário. *Introdução à Organização de Computadores*. 3a.ed. Rio de Janeiro: Livro Técnicos e Científicos AS.A. 1996.

• [6] Alberto José Proença, *Arquitetura e Organização de Computadores*, capturado On-Line em 12/08/2003, no site: http://gec.di.uminho.pt/discip/TextoAC/indice.html

• [7] Clube do Hardware : *Testes de processadores AMD e Pentium,* Gabriel Torres, capturado On-Line em 12/08/2003 no site: www.clubedohardware.com.br

• [8] SPEC, *Standard Performance Evaluation Corporation*. 1996/2000. Capturado OnLine em 19/07/2003, no site : http://www.spec.org

• [9] TORRES, Gabriel. *Hardware Curso Completo*. 3a.ed. Rio de janeiro: Axcels Books, 1999.

• [10] AMD Advanced Micro Devices, Inc. Site oficial da AMD. www.amd.com

• [11] Intel, Site Oficial. www.intel.com

• [12] Tratamento de Imagens, MARQUES FILHO; VIEIRA NETO, 2000

• [13] OSSAMU, César. B*enchmarks*. Capturado OnLine em 09/2003, no site http://www.inf.ufrgs.br/procpar/disc/cmp134/trabs/T1/001/benchmarks/Bencmarks_p2.html

• [14] Ivan Luiz Marques Ricarte*, Organização de Computadores (EA960)* - 1999, capturado On-Line em 09/2003, no site: http://www.dca.fee.unicamp.br/courses/EA960/

• [15] john.a.stockton@btopenworld.com, suite com funções para auxiliar a explorar os números primos inteiros, version: 1.1, 13 June 2002.

• [16] Victory Fernandes, *Implementação do Processo de Eliminação Gaussiana*, capaturado On-Line em agosto/2003 no site: http://www.inf.ufsc.br/~prass/home.html contato: victory@e-net.com.br.

• [17] Matemática Aplicada, capturado Online no site:
http://www.terra.com.br/matematica/arq12-2.htm

• [18] Silvia Carla Menti Propicio, *APLICAÇÕES DE NÚMEROS COMPLEXOS,* Universidade de Caxias do Sul , Centro de Ciências Exatas e Tecnologia , Departamento de Matemática e Estatística. Capturado Online no site:
http://www.ucs.br/ccet/deme/emsoares/inipes/complexos/

• [19] TIM, Wegner. BRANDERHORST, Pieter. TYLER, Bert. PETERSON, Mark. Fractais para Windows; tradução Cláudio José da Costa. – Rio de Janeiro : Berkeley, 1993.

ANEXO A

Informações sobre o Sisoftware Sandra 2004 Professional

O que é o Sandra?

SiSoftware Sandra (the System ANalyser, Diagnostic and Reporting Assistant) (o analisador de sistema, assistente de diagnóstico de relatório) é um utilitário de informação e diagnóstico. Ele deve prover informações (incluindo não documentadas) que você precisa saber sobre o seu hardware, software e outros dispositivos.

Sandra prove um nível similar de informações às do Norton SI, Quarterdeck, Winprobe/Manifest, etc. Ele é nativo de aplicações Win32 bits e opera nos novos sistemas operacionais de 64bit (Windows XP Service Pack 1). Ele também provê informações de windows 16 bits e DOS sem usar muitos outros programas de informação.

Algoritmos presente no Sandra:

Dhrystone

O benchmark de Dhrystone original ainda é usado amplamente para medir desempenho de UCP. Existem várias versões e variantes do algoritmo original. O benchmark é projetado para conter uma representativa carga de cálculos, principalmente numéricos, usados por diversas aplicações. Infelizmente isto nem sempre representa um verdadeiro índice de desempenho real, mas é útil comparar a velocidade de vários UCPs.

O benchmark Dhrystone usado aqui é "multi-threaded", 32/64-bit variante do original. Até 64 UCPs são suportadas em sistemas de SMP. O resultado é determinado medindo o tempo que o algoritmo leva para executar algumas instruções sucessivas. Devido a várias mudanças, o resultado não está diretamente comparável com outro benchmarks de Dhrystone. Porém o MIPS (Milhões de Instruções Por Segundo) deveria ser o mesmo para o mesmo sistema (+5-10% variação) entre benchmarks.

Enquanto o benchmark original não computa nada, esta versão confere os resultados por via de dúvida, examinando se há problemas com o UCP/Memória.

Whetstone

O benchmark de Whetstone é extensamente usado na indústria de computador como uma medida de FPU ou desempenho de Co-processador. A aritmética de ponto flutuante é muito significante em programas que requerem um Co-processador. Estes testes são principalmente científicos e criam estatísticas que ajudam na designação de processadores com melhor desempenho aritmético.

O benchmark de Whetstone usado aqui é "multi-threaded" de 32/64-bit, variante do original que roda debaixo do UNIX. Até 64 UCPs são suportadas em sistemas de SMP. O resultado é determinado medindo o tempo que o algoritmo leva para executar algumas sucessões de instruções de flutuante-ponto. Devido a várias mudanças, o resultado não está diretamente comparável com outro benchmarks de Whetstone. Porém o MFLOPS (Milhões de Operações Flutuantes por Segundo) deveria ser o mesmo para o mesmo sistema (+5-10% variação) entre benchmarks.

Whetstone iSSE2

Com a introdução da tecnologia SSE2, presente no Pentium 4, e sua capacidade de manipular pontos flutuadores duplos (64-bit) é possível escrever código que não usa diretamente a FPU. Aproveita-se, então, neste algoritmo, as intruções SSE2 e é tirado proveito do modo SIMD de operação.

Endereço na Internet: http://www.sissoftware.com

ANEXO B

CPUz.

Versão 1.20a de 11/04/2003
Contato : cpuz@cpuid.com
Página Web: http://www.cpuid.com/cpuz.php

Cpuz é um freeware de livre distribuição, que reporta as características técnicas dos computadores testados. Uma página html é gerada como o exemplo seguinte: (próxima página)

CPU-Z Report

CPU-Z version **1.20a**.

CPU(s)	
Number of CPUs	1

Code Name	Barton
Specification	AMD Athlon(TM) XP 2800+
Family / Model / Stepping	6 A 0
Extended Family / Model	7 A
Core Stepping	
Technology	0.13 µ
Supported Instructions Sets	MMX, Extended MMX, 3DNow!, Extended 3DNow!, SSE
CPU Clock Speed	2083.1 MHz
Clock multiplier	x 12.5
Front Side Bus Frequency	166.6 MHz
Bus Speed	333.3 MHz
L1 Data Cachê	64 KBytes, 2-way set associative, 64 Bytes line size
L1 Instruction Cachê	64 KBytes, 2-way set associative, 64 Bytes line size
L2 Cache	512 KBytes, 16-way set associative, 64 Bytes line size
L2 Speed	2083.1 MHz (Full)
L2 Location	On Chip
L2 Data Prefetch Logic	Yes
L2 Bus Width	64 bits

Mainboard and chipset	
Motherboard manufacturer	ASUSTeK Computer INC.
Motherboard model	A7V8X-X, REV 1.xx
BIOS vendor	Award Software, Inc.
BIOS revision	ASUS A7V8X-X ACPI BIOS Revision 1005
BIOS release date	05/08/2003
Chipset	VIA KT400 (VT8377) rev. 0
Southbridge	VIA VT8235 rev. 0
Sensor chip	FFFF

AGP Status	enabled, rev. 3.5

AGP Side Band Addressing	supported, not enabled

Memory	
DRAM Type	DDR-SDRAM
DRAM Size	512 MBytes
DRAM Frequency	166.6 MHz
FSB:DRAM	1:1
DRAM Interleave	4-way
CAS# Latency	2.5 clocks
RAS# to CAS#	3 clocks
RAS# Precharge	3 clocks
Cycle Time (TRAS)	7 clocks
# of memory modules	1
Module 0	DDR-SDRAM PC3200 - 512 MBytes

Software	
Windows version	Microsoft Windows XP Workstation Service Pack 1 (Build 2600)

ANEXO C

Instruções suportadas pela AMD e INTEL

Set de Instruções	Descrição	Fabricantes	Introduzido no
OPTIMIZED FOR 3DNOW!	3DNow!	AMD	K6-2
OPTIMIZED FOR 3DNOW!	3DNow! Enhanced	AMD	Athlon (K7)
OPTIMIZED FOR 3DNOW!	AMD MMX Extensions	AMD	Athlon (K7)
OPTIMIZED FOR 3DNOW!	3DNow! Professional	AMD	Athlon 4 (A4)
MMX	Multi-Media eXtensions (MMX)	Intel	Pentium MMX
SSE	Streaming SIMD (SSE)	Intel	Pentium III
SSE2	Streaming SIMD2 (SSE2)	Intel	Pentium 4

ANEXO D

Trechos de códigos do programa WinBenchGLC

Linguagem : Object Pascal (Delphi versão 7)

Constantes

```
const
  IntValor = 1800000; //valor de calibragem dos teste de inteiros
  FloatValor = 1800000; //valor de calibragem dos teste de ponto flutuante
```

Função Soma do módulo aritmética de inteiros

```
function TFormPrincipal.SomaInt(var Tempo : TDateTime): String;

var vetor : array[1..100] of Integer;
    i,j : integer;
    soma : int64;
    antes,depois : TDateTime;
begin
  soma := 1;
  antes := now;
  for j := 1 to IntValor do
  for i := 1 to 100 do
  begin
    vetor[i] := random(32000);
    soma := soma + vetor[i]
  end;
```

```
  depois := now;
  vetor[1] := (Soma + 1) - Soma;
  Tempo := depois-antes;
  SomaInt := FormatDateTime(TempoFormatado,Tempo);
end;
```

Função Soma do módulo aritmética de ponto flutuante

```
function TFormPrincipal.SomaFloat(var Tempo : TDateTime): String;
var vetor : array[1..100] of Extended;
    i,j : integer;
    Soma : Comp;
    antes,depois : TDateTime;

begin
  soma := 1;
  antes := now;
  for j := 1 to FloatValor do
  for i := 1 to 100 do
  begin
    vetor[i] := random(32000);
    soma := soma + vetor[i]
  end;
  depois := now;
  vetor[1] := (Soma + 1) - Soma;
  Tempo := depois-antes;
  SomaFloat := FormatDateTime(TempoFormatado,Tempo);
end;
```