

MAC0499 - Trabalho de Formatura Supervisionado



IME - Instituto de
Matemática e Estatística

Monografia

Previsão e prevenção de falhas em sistemas computacionais

Segundo semestre de 2009

Supervisor

Prof. Dr. Siang Wun Song

Integrantes

Daniel Hanna Leite El Ottra - 5123031

Thiago Miranda Ferreira - 5123010

Sumário

1. Introdução
2. Breve descrição dos conceitos utilizados
 - 2.1. Contexto histórico
 - 2.2. Reguladores lineares
 - 2.3. Reguladores chaveados
 - 2.4. Tipos de filtro
 - 2.4.1. Low-pass
 - 2.4.2. High-pass
 - 2.4.3. Implementação de filtros
 - 2.4.4. Capacitores
 - 2.4.5. Indutores
 - 2.5. Funcionamento dos reguladores
 - 2.5.1. Controlador VRM
 - 2.6. Gerenciamento de energia
 - 2.6.1. ACPI (Advanced Configuration & Power Interface)
3. Metodologia
 - 3.1. Hardware
 - 3.1.1. A.T.E. (Automated Test Equipment)
 - 3.1.2. A escolha das placas-mãe
 - 3.1.3. Potenciômetros na placa-mãe PCChips
 - 3.2. Software
 - 3.2.1. DSL (Damn Small Linux)
 - 3.2.2. O script de testes
 - 3.2.3. O código de teste em C
4. Resultados e produtos obtidos
 - 4.1. Experimentos com potenciômetros
 - 4.2. Hipótese: Interferência eletromagnética
 - 4.3. Identificação de erros por capacitores danificados
 - 4.4. Detecção do fim da vida útil de capacitores eletrolíticos
5. Conclusões e futuro do projeto
6. Referências bibliográficas

1 Introdução

Nas últimas décadas, observamos uma crescente popularização dos computadores pessoais, que se tornaram itens indispensáveis em nosso cotidiano. Muito mais do que isto, estas máquinas passaram a exercer papéis fundamentais em ambientes corporativos, quaisquer que sejam seus ramos de atuação no mercado.

Neste contexto, torna-se desejável a existência de um projeto que objetiva estudar a previsão e prevenção de falhas em sistemas computacionais. A intenção deste trabalho é criar uma metodologia mais abrangente e completa do que as já conhecidas SMART e POST. Desta forma, temos como objetivo trabalhar em três etapas básicas: estudo de falhas de software, estudo de falhas em hardware X86, e um estudo sobre uma possível interação entre estes dois tipos de falhas.

O projeto, que vem sendo desenvolvido desde meados de 2006, tem como tema “previsão e prevenção de falhas em sistemas computacionais”, e será desenvolvido através de uma série de análises sistêmicas de hardware e software, com foco em componentes de placas mãe diretamente ligados à CPU, além de técnicas de programação para contornar limitações provocadas por componentes de baixa qualidade, ou mesmo por desgastes ocasionados pelo envelhecimento natural dos mesmos, impedindo que determinados resultados sejam fielmente observados.

O principal objetivo deste projeto é obter sucesso na tentativa de antecipar falhas causadas por componentes desgastados e/ou de baixa qualidade, utilizando como base um sistema AMD de 32 bits.

2 Breve descrição dos conceitos envolvidos

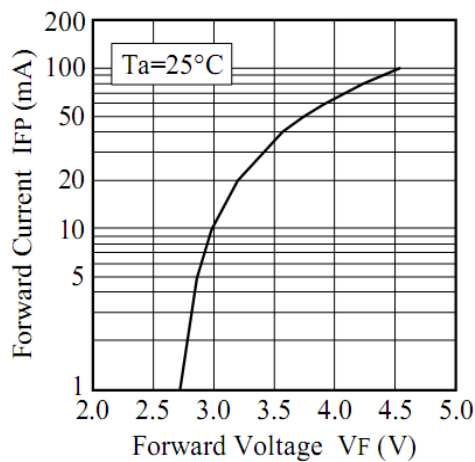
2.1 Contexto histórico

Em 1879, Thomas Alva Edison inventou a lâmpada de vácuo com filamento de carbono, que gerava luz a partir da utilização de pouca corrente elétrica. Esta foi a primeira vez que este tipo de experimento foi realizado de forma prática e viável. A lâmpada incandescente, tal como a conhecemos hoje, produz luz através do aquecimento do filamento, ocasionado pela resistência do material à passagem de elétrons.

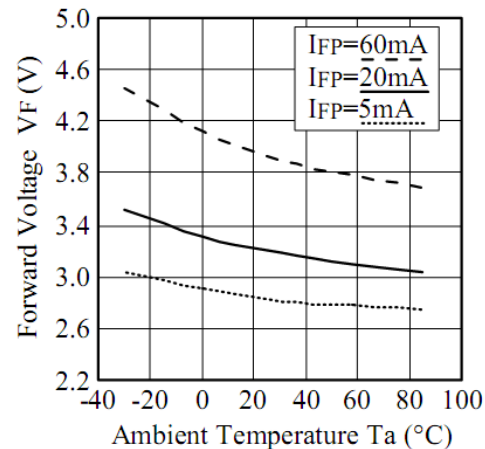
Atualmente, devido à crise energética, novas tecnologias foram desenvolvidas com o objetivo de melhorar a eficiência dos sistemas de iluminação. O LED (light-emitting diode) por exemplo, é um tipo de semicondutor, e foi produzido pela primeira vez em 1962, pela General Electric. Sua eficiência pode ser superior a 100 lúmens por watt.

Ao contrário da lâmpada, não existe um padrão de voltagem para os LEDs. Normalmente o intervalo varia entre 1.8V e 3.7V, e o valor exato depende da temperatura do ambiente. Para simplificar, a maioria dos projetos adiciona um resistor em série, limitando a corrente máxima no circuito.

■ Forward Voltage vs.
Forward Current

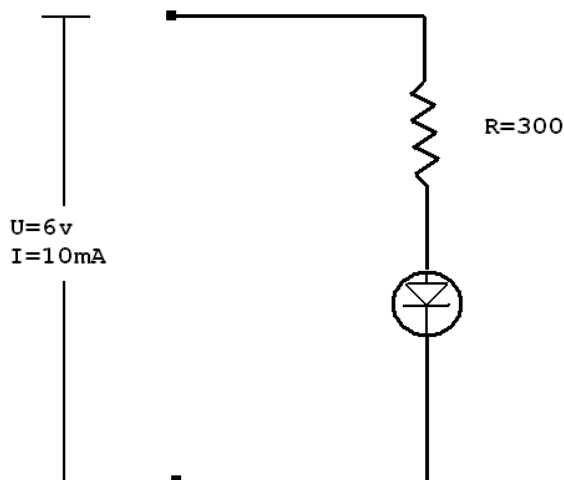


■ Ambient Temperature vs.
Forward Voltage



Fonte: Nichia Corporation

Através de uma rápida análise dos gráficos anteriores, e supondo uma temperatura de 25°C, temos o seguinte esquema:



Fonte: Imagem ilustrativa criada para o projeto

De acordo com o primeiro gráfico, temos que o LED permite uma corrente de 10mA, quando submetido a uma tensão de 3V. Logo, para transformar os 6V em 3V, é preciso adicionar um resistor em série, de tal modo que, ao ser submetido à corrente total, este diminua o potencial inicial em 3V. Usando a Primeira Lei de Ohm (potencial é igual a resistência multiplicada pela corrente), temos:

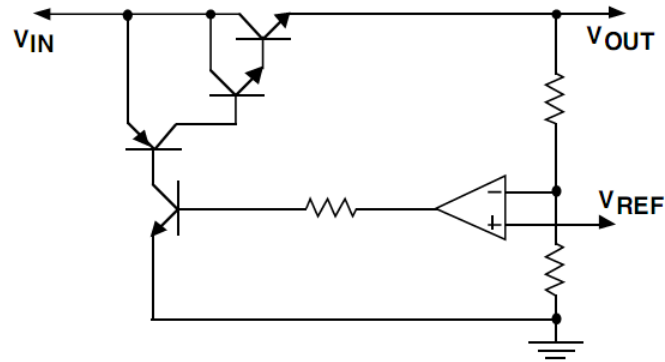
$$U = R \cdot i$$

$$3 = R \cdot 0.01$$

$$R = 300\Omega$$

No entanto esta solução só funciona para valores de tensão próximos de 6V pois, acima deste valor, existe a possibilidade do LED ser danificado, ou ainda do mesmo não emitir luz, caso a tensão fique muito baixa. Para contornar esta situação, foram desenvolvidos os reguladores lineares.

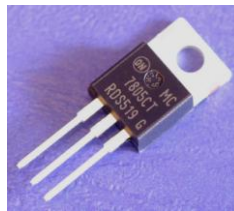
2.2 Reguladores lineares



Exemplo da estrutura interna do regulador

Fonte: National Semiconductor

Os reguladores lineares atuam como o resistor do esquema visto anteriormente, e possuem a vantagem de ajustar a tensão do circuito de modo independente da corrente ou da tensão de entrada. Esta facilidade permitiu o uso deste tipo de circuito em diversas aplicações, inclusive nos microprocessadores.



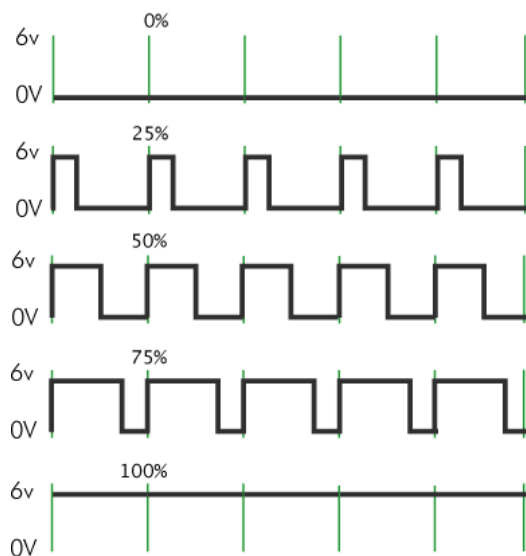
Exemplo de um regulador linear para 5V

Fonte: Google imagens (busca pelo termo "regulador 7805")

De forma diferente dos LEDs, um processador pode aumentar o seu consumo dependendo da sua utilização, porém isto será abordado na segunda parte deste trabalho.

2.3 Reguladores Chaveados

Para facilitar a compreensão dos reguladores chaveados, devemos antes abordar a técnica de PWM. PWM é a sigla de *Pulse Width Modulation*, e seu nome já dá uma idéia do que se trata. A modulação de largura de pulso é utilizada em diversas situações, e uma dessas aplicações é a de controlar a potência de equipamentos elétricos. Voltando ao nosso exemplo do LED, e supondo que agora precisamos diminuir a potência, podemos aplicar uma série de pulsos, como no gráfico abaixo:



Fonte: Arduino

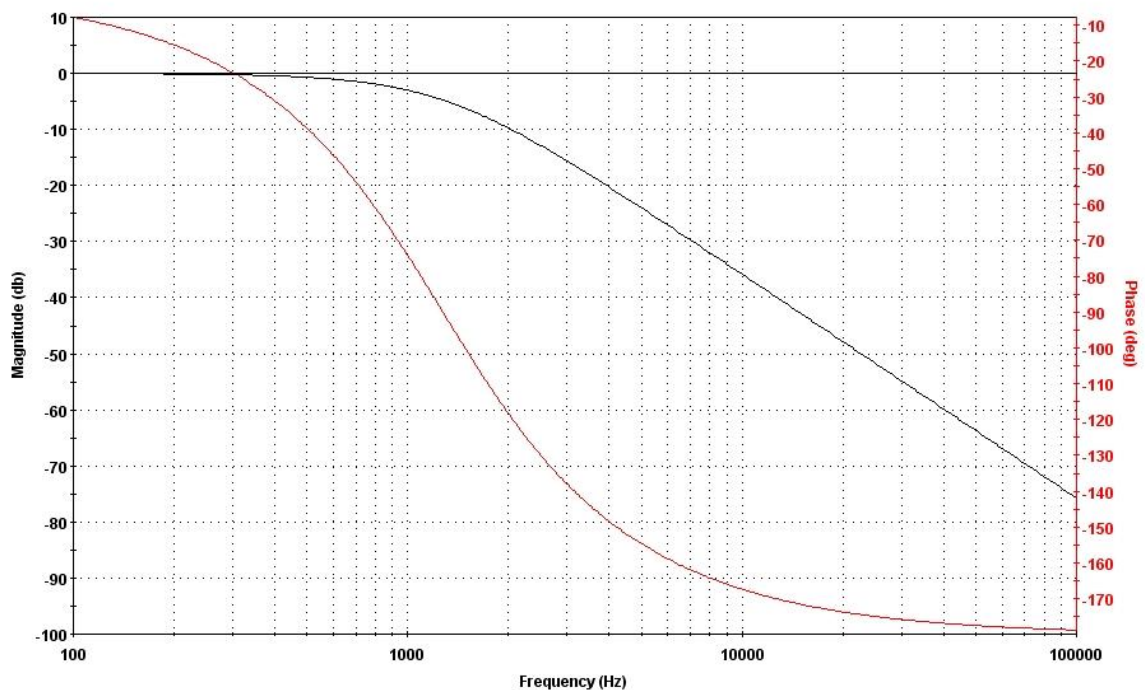
Com isso o LED ficará parte do tempo ligado e parte do tempo desligado. Para que o brilho pareça constante basta que a frequência fique suficientemente alta. Porém como podemos aplicar esse princípio para fornecer energia para os nossos equipamentos digitais? Uma maneira possível seria aplicar algum tipo de filtragem para suavizar os pulsos de modo a produzir uma saída de tensão constante.

2.4 Tipos de filtro

Para melhor entender o funcionamento dos filtros iremos citar os mais comuns e destacar suas propriedades.

2.4.1 Low-pass

Este tipo de filtro é muito utilizado para remover componentes de alta frequência, e é comumente encontrado em fontes e sistemas de aquisição de dados. Quanto maior a frequência do sinal, maior será a atenuação. O gráfico abaixo mostra a resposta de frequência para um filtro Bessel de segunda ordem (curva preta).

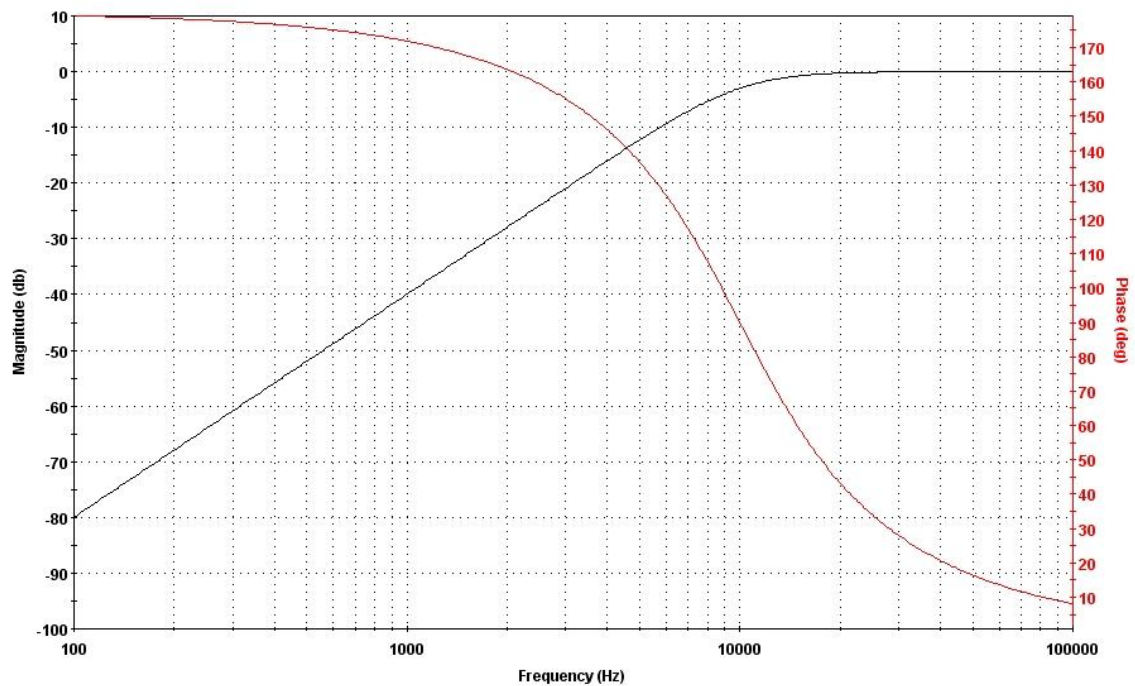


Em vermelho, podemos observar o atraso de fase provocado pelo filtro.

Fonte: Imagem gerada para o projeto pelo software Filterlab

2.4.2 High-pass

Ao contrário do anterior, este filtro rejeita freqüências baixas, sendo muito útil em sistemas de som, onde geralmente é mais interessante passar os sinais de alta freqüência para falantes específicos. O gráfico abaixo exemplifica a resposta para um filtro Butterworth (curva preta).



Em vermelho, podemos observar o atraso de fase provocado pelo filtro.

Fonte: Imagem gerada para o projeto pelo software Filterlab

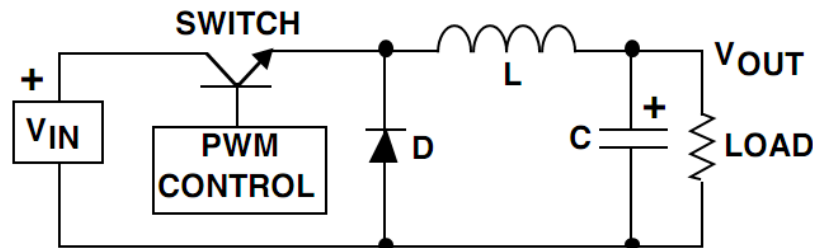
É importante ressaltar que um parâmetro importante na escolha do filtro é a freqüência de corte. Esta é determinada quando ocorre uma atenuação de um sinal puro de uma freqüência F^1 em 3dB. Dizemos então que F^1 é a freqüência de corte.

2.4.3 Implementação de filtros

Atualmente, graças aos avanços da indústria de semicondutores podemos implementar filtros de 4 maneiras diferentes:

- Filtros passivos
- Filtros ativos
- Filtros digitais
- Filtros por software

Normalmente os filtros utilizados na conversão de energia são do tipo passivo, já que apenas eles toleram grandes quantidades de corrente com o menor custo. Voltando ao exemplo da modulação de pulso, e lembrando que podemos aplicar um filtro Low-pass para suavizar a ondulação dos pulsos, chegamos ao esquema abaixo:



Estrutura simplificada de um regulador chaveado assíncrono.

Fonte: National Semiconductor

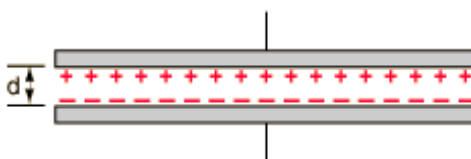
Na figura, L e C são os componentes que constituem o filtro do regulador, e representam, respectivamente, o indutor e o capacitor. Para entender a função do diodo D, precisamos explicar um pouco sobre capacitores e indutores.

2.4.4 Capacitores

Estes componentes armazenam energia elétrica sob a forma de um campo eletrostático. Esta propriedade é chamada de capacitância ou capacidade (C), e é medida pelo quociente da quantidade de carga (Q) armazenada pela diferença de potencial ou tensão (V) aplicada entre as placas:

$$C = \frac{Q}{V}$$

No SI (Sistema Internacional de Unidades), um capacitor tem a capacidade de um Farad (F) quando um coulomb de carga causa uma diferença de potencial de um Volt (V) entre as placas. O Farad é uma unidade muito grande e por isso são utilizados valores expressos em microfarads (μF), nanofarads (nF) ou picofarads (pF).



Fonte: *HyperPhysics*

A capacitância de um capacitor de placas paralelas, ambas com áreas A idênticas, separadas por uma distância constante d , é aproximadamente igual a:

$$C = \frac{k\epsilon_0 A}{d}$$

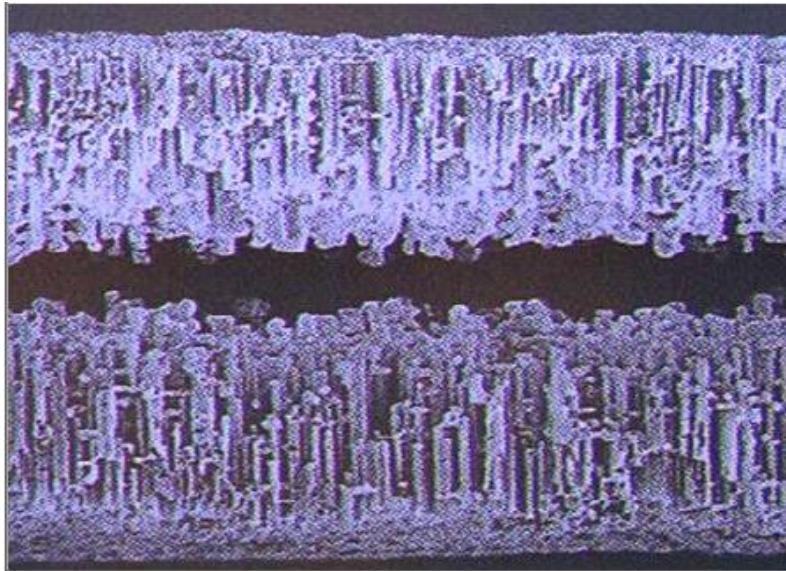
Onde,

C é a capacitância em Farads

k é a constante dielétrica (relativa ao isolante)

ϵ_0 é a permissividade eletrostática do vácuo

Para capacitores eletrolíticos, a área da superfície entre as placas é aumentada até 150 vezes, através de um processo de corrosão induzido durante sua produção. Entre estas placas encontra-se um papel impregnado com fluido eletrolítico que permite o transporte iônico entre os eletrodos.



Corrugações das folhas de alumínio num microscópio eletrônico.

Fonte: Rubycon Corporation

Os capacitores normalmente possuem uma vida útil dependente de sua temperatura. Observando a equação abaixo, pode-se notar que, para cada 10°C abaixo da temperatura máxima, dobramos a vida útil estimada dos capacitores:

$$L = L_0 \times 2^{\frac{T_{\max} - T_a}{10}}$$

Onde,

L: Vida útil estimada (em horas)

L_0 : Duração estimada na temperatura máxima (em horas)

T_{\max} : Temperatura máxima de operação (em °C)

T_a : Temperatura ambiente

2.4.5 Indutores

A principal característica dos indutores é a de se opor à mudança da corrente. O comportamento dos indutores pode ser representado pela equação abaixo:

$$V = L(di/dt)$$

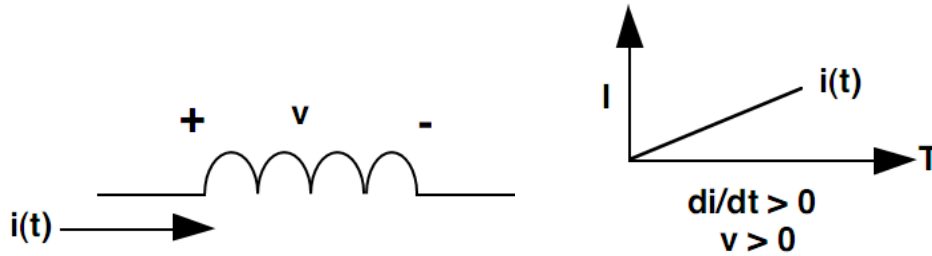
Onde,

V é a tensão nos terminais do indutor

L é a indutância

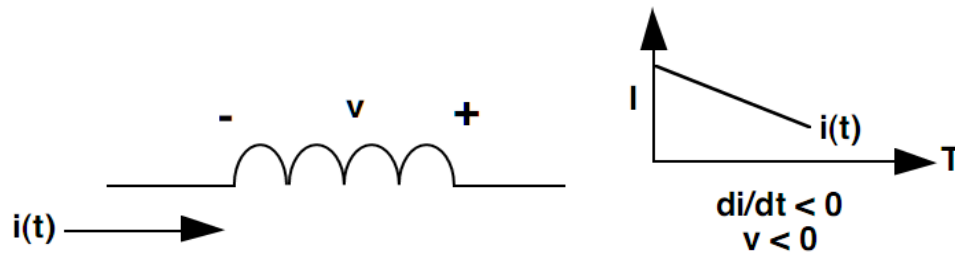
di/dt corresponde a variação instantânea da corrente que passa pelo indutor

Quando temos um aumento na corrente que passa pelo indutor, surge uma tensão positiva entre os terminais, tal como representado abaixo:



Fonte: National Corporation

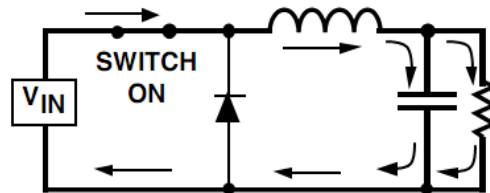
Por outro lado se houver uma diminuição da corrente, teremos uma tensão negativa no indutor, como pode ser visto abaixo:



Fonte: National Corporation

Caso a corrente se torne constante, não teremos nenhuma tensão relevante no indutor. Com isso podemos entender melhor como o regulador funciona, bastando dividir a análise em duas partes.

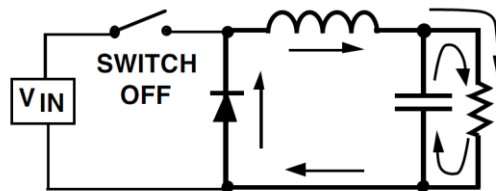
2.5 Funcionamento dos reguladores



Estado 1: com o transistor aberto

Fonte: National Corporation

Na figura acima, as setas indicam a corrente. Como a tensão que entra no regulador é maior que a tensão de saída, temos um aumento na corrente do indutor, que é dividida entre o capacitor de saída e a carga (por exemplo um processador).



Estado 2: com o transistor fechado

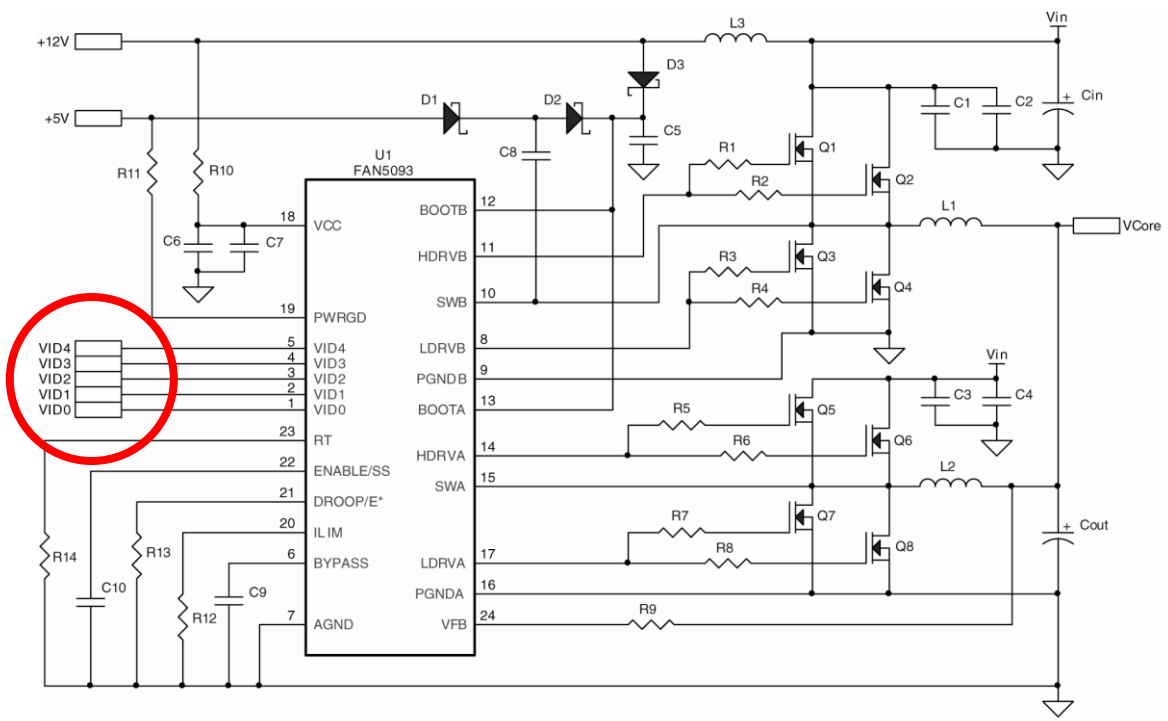
Fonte: National Corporation

Agora o indutor passa a fornecer energia para a carga, junto com o capacitor de saída. Como a corrente no indutor tende a diminuir, surgirá uma tensão negativa na entrada do indutor. O diodo então permitirá a passagem de corrente, fechando o circuito.

É claro que precisamos de algo para comandar o transistor, de modo que a tensão fique dentro dos limites. Normalmente essa é a tarefa do controlador de chaveamento.

2.5.1 Controlador VRM

Na maioria das placas-mãe modernas o controlador identifica a tensão necessária através dos pinos VID. Através de uma porta de 5 bits, o processador informa o valor recomendado para manter o processador dentro de suas especificações.



Arranjo de componentes ligados ao controlador.

Em destaque encontram-se a porta de comunicação com a CPU.

Fonte: Fairchild Semiconductor

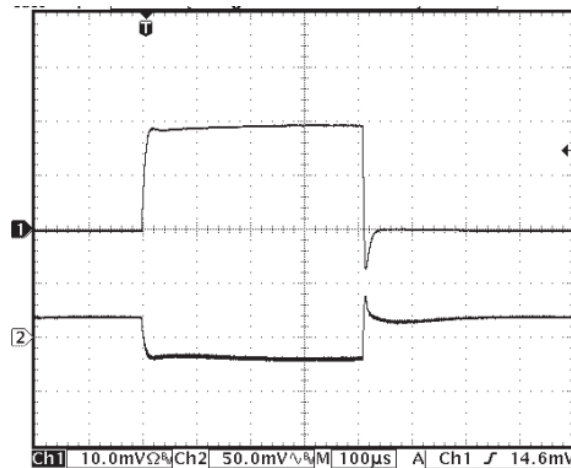
Para descobrir o valor correto, podemos utilizar a tabela fornecida pelo fabricante. É importante ressaltar que a tensão deve manter-se dentro de certos limites mesmo com grandes alterações na corrente.

VID[4:0]	VCC_CORE (V)	VID[4:0]	VCC_CORE (V)
00000	1.850	10000	1.450
00001	1.825	10001	1.425
00010	1.800	10010	1.400
00011	1.775	10011	1.375
00100	1.750	10100	1.350
00101	1.725	10101	1.325
00110	1.700	10110	1.300
00111	1.675	10111	1.275
01000	1.650	11000	1.250
01001	1.625	11001	1.225
01010	1.600	11010	1.200
01011	1.575	11011	1.175
01100	1.550	11100	1.150
01101	1.525	11101	1.125
01110	1.500	11110	1.100
01111	1.475	11111	No CPU

Na tabela acima, estão representados os 32 códigos VID de 5 bits, seguidos pelos respectivos valores esperados em milivolts. A tabela se aplica aos processadores Athlon.

Fonte: AMD

No gráfico abaixo temos um exemplo da reação do regulador para mudanças bruscas na corrente fornecida para o processador. Durante as transições de aumento e diminuição da corrente ocorrem flutuações na tensão.



Acima, a linha 1 indica um pulso de 40A de corrente e na linha 2 temos queda de 50mV

Fonte: Fairchild Semiconductor

2.6 Gerenciamento de energia

Desde o surgimento do Pentium 75, os processadores Intel passaram a diminuir o consumo de energia através da instrução halt. Com o passar do tempo, novas medidas foram adotadas para aperfeiçoar e centralizar o gerenciamento de energia para outros componentes do sistema.

2.6.1 ACPI (Advanced Configuration & Power Interface)

ACPI é o conjunto de funções para gerenciamento de energia e de interfaces para o S.O. Sua adoção se iniciou em 1997 e a versão mais atual é a 4.0.

No Linux, o suporte para ACPI está disponível para as versões 2.4 e 2.6 do kernel mas, dependendo do hardware envolvido, algumas funcionalidades podem não estar disponíveis. No caso dos processadores Athlon, existem 4 estados possíveis para o processador:

- C0: Estado ativo, processador está executando instruções.
- C1: Estado halt, processador executou uma instrução halt e pode voltar ao estado C0 caso receba um sinal externo.
- C2: estado que pode ser ativado por software ou hardware. Memória cache permanece ativada.
- C3: estado semelhante ao C2, mas agora a memória cache não está disponível.

Infelizmente, no caso dos processadores Athlon só é possível diminuir drasticamente o consumo de energia se o processador estiver desconectado do barramento, algo que precisa ser configurado na ponte norte (north bridge). Tal configuração é específica do para cada tipo de chipset.

Porém, não basta ter um sistema bem configurado para economizar energia, já que para entrar no estado C1 precisamos que a CPU execute a Idle thread, e isso só acontece se não tivermos outros processos em execução.

3 Metodologia

3.1 Hardware

Após consultar diversas notas de aplicação sobre testes de reguladores percebemos que existe um padrão comum, disponibilizado abertamente pela Intel e AMD.

O teste consiste em ativar e desativar a entrada STPCLK# do processador durante a execução de algum software que eleve o consumo energético da CPU. Ao fazer isso forçamos a entrada da CPU para estados C2 ou C3, e assim provocamos grandes alterações na corrente que vai para a CPU.

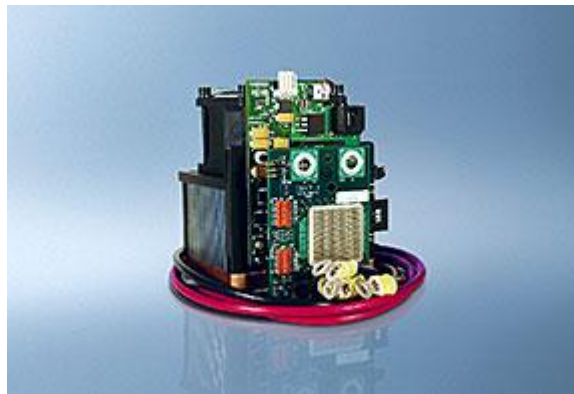
Frequência (MHZ)	Tensão Nominal	Tensão máxima	Potência máxima no estado C3 (1,3V)	Corrente Máxima@1,75V	Temperatura do Núcleo
650	1.75 V	1.85 V	5 W	22 A	90°C
700				23 A	
750				25 A	
800				26 A	
850				27 A	
900				29 A	
950				30 A	
1000				31 A	
1100				34 A	95°C
1133				36 A	
1200	38 A				

Na tabela acima temos os dados de potência e corrente para diversos processadores Athlon

Fonte: AMD

3.1.1 A.T.E. (Automated Test Equipment)

Devido à complexidade do procedimento, surgiram muitos equipamentos para facilitar o teste e a validação de plataformas para os diversos componentes de uma placa-mãe.



A ferramenta VRTT (Voltage Regulator Test Tool) da Intel auxilia na validação de plataformas.

Acima temos o modelo para o soquete mPGA V2.

Fonte: Intel Embedded Design Center

Para efetuar nossas análises e testes, nos baseamos nesses equipamentos de uso industrial, que simulam situações extremas de consumo através de flutuações de corrente ao longo do tempo. Essas flutuações são comumente chamadas de carga dinâmica.

Percebemos que o uso destes equipamentos possui alguns inconvenientes:

- É necessário que o computador seja totalmente desmontado para sua execução;
- São grandes os riscos associados ao seu mau uso, que pode causar danos por sobrecarga na placa-mãe;
- Os preços associados aos dispositivos são extremamente elevados.



Ferramenta desenvolvida pela Chroma ATE, para uso industrial.

Fonte: Chroma ATE Inc.

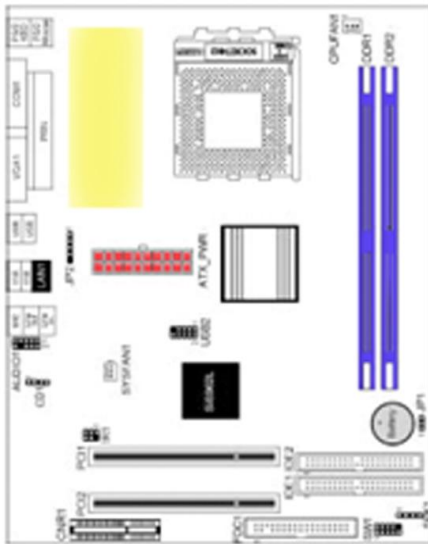
Desta forma, nossas pesquisas foram focadas em desenvolver uma solução que seja uma alternativa viável e semelhante aos equipamentos citados.

3.1.2 A escolha das placas-mãe

Para executar nossos testes, tínhamos a nossa disposição duas placas-mãe, das marcas PCChips e Asus, ambas muito populares e amplamente utilizadas no mercado. Os modelos que utilizamos foram o M810D (PCChips) e o A7V8X-X (Asus). As especificações técnicas podem ser encontradas no site dos fabricantes. O processador que utilizamos foi um AMD Athlon 1333MHz e a memória utilizada era do tipo DDR operando em 266MHz.

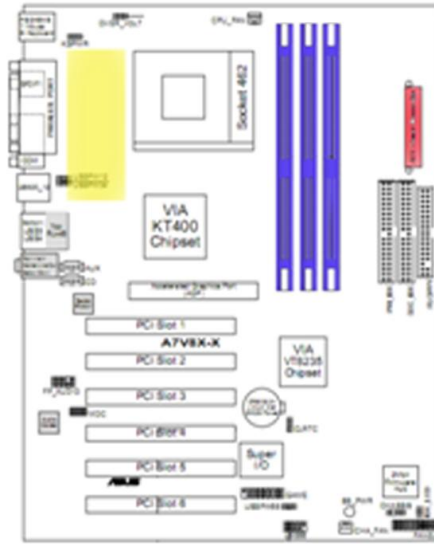
Temos abaixo os layouts das placas, com destaque para algumas regiões que se mostraram de fundamental importância para os resultados obtidos ao longo do projeto:

- Em azul os slots de memória;
- Em vermelho a entrada de energia;
- Em amarelo a localização do regulador de tensão.



PCChips, modelo M810D

Fonte: PCChips



Asus, modelo A7V8X-X

Fonte: Asus

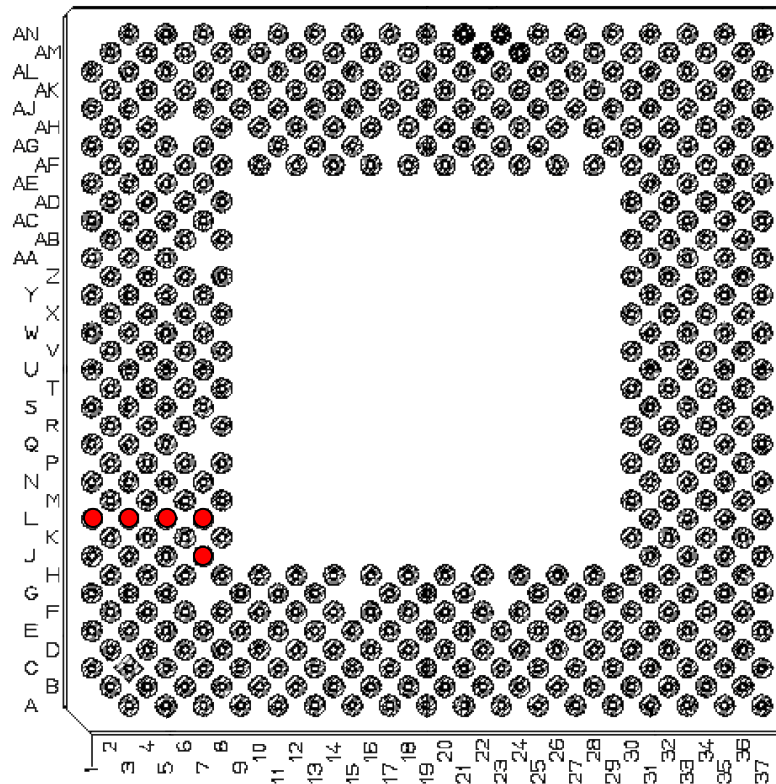
3.1.3 Potenciômetros na placa-mãe PCChips

Duas importantes alterações de hardware que realizamos ao longo de nosso projeto, foram efetuadas na placa-mãe da PCChips (M810D). Estas alterações consistiram em incluir dois potenciômetros (resistores variáveis), que nos permitiram ajustar manualmente os valores da tensão fornecida ao núcleo do processador, e a frequência de chaveamento de entrada do regulador.

O objetivo destas alterações foi simular problemas nos reguladores chaveados.

A primeira alteração que fizemos na placa teve como objetivo observar o comportamento do processador em baixa tensão, bem como o reflexo dessa modificação na execução de softwares.

Como já vimos anteriormente, o processador informa a tensão necessária através da entrada VID [4:0], e se tivéssemos uma maneira de alterar esse valor, seria possível controlar o regulador para fornecer valores mais baixos.

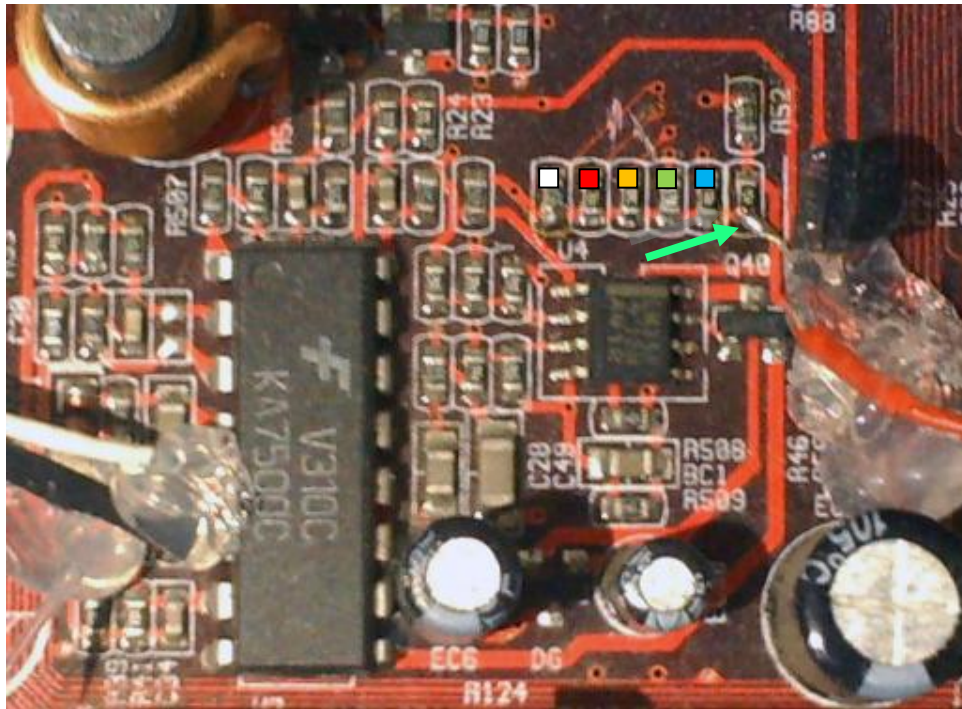


No desenho acima, estão em destaque os pinos do processador que se comunicam com o controlador de chaveamento (VID[4:0]).

Fonte: AMD

Nossa primeira idéia foi adicionar jumpers na entrada do controlador do regulador. Porém, para nossa surpresa, o regulador da placa PCChips não possuía uma entrada digital para ler o VID correto. Ao invés disso encontramos um controlador genérico que possuía apenas entradas analógicas.

Para resolver esse impasse tivemos que analisar as trilhas que ligavam o processador para a região do regulador. Ao fazer isso, chegamos aos seguintes componentes:



Fonte: Foto ampliada das mudanças efetuadas na placa mãe PCChips M810D

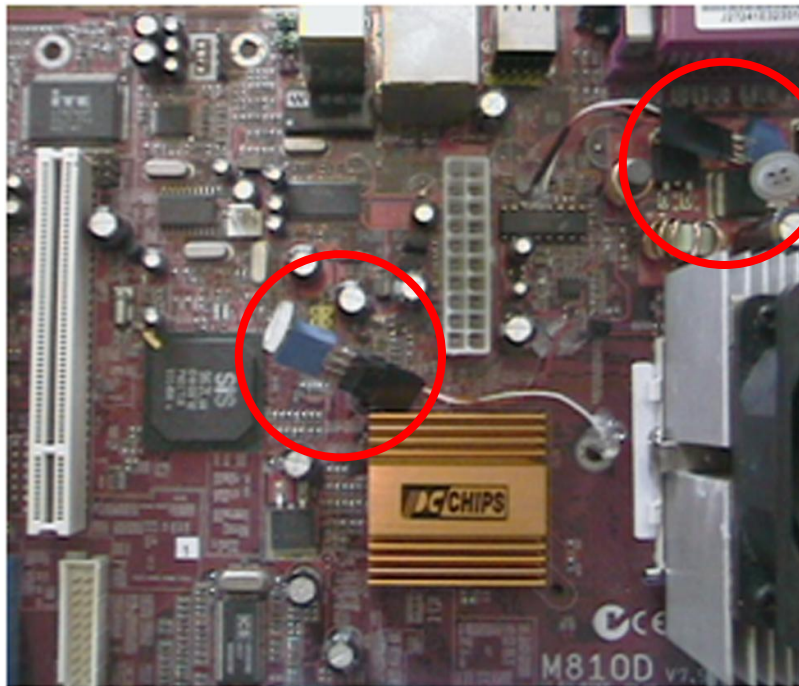
Destacado em branco temos um resistor de 750Ω ligado ao VID[4]. Em vermelho um resistor de $6K\Omega$ ligado ao VID[1]. Em laranja um resistor de $12K\Omega$ ligado ao VID[0]. Em verde um resistor de $6K\Omega$ ligado ao VID[2] e, finalmente em azul, um resistor de $1,5K\Omega$ ligado ao VID[3]. A trilha comum está destacada pela seta verde.

Quando $VID[0] = 1$, existe um caminho ligando o resistor de $12K\Omega$ com o plano terra da placa, e como todos os resistores estão em paralelo (levando em consideração apenas os resistores ativos onde $VID = 1$), podemos calcular o inverso da resistência equivalente como a soma dos inversos dos resistores, como na fórmula abaixo:

$$1/R_{eq} = \sum 1/R_n$$

Decidimos que seria mais conveniente trabalhar com um único resistor para substituir os demais. Para tanto, simplesmente cortamos as trilhas associadas logo após o resistor, e em seguida ligamos um potenciômetro de $10K\Omega$ entre a trilha comum dos resistores e o terra da placa (seta verde na imagem anterior).

Já para controlar a frequência de chaveamento, simplesmente arrancamos o resistor original de $10K\Omega$, e acrescentamos um potenciômetro de $100K\Omega$ entre os pinos 6 e 5 do controlador de chaveamento.



Em destaque, mais ao centro, o potenciômetro que regula a tensão do núcleo do processador.

Mais acima, o potenciômetro que regula a frequência de chaveamento do regulador.

Fonte: Foto ampliada das mudanças efetuadas na placa mãe PCChips M810D

3.2 Software

3.2.1 DSL (Damn Small Linux)

Para executar nossos testes, escolhemos a distribuição Linux DSL, versão 3.1, com kernel versão 2.4.26. Esta escolha foi baseada na flexibilidade e portabilidade desta versátil distribuição. Bastam 50 MB para guardar todos os arquivos do DSL e,

através do sistema MyDSL, foi possível criar pacotes que são extraídos diretamente para a memória, possibilitando a adição de nossos arquivos de teste de forma mais rápida e segura. Ao gravarmos os arquivos em uma mídia ótica (Live CD), garantimos que em caso de mau funcionamento não existe possibilidade de corromper os dados do teste.

3.2.2 O script de testes

```
#!/bin/bash
echo "tcc"

if [ "$#" -ne "3" ]
then
    exit
fi

setpci -v -H1 -s 0:0.0 6B=$(printf %x $((0x$(setpci -H1 -s 0:0.0 6B) | 0x01)));

x = 0
while [ "$x" -le "$1" ];
do
    echo "x=$x";
    echo $'\a';
    ./loadcpu;
    echo $?
    x=$((x+1));
done;

x = 0
while [ "$x" -le "$2" ];
do
    echo "x=$x";
    echo $'\a';
    ./loadcpu;
    echo $?
    usleep $3;
    x=$((x+1));
done;

exit
```

Inicialmente executamos o comando “setpci” para ativar a opção de desconexão do barramento do sistema durante os estados (C1, C2 e C3) de baixo consumo.

Para executar o script são necessários 3 parâmetros. A parte principal do script é composta por 2 laços. O primeiro parâmetro define o número de execuções do 1º laço, responsável pelo aquecimento do sistema até uma temperatura de equilíbrio que,

no caso do processador que utilizamos, foi de 45°C. Uma vez atingido este equilíbrio térmico, iniciamos o teste com pausas. O segundo parâmetro corresponde ao número de execuções do segundo laço, no qual alternamos um teste e uma pausa, cuja duração em micro segundos é passada pelo 3º parâmetro.

3.2.3 O código de teste em C

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(){
    double a, b, e, f;
    int c, d;
    a = 8;
    b = 64000000;
    e = b + 2;
    f = e + 1;
    c = 0;
    d = c;

    while (d+-100000) {
        b = 64000000 + b;
        while (b > 0.1) {
            c++;
            atan2 (b, a);
            atan2 (e, a);
            b = b/a;
            e = e/a;
            f = f/a;
        }
    }

    printf("%f ", b + e + f);

    return 0;
}
```

Compilamos o código de testes com o GCC, utilizando o nível 2 de otimização. O código consiste de 2 laços encadeados, que realizam operações matemáticas que mudam a cada nova iteração. O laço mais externo é executado até que 'd' seja 100000, quando a condição de parada é zero. A cada iteração, a diferença entre 'd' e 100000 é calculada, e a comparação é realizada, contribuindo para a complexidade das operações efetuadas. O laço mais interno é limitado pelo valor de 'b', e contém

diversas operações matemáticas que visam o aumento da demanda de energia do processador.

Os valores das variáveis foram ajustados de forma experimental, de acordo com a reação da placa conforme executávamos diferentes testes. À medida que o código se tornava mais exigente para o processador, era possível escutar alterações no som proveniente dos indutores associados ao regulador. Além disso, a temperatura era também monitorada. Assim, pudemos “calibrar” os valores destas variáveis, encontrando números suficientemente grandes para executar nossos testes.

4 Resultados e produtos obtidos

4.1 Experimentos com potenciômetros

Durante nossos testes com a placa mãe PCChips, inicialmente rodamos nosso script de testes com alterações efetuadas no potenciômetro que regula a tensão do núcleo do processador, com o potenciômetro que regula a frequência do regulador no valor nominal fixo de 100KHz. A tensão nominal de fábrica para o AMD Athlon 1333MHz é de 1750mV com tolerância de 100mV para mais ou para menos. Abaixo segue o gráfico dos resultados encontrados durante estes testes:

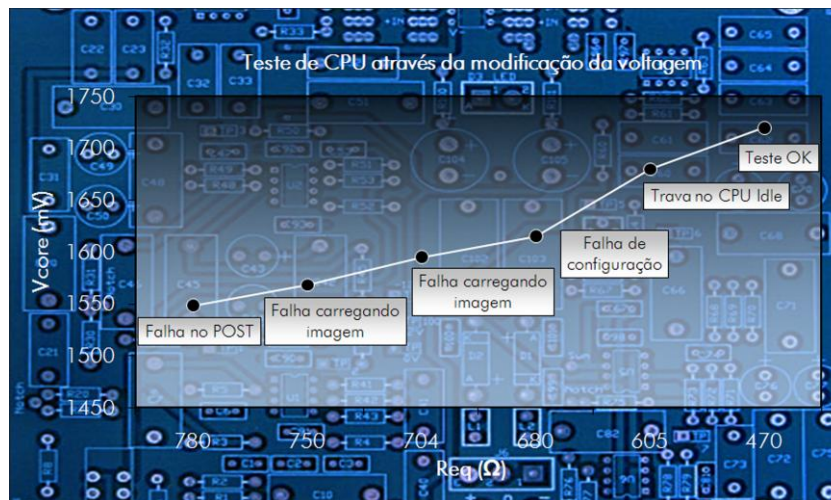


Gráfico da tensão do núcleo pela resistência, para a frequência fixa de 100KHz no regulador.

Fonte: Gráfico criado para o projeto com o Excel, com dados obtidos experimentalmente

Conforme diminuimos a resistência do potenciômetro de tensão do núcleo, obtivemos alguns resultados bastante interessantes:

- ✓ Para um valor bastante baixo, de 1548mV, ocorrem falhas no POST (Power on self test): na tentativa de inicializar os periféricos, o computador travava antes mesmo de inicializar.
- ✓ Para os valores de 1568mV e 1595mV, ocorre falha no carregamento da imagem do sistema operacional para a memória.
- ✓ Em 1615mV, ocorrem falhas de configuração, relativas ao processo de configurações do sistema, após a cópia da imagem do mesmo para a memória.
- ✓ Já no valor de 1680mV, o sistema é inicializado com sucesso, mas trava por completo quando é efetuada uma tentativa de configuração do chipset para economia de energia.
- ✓ Finalmente, para o valor de 1720mV, não ocorrem travamentos, mesmo com a execução dos nossos testes.

Desta forma, para a frequência do regulador fixada em 100KHz (valor nominal), encontramos, após diversas tentativas, o valor mínimo de 1720mV de tensão para o qual o processador atua de forma satisfatória.

Após o carregamento do sistema operacional, decidimos então monitorar a tensão mínima de operação tal que os testes ainda pudessem ser executados. Mesmo com poucas tentativas, observamos que o valor mínimo de operação não varia muito, permanecendo na faixa de 1560 a 1570mV.

Em seguida, passamos a executar nossos testes com o potenciômetro da tensão do núcleo fixado no valor nominal de 1750mV, mas agora variando o potenciômetro que controla a frequência do regulador. Por questões de segurança (para não perdermos nosso processador nos testes), mantivemos os valores num limite estreito entre 80 e 110KHz.

A seguir temos o gráfico dos resultados dos testes, no qual se pode notar que, mesmo alterando a freqüência de forma significativa (porém segura), o fato do processador estar na tensão nominal de 1750mV, é suficiente para que todos os testes sejam bem sucedidos.

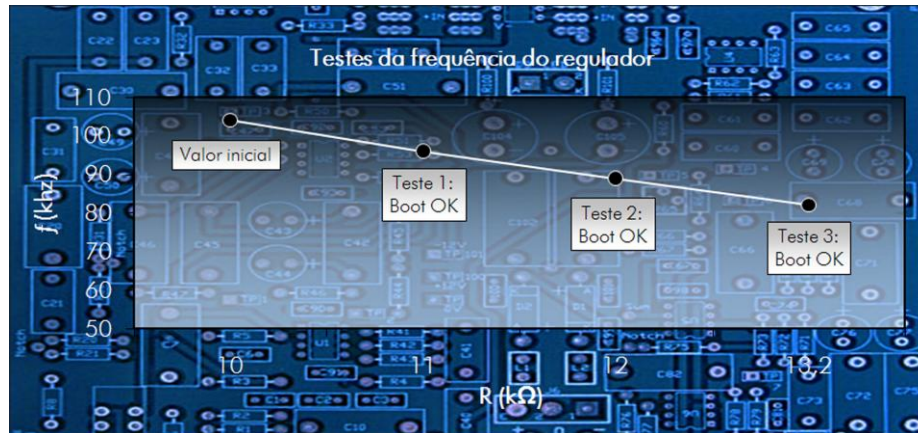


Gráfico da freqüência do regulador pela resistência, para a tensão fixa de 1750mV no processador.

Fonte: Gráfico criado para o projeto com o Excel, com dados obtidos experimentalmente

4.2 Hipótese: Interferência eletromagnética

Ao longo dos testes que realizamos, a placa-mãe PCChips M810D não apresentou quaisquer erros de segmentation fault. Apenas conseguimos provocar erros através da alteração manual de tensão do processador e freqüência do regulador, conforme visto no tópico anterior. Já a placa-mãe Asus A7V8X-X, freqüentemente apresentava erros deste tipo, quando executávamos nossos testes.

Isto nos fez levantar a Hipótese de que pode existir uma relação entre os designs das placas e a ocorrência de erros. No caso da placa-mãe PCChips, o caminho percorrido pela corrente elétrica a partir do conector de energia, é bem mais curto e direto se comparado ao modelo da Asus. Nesta, a trilha percorre uma trajetória repleta de componentes sensíveis, como as memórias, o gerador de clock, chipset da placa-mãe e o próprio processador, o que pode ocasionar uma forte interferência

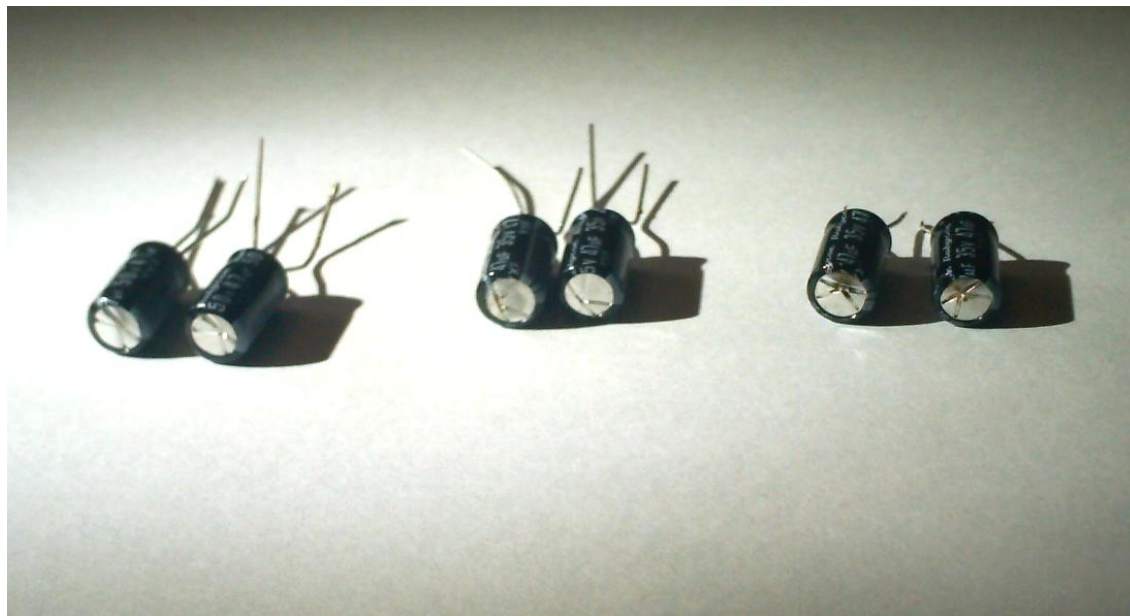
eletromagnética na placa, eventualmente causando segmentation faults. Nos modelos mais atuais, existem conectores dedicados de energia para o processador, diminuindo esta distância e conseqüente ocorrência de interferências significativas.

4.3 Identificação de erros por capacitores danificados

O sucesso e eficiência de nossos testes puderam ser constatados quando, ao trocar os capacitores antigos e desgastados por novos e de boa qualidade, a ocorrência destes erros na placa-mãe Asus deixou de ser observada. Assim, as interferências entre os componentes da placa, tal qual levantamos em nossa Hipótese no item anterior, teriam sido atenuadas graças aos novos filtros de entrada do regulador.

4.4 Detecção do fim da vida útil de capacitores eletrolíticos

Nas páginas do nosso trabalho (<http://linux.ime.usp.br/~rhaiger/mac499/> e <http://linux.ime.usp.br/~monta/mac499/>), é possível encontrar um vídeo no qual exibimos uma alteração proposital que efetuamos na placa-mãe PCChips M810D, trocando os capacitores do regulador por outros de menor capacitância ($47 \mu\text{F}$) e, além disto, executando nosso script de testes sistematicamente. O resultado é uma aceleração bastante acentuada do desgaste sofrido por capacitores eletrolíticos na sua vida útil. Na prática, é possível ouvir e ver uma falha por ventilação bastante crítica, que consiste no vazamento do fluido eletrolítico pela parte superior do capacitor, acompanhada de uma forte emissão de vapor e odor característico.



Na foto acima estão os capacitores destruídos durante o teste citado anteriormente.

Fonte: Foto dos capacitores danificados experimentalmente no projeto

Após cerca de 180 execuções do código de testes, dava-se início o processo de ventilação descrito acima. Na tabela a seguir, temos alguns destes resultados:

Número de execuções	Orientação da placa
186	Horizontal
213	Horizontal
482	Vertical

Fonte: Dados obtidos experimentalmente em testes efetuados para o projeto

5 Conclusões e futuro do projeto

Através dos testes efetuados na placa-mãe PCChips M810D, verificamos que é possível simular a diminuição da vida útil dos componentes da placa, utilizando apenas a execução do script de teste.

Executando nosso script de forma controlada, é possível identificar capacitores desgastados, subdimensionados ou perto do fim de sua vida útil, acompanhando o excessivo aumento de sua temperatura, ao longo da execução da nossa rotina de testes. Esta variação de temperatura (antes e imediatamente após os testes) chegou a 43°C.

Embora tenhamos ficado satisfeitos com os resultados obtidos, notamos que parece existir uma forte influência do design das placas-mãe na ocorrência de determinados erros em outras regiões de suas constituições.

No futuro, pretendemos combinar outras metodologias de testes com o intuito de aprimorá-las, tendo como objetivo aumentar a confiabilidade dos sistemas computacionais. Nosso próximo objetivo é focar nossas análises nas interações entre a CPU e a memória RAM.

6 Referências bibliográficas

1. *AMD Athlon™ Processor Voltage Regulation Design*, AMD - 2008. Disponível em: http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/22651.pdf. Acessado em janeiro de 2007.
2. *Intel Pentium 4 Processor VR-Down*, Intel - 2008. Disponível em: <http://download.intel.com/design/Pentium4/guides/24989104.pdf>. Acessado em 2008.
3. *Two Slice Interleaved Synchronous Buck Converter*, Fairchild Semi. Disponível em: <http://www.alldatasheet.com/datasheet-pdf/pdf/51180/FAIRCHILD/FAN5091.html>. Acessado em fevereiro de 2008.
4. *Life of Aluminum Electrolytic Capacitors*, Rubycon Corp. Disponível em: <http://www.rubycon.co.jp/en/products/alumi/pdf/Life.pdf>. Acessado em 2008 e recuperado em junho de 2009.
5. SMITH, Steven W. *The Scientist and Engineer's Guide to Digital Signal Processing*: California Technical Pub, 1997. Disponível em: <http://www.dspguide.com/pdfbook.htm>. Acessado em 2009.
6. *Advanced Configuration and Power Interface Specification*, Hewlett-Packard Corporation, Intel Corporation, Microsoft Corporation, Phoenix Technologies Ltd., Toshiba Corporation. Disponível em: <http://www.acpi.info/DOWNLOADS/ACPIspec40.pdf>. Acessado em 2008 e recuperado em novembro de 2009.
7. *The Intel® Architecture Embedded Design Center Tool Loaner Program*, Intel – 2009. Disponível em: <http://edc.intel.com/Platforms/Tool-Loaner-Program/>. Acessado em 2009.
8. *Bash Reference Manual*, GNU.org – 2009. Disponível em: <http://www.gnu.org/software/bash/manual/bashref.html>. Acessado em 2007 e recuperado em 2009.
9. *Less Watts – Saving Power with Linux* – 2009. Disponível em: <http://www.lesswatts.org/projects/>. Acessado em 2008.