

Microarquiteturas de Baixo Consumo

Leonardo Silva Vidal

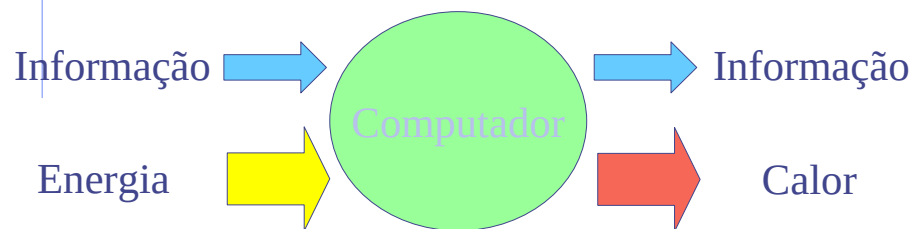
Doutorado NCE/IM - UFRJ

Microarquiteturas de Baixo Consumo

- ◆ Introdução
- ◆ Revisão de CMOS
- ◆ Consumo de Potência e Energia
- ◆ Redução do Consumo de Energia
- ◆ Estudo de Caso
- ◆ Simuladores Arquiteturais
- ◆ Bibliografia

Introdução

Informação x Energia x Calor



Informação x Energia x Calor

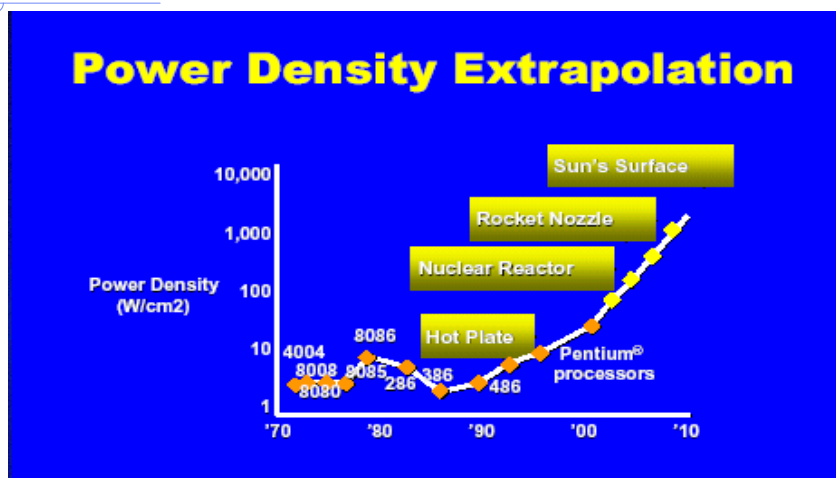
- ◆ A maior parte da energia fornecida é transformada em calor
- ◆ O consumo de energia e a dissipação térmica crescem exponencialmente com o crescimento da produção de informação
- ◆ O controle da dissipação térmica está mais complexo e difícil

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

5

Dissipação Térmica



29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

6

Vantagens de Reduzir o Consumo

◆ Aumenta a vida das baterias

- Celulares, PDAs, Robótica, Sistemas Embutidos, Implantes, etc.

◆ Reduz a dissipação térmica

- Estações de Trabalho, Servidores, Servidores em Lâmina, etc.
- Reduz a necessidade de resfriamento
- Aumenta a confiabilidade

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

7

Vantagens de Reduzir o Consumo

◆ Reduz a necessidade de resfriamento

- Encapsulamento mais simples e barato
- Reduz o volume e o peso
- Reduz o gasto de energia para resfriamento
- Reduz a poluição sonora

◆ Menor impacto ambiental

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

8

Revisão de CMOS

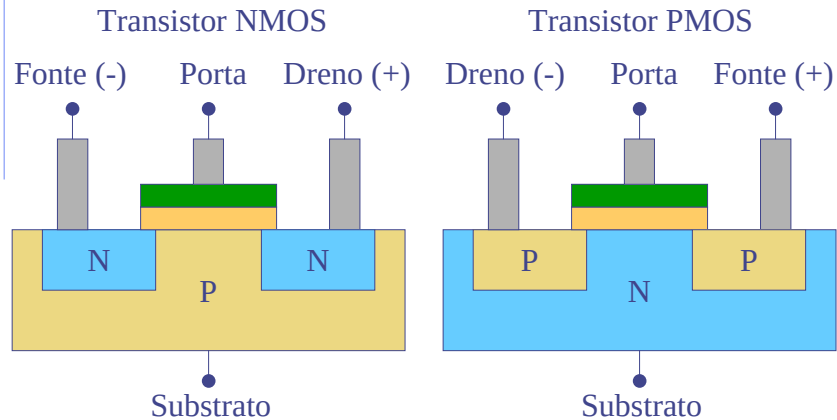
Tecnologia MOS

◆ MOS – Metal Oxide Semiconductor

- Metal – Alumínio ou Cobre
- Isolante – Óxido de Silício
- Semicondutor – Silício

◆ Permite a construção de diodos, transistores, capacitores e outros

Transistor MOS-FET



29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

11

Transistor MOS-FET

◆ FET – Field Effect Transistor

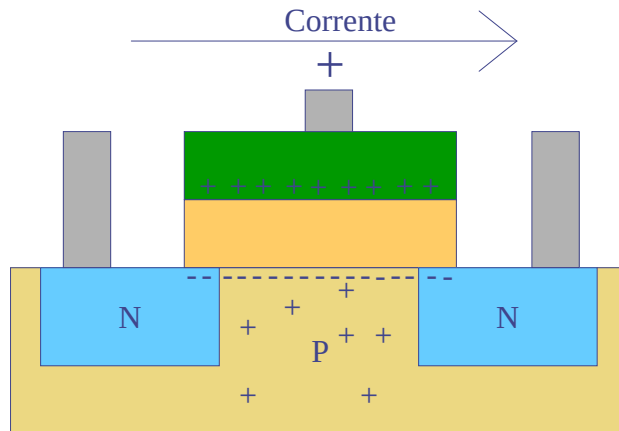
◆ Efeito de Campo → A aplicação de corrente na porta cria um canal condutivo que transfere portadores entre a fonte e o dreno por indução eletromagnética

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

12

Efeito de Campo



29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

13

Transistor MOS-FET

- ◆ Transistor NMOS → Passagem de elétrons quando a voltagem da porta é maior que a voltagem do substrato
- ◆ Transistor PMOS → Passagem de buracos quando a voltagem da porta é menor que a voltagem do substrato

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

14

Tecnologia CMOS

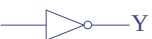
- ◆ CMOS – Complementary MOS
- ◆ Emprega pares de transistores NMOS e PMOS de forma complementar
- ◆ Permite construir portas lógicas (NAND, NOR, XOR, somadores e outros)

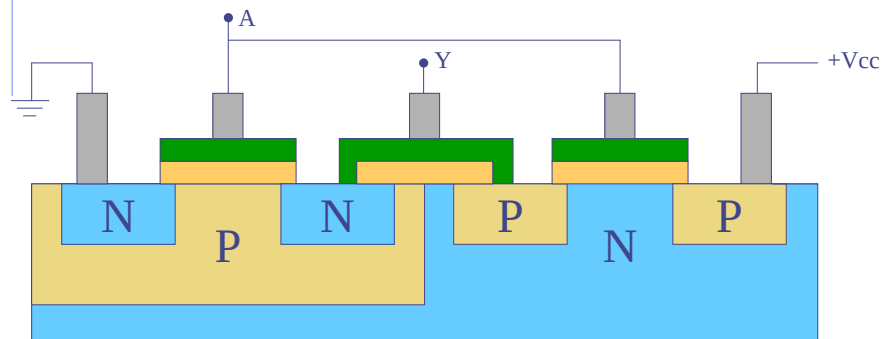
29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

15

Exemplo de CMOS

- ◆ Inversor CMOS : $Y = \text{NOT } A$ 

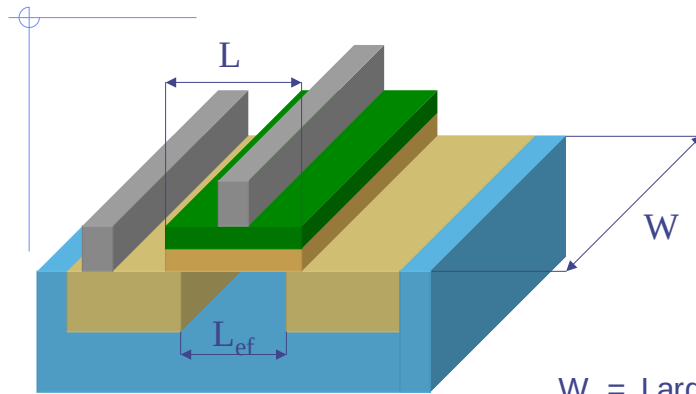


29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

16

Estrutura de um MOS-FET



W = Largura do Canal
 L = Comprimento do Canal
 L_{ef} = Comprimento Efetivo do Canal

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

17

Transistor MOS

- ◆ A porta é um capacitor de placa com área $W \times L$
- ◆ As junções PN são capacitores de interface
- ◆ A mudança de estado de um transistor envolve a carga ou descarga de todos os seus capacitores

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

18

Transistor MOS

- ◆ Grande parte da energia consumida na mudança de estado é utilizada na carga dos capacitores
- ◆ O tempo para um transistor mudar de estado depende do tempo para carregar ou descarregar todos os seus capacitores
 - A duração do ciclo do processador depende do tempo para a mudança de estado

Transistor MOS

- ◆ O transistor MOS está sujeito a diversos tipos de vazamentos de corrente
 - Entre a fonte e o dreno
 - Entre a porta e o substrato
 - Nas junções PN
 - Entre os transistores adjacentes

Transistor MOS

- ◆ O canal apresenta resistência
 - Diminui com o aumento da largura do canal
 - Diminui com a redução do comprimento do canal
- ◆ O vazamento de corrente entre a fonte e o dreno aumenta com a redução do comprimento do canal

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

21

Transistor MOS Ideal

- ◆ Um transistor ideal só deveria gastar energia durante a mudança de estado na carga ou descarga da porta

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

22

Transistor MOS Real

- ◆ Também gasta energia durante a mudança de estado para carregar ou descarregar os capacitores das junções
- ◆ Gasta energia mesmo durante o equilíbrio devido aos vazamentos de corrente

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

23

Consumo de Potência e Energia

Potência Consumida

◆ $P = P_{\text{dyn}} + P_{\text{short}} + P_{\text{leak}}$

◆ Potência Dinâmica

- P_{dyn} – Potência gasta na carga das portas dos transistores
- P_{short} – Corrente de curto-circuito na mudança de estado dos transistores interligados

◆ Potência Estática

- P_{leak} – Vazamentos de corrente

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

25

Potência Estática

◆ Vazamento de corrente

- O principal vazamento ocorre entre a fonte e o dreno
- Os outros vazamentos são muito menores

◆ $P_{\text{leak}} = I_{\text{leak}} V$

- I_{leak} – intensidade da corrente que vaza
- V – voltagem da fonte

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

26

Potência Estática

◆ Vazamento de corrente

- Cresce exponencialmente com o decréscimo da voltagem de limiar
- A voltagem de limiar é proporcional à voltagem da fonte
- Atualmente, corresponde a menos de 10% do consumo total
- Sua importância está crescendo com a redução da voltagem do processador

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

27

Potência Dinâmica

◆ Corrente de curto-circuito

- Ocorre durante as mudanças de estado de transistores interconectados

◆ $P_{\text{short}} = t A I_{\text{short}} V$

- t – fração do ciclo onde ocorre o curto-circuito
- A – média de transições por ciclo
- I_{short} – corrente do curto-circuito

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

28

Potência Dinâmica

◆ Carga da porta de um transistor

- $P = dE/dt = E/\text{período} = E \cdot f$
- $dE/dt = V \cdot i = V \cdot dQ/dt = C_t \cdot V \cdot dV/dt$

◆ $E = \frac{1}{2} C_t \cdot V^2$

◆ $P = \frac{1}{2} C_t \cdot V^2 \cdot f$

- f – frequência
- C_t – capacitância da porta

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

29

Potência Dinâmica

◆ Potência do processador

- O principal componente é a potência necessária para carregar os transistores
- A potência consumida devido a resistividade dos condutores é muito menor

◆ $P_{\text{dyn}} = \frac{1}{2} A \cdot C \cdot V^2 \cdot f$

- C – capacitância total
- f – frequência

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

30

Energia Dinâmica

◆ Energia do processador ideal

- $E = P \cdot t = \frac{1}{2} A \cdot C \cdot V^2 \cdot f$
- $E = \frac{1}{2} A \cdot C \cdot V^2 \cdot f \cdot (IC \cdot CPI \cdot 1/f)$
- $E = \frac{1}{2} (\text{transições/ciclo}) \cdot (n^\circ \text{ de ciclos}) \cdot C \cdot V^2$
- **$E = \frac{1}{2} (\text{total de transições}) \cdot C \cdot V^2$**

◆ Não depende do tempo e da frequência

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

31

Energia Dinâmica

◆ Energia do processador real

- Também depende da frequência devido à resistência e aos vazamentos
- Quanto maior a potência, maior a energia consumida

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

32

Redução do Consumo de Energia

Redução do Consumo

- ◆ A frequência máxima depende da voltagem
 - $f_{\max} \sim (V - V_{\text{limiar}})^2 / V$
- ◆ A voltagem mínima e a frequência máxima dependem da capacitância
 - Tempo e energia para carregar os capacitores
- ◆ O desempenho depende da frequência e do número de transições

Redução do Consumo

$$P = \frac{1}{2} A \cdot C \cdot V^2 \cdot f + t A I_{\text{short}} V + I_{\text{leak}} V$$

◆ Opções para reduzir o consumo dinâmico

- C
- A
- V e f

◆ Opções para reduzir o consumo estático

- I_{leak}

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

35

Técnicas para Reduzir o Consumo

◆ Novos Processos de Produção

- Transistores menores
- Transistores com novas geometrias e composições

◆ Técnicas de Hardware

- Reduzir a voltagem e a frequência
- Reduzir o número de transições

◆ Técnicas de Software

- Compilador

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

36

Técnicas para Reduzir o Consumo

- ◆ As técnicas de hardware com grande latência precisam de apoio do software (Sistema Operacional e/ou Compilador)
 - O hardware provê o mecanismo
 - O software provê a política
 - Simplifica o hardware
 - Políticas mais abrangentes e eficientes

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

37

Transistores Menores

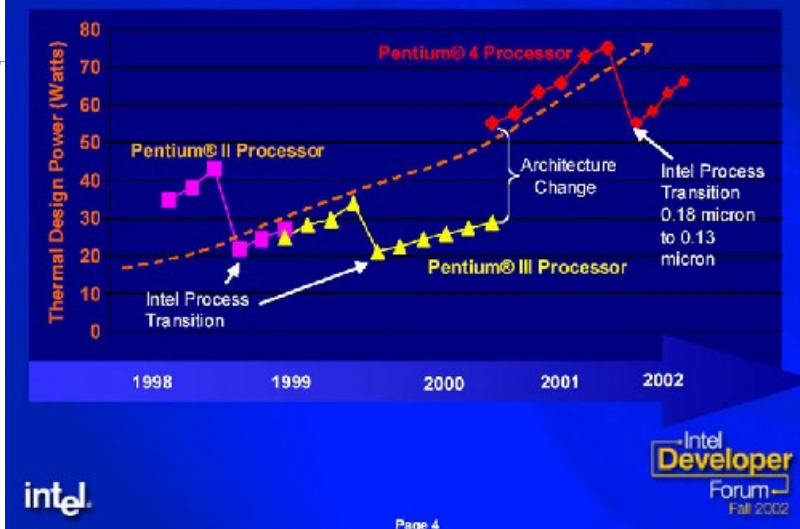
- ◆ Portas mais curtas e estreitas
 - Reduz C e V
 - Aumenta I_{leak}
- ◆ A redução da capacitância da porta é proporcional a $W \times L$
 - A redução da capacitância total do transistor é no melhor caso linear

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

38

Desktop Processor Power Trends

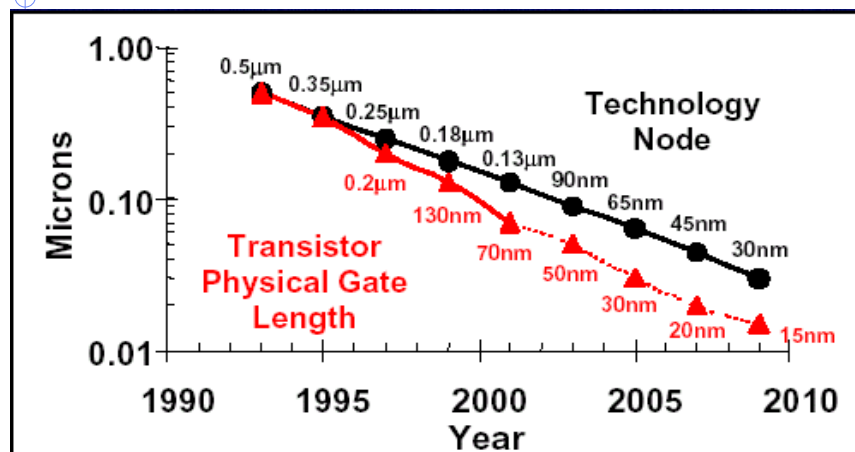


29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

39

Processos Intel



29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

40

Processos de Fabricação da Intel

◆ P858

- $W = 0,18 \mu\text{m}$ e $L = 0,13 \mu\text{m}$

◆ P860 e P1260

- $W = 0,13 \mu\text{m}$ e $L = 0,07 \mu\text{m}$

◆ P1262

- $W = 0,09 \mu\text{m}$ e $L = 0,05 \mu\text{m}$

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

41

Tecnologia SOI

◆ SOI – Silicon on Insulator

◆ Envolve os semicondutores com isolante

- Reduz a área das junções PN
- Menor C , V e I_{leak}
- Reduz a fuga de corrente para os transistores vizinhos

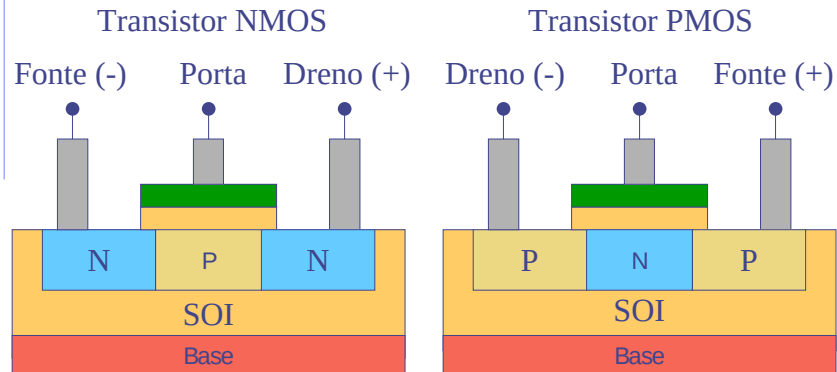
◆ Empregada pela AMD (Hammer) e IBM

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

42

Transistor MOS-FET SOI



29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

43

Técnicas de Hardware

- ◆ Redução da voltagem e da frequência
 - Dynamic Voltage Scaling
- ◆ Redução do número de transições
 - Clock Gating
 - Power Gating
 - Fetch Gating
 - Dynamic Pipelining Scalling

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

44

Dynamic Voltage Scaling

- ◆ Regula a voltagem e a frequência dinamicamente
 - A frequência escolhida tende a ser o máximo permitido pela voltagem escolhida a fim de não prejudicar muito o desempenho
- ◆ Reduz a potência ao cubo e a energia ao quadrado

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

45

Dynamic Voltage Scaling

- ◆ Executar um período inteiro em baixa voltagem é mais econômico do que executar em voltagens maiores por períodos curtos e permanecer inativo pelo resto do período
- ◆ A alteração da voltagem e da frequência envolve grandes latências

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

46

Dynamic Voltage Scaling

- ◆ Seu emprego eficiente depende de um planejamento com muita antecedência
 - Pode ser muito caro ou complexo para o hardware
 - O planejamento deve ser feito por software pelo sistema operacional, pelo compilador ou pela combinação dos dois
 - O hardware disponibiliza interfaces para definir a voltagem e a frequência

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

47

Dynamic Voltage Scaling

- ◆ A voltagem pode assumir apenas valores discretos dentro de um intervalo
 - Limita as escolhas do software
 - As voltagens e as frequências disponíveis raramente corresponderão aos valores ideais para executar uma determinada aplicação
 - Desperdiça de energia

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

48

Dynamic Voltage Scaling

◆ Aplicações comerciais para computadores móveis

- Transmeta LongRun
- AMD PowerNow
- Intel SpeedStep

Dynamic Voltage Scaling

◆ Políticas de software para a aplicação do Dynamic Voltage Scaling

- Algoritmos Baseados em Prazos
- Algoritmos Baseados em Intervalos

Algoritmos Baseados em Prazos

- ◆ Se a necessidade de processamento da aplicação é conhecida, atribui a voltagem e a frequência que garantem o término do processamento no momento em que o resultado é esperado
 - Apropriados para aplicações de tempo real e multimídia
 - Podem ser empregados pelo sistema operacional ou explicitamente pelo compilador

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

51

Algoritmos Baseados em Intervalos

- ◆ Aplicados quando não é possível prever o processamento da aplicação
 - Utilizados pelo sistema operacional
- ◆ Dividem o tempo de processamento em intervalos iguais
- ◆ Dependem da previsão da carga de trabalho para o próximo intervalo e de uma política para definir a frequência escolhida

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

52

Algoritmos Baseados em Intervalos

◆ Algoritmos para prever a utilização do próximo intervalo

- Flat-u – sempre prevê que será u , onde u é uma constante menor que u_m
- PAST – igual ao anterior
- Aged-a – média ponderada por a^k da carga dos intervalos anteriores, onde a é menor que u_m e k é a idade do intervalo

Algoritmos Baseados em Intervalos

◆ Algoritmos para prever a utilização do próximo intervalo

- LongShort – média ponderada dos doze últimos intervalos, sendo que os três mais recentes têm peso três e os demais têm peso um

Algoritmos Baseados em Intervalos

- ◆ Algoritmos para calcular a frequência para o próximo intervalo
 - Weiser – se a previsão x é maior que 70% aumenta a frequência em 20%; se for menor que 50%, reduza a frequência por $(60-x)\%$
 - Peg – se a previsão é maior que 98% aumente a frequência para o máximo; se for menor que 93%, reduza para o mínimo
 - Chan – multiplique a previsão pela frequência máxima

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

55

Algoritmos Baseados em Intervalos

- ◆ As melhores associações são
 - Past/Weiser
 - LongShort/Chan
 - Flat/Chan
 - Past/Peg

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

56

Clock Gating

- ◆ Suprime o sinal de relógio para algumas partes do processador
 - Reduz o número de transições
 - Latência mínima ou inexistente
- ◆ Desliga partes do processador
 - Reduz o número de transições e o consumo estático
 - Latência considerável

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

57

Clock Gating

- ◆ Suprimir o ramo da árvore de distribuição do sinal do relógio que alimenta uma unidade funcional
 - Pode ser controlado pelo hardware
 - Complexo em pipelines super escalares profundos
 - Sem latência

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

58

Clock Gating

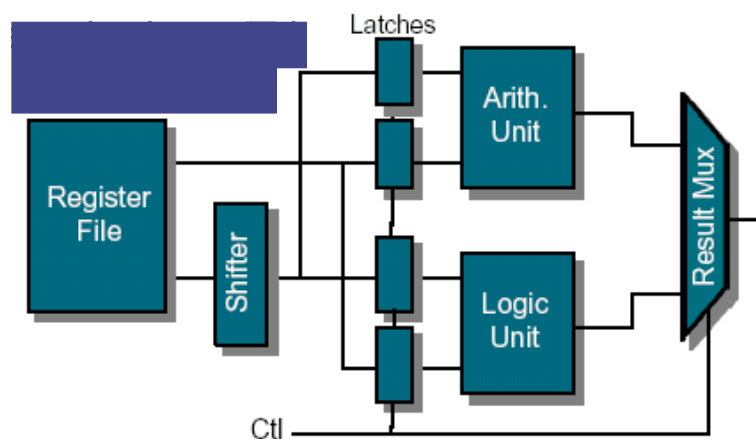
- ◆ Suprimir transições de estados desnecessárias para uma unidade funcional ou partes dela
 - Pode ser controlado pelo hardware
 - Sem latência

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

59

Clock Gating



29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

60

Power Gating

◆ Desliga a alimentação da unidade

- Unidades de ponto flutuante
- Unidades para operações complexas com inteiros
- Módulos de SOCs
- Latência não desprezível
- Precisa de suporte do software

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

61

Clock Gating

◆ Reduzir o número de entradas ativas em tabelas

- Suprime o sinal de relógio ou a alimentação para linhas dos caches, TLBs, BHTs, PHTs e BTBs
- Reduz o desempenho devido aos sleep misses
- O algoritmo de predição de desvio deve suportar tabelas com um número variável de entradas

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

62

Clock Gating

◆ Reduzir o número de entradas ativas em tabelas

- Escolher o momento ideal para desligar as linhas é muito complexo
- Reduzir as entradas ativas prematuramente aumenta os sleep misses
- Demorar para reduzir as entradas ativas provoca a perda de oportunidades para economizar energia

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

63

Clock Gating

◆ Reduzir o número de entradas ativas em tabelas

- Só deve ser empregado quando há certeza de que o recurso não será utilizado no futuro próximo
- Recarregar as entradas de caches e tabelas desligadas envolve uma grande latência associada a um pico de consumo
- Precisa de apoio do compilador e/ou do sistema operacional

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

64

Fetch Gating

- ◆ Erros de predição implicam no desperdício de energia com a busca, decodificação e execução de instruções que serão descartadas
- ◆ Apenas as instruções de um desvio predito com alto grau de certeza são buscadas e executadas

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

65

Fetch Gating

- ◆ Depende de uma medida de confiança para cada predição
- ◆ Permite grande economia de energia com pequena perda de desempenho
- ◆ Sem latência

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

66

Fetch Gating

- ◆ Confiante e correto – não prejudica o desempenho, mas não economiza energia
- ◆ Confiante e incorreto – desperdiça energia
- ◆ Não confiante e correto – perda de desempenho
- ◆ Não confiante e incorreto – economiza energia sem prejudicar o desempenho

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

67

Dynamic Pipelining Scalling

- ◆ Une estágios adjacentes da pipeline
- ◆ Um pipeline profundo desperdiça mais energia em função das dependências e das penalidades dos erros dos caches e da predição de desvios

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

68

Dynamic Pipelining Scalling

- ◆ Permite frequências que são frações da frequência máxima
- ◆ Suprime alguns caminhos de adiantamento de dados
 - Economiza energia
- ◆ A implementação da pipeline de tamanho variável é mais complexa

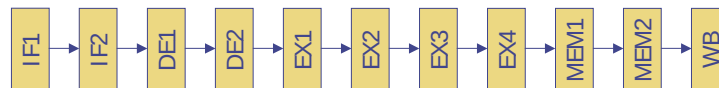
29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

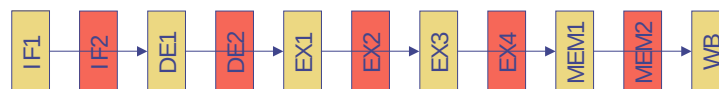
69

Dynamic Pipelining Scalling

Alta Frequência



Baixa Frequência



29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

70

Energy Aware Software

- ◆ O sistema operacional ou a própria aplicação fazem uma estimativa da possibilidade de executar uma aplicação dentro de certos níveis de desempenho em função dos requerimentos de energia da aplicação, da carga da bateria e da entrada de dados do usuário

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

71

Estudo de Caso Processador DEC SA-110

Estudo de Caso

Processador DEC SA-110

- ◆ Família StrongARM
- ◆ Conjunto de instruções ARMv4
- ◆ Voltagem variável de 1,5 a 2,2V
- ◆ Consumo de 300mW a 100MHz e 1,65V
- ◆ Consumo de 900mW a 200MHz e 2,0V

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

73

Estudo de Caso

Processador DEC SA-110

- ◆ 2,5 milhões de transistores
- ◆ 90% dos transistores nos caches de 16kB
- ◆ Processo de 0,35 μ m com três camadas de metal
- ◆ Comprimento efetivo do canal 0,25 μ m
- ◆ Voltagem de limiar 0,35V

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

74

Estudo de Caso

Processador DEC SA-110

- ◆ Derivado do Alpha 21064
 - Processo de 0,75μm
 - 26W a 200MHz e 3,45V
- ◆ Sem unidade de ponto flutuante
- ◆ Sem predição de desvio

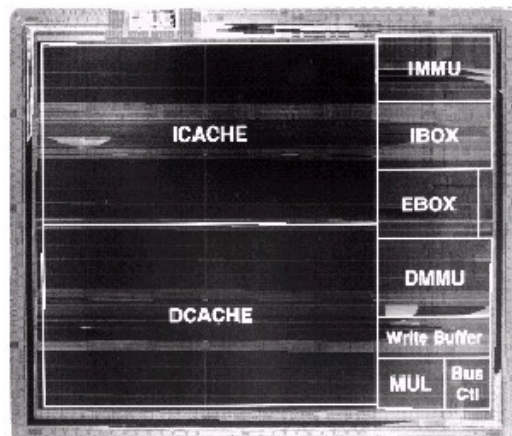
29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

75

Estudo de Caso

Processador DEC SA-110



29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

76

Estudo de Caso

Processador DEC SA-110

- ◆ IBOX – unidade de busca e decodificação
- ◆ EBOX – unidade de execução de inteiros
- ◆ MUL – Unidade de multiplicação e acúmulo
- ◆ DMMU e IMMU – unidades de gerência de memória para dados e instruções
- ◆ WB – buffer de escrita
- ◆ BIU – unidade de interface com o barramento
- ◆ PLL – phase locked loop
- ◆ Dcache e Icache – caches de dados e instruções

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

77

Estudo de Caso

Processador DEC SA-110

- ◆ Pipeline de cinco estágios
 - Busca, envio, execução, buffer/memória e escrita
- ◆ Adiantamento de dados
 - Entre a saída e a entrada da ALU
 - Entre o cache de dados e a entrada da ALU
- ◆ Bolhas de um ciclo para dependências de loads

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

78

Estudo de Caso

Processador DEC SA-110

◆ Pipeline

- Desvios condicionais resolvidos no estágio de envio
- Somador no IBOX para gerar o endereço alvo
- Atraso de um ciclo para a geração do endereço alvo

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

79

Estudo de Caso

Processador DEC SA-110

◆ Banco de registradores

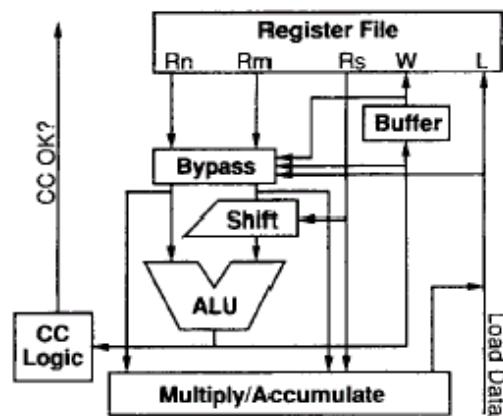
- 31 entradas
- Três portas de leitura e duas portas de escrita
- 15 registradores de uso geral
- PC
- 15 registradores escondidos para tratamento de exceções

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

80

Estudo de Caso Processador DEC SA-110



29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

81

Estudo de Caso Processador DEC SA-110

- ◆ Unidade de multiplicação e acumulo
 - Resultados de 32 ou 64 bits
 - Multiplica 12 bits por ciclo
 - Somador final para o resultado
 - Dois a quatro ciclos (32 bits)
 - Três a cinco ciclos (64bits)
 - Seus circuitos são desligados quando não está em uso (Clock gating)

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

82

Estudo de Caso

Processador DEC SA-110

◆ Caches

- Caches separados de 16kB
- Associativos com 32 conjuntos e Blocos de 32 bytes
- 16 bancos totalmente associativos
- Dcache writeback no write allocation
- Guarda o endereço físico
- Dirty bit para cada meio bloco
- Acesso em um ciclo

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

83

Estudo de Caso

Processador DEC SA-110

◆ Gerência de memória

- TLB com 32 entradas
- Páginas de 4kB, 64kB ou 1MB
- Buffer de escrita com 8 entradas de 16 bytes
- Escritas no mesmo bloco são unidas
- Loads tem maior prioridade
- A prioridade dos stores é maior se há um load em uma posição que está no buffer de escrita

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

84

Estudo de Caso

Processador DEC SA-110

◆ Modos de baixo consumo

- Modo Sleep e Modo Idle

◆ Modo Sleep

- Longos períodos
- O processador é desligado
- 140 μ s para retornar
- Mantém o estado do barramento

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

85

Estudo de Caso

Processador DEC SA-110

◆ Modo Idle

- Períodos curtos
- PLL continua ativo
- Os sinais de relógio do processador e do barramento não são distribuídos
- Retorno imediato
- Consumo de 20 mW (18 mW devido aos vazamentos)

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

86

Estudo de Caso Processador DEC SA-1100

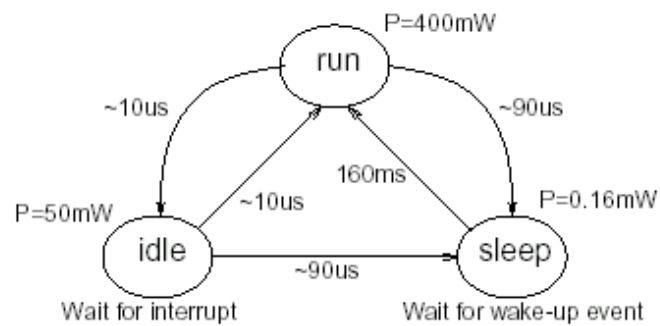


Fig. 24. Power state machine for the StrongARM SA-1100 processor.

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

87

Simuladores Arquiteturais

Simuladores Arquiteturais

- ◆ SimplePower
- ◆ PowerAnalyzer

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

89

SimplePower

- ◆ Baseado no SimpleScalar
- ◆ Pipeline de cinco estágios
- ◆ Trabalha apenas com as instruções inteiras
- ◆ Consulta o consumo da memória, do barramento e das unidades funcionais em tabelas

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

90

SimplePower

- ◆ Não mede o consumo da unidade de controle e da geração e distribuição do sinal de relógio
- ◆ A cada ciclo calcula o consumo das instruções, da memória, do barramento e das unidades funcionais ativas
- ◆ É sensível às transições
- ◆ O tamanho das unidades é alterado fornecendo novas tabelas

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

91

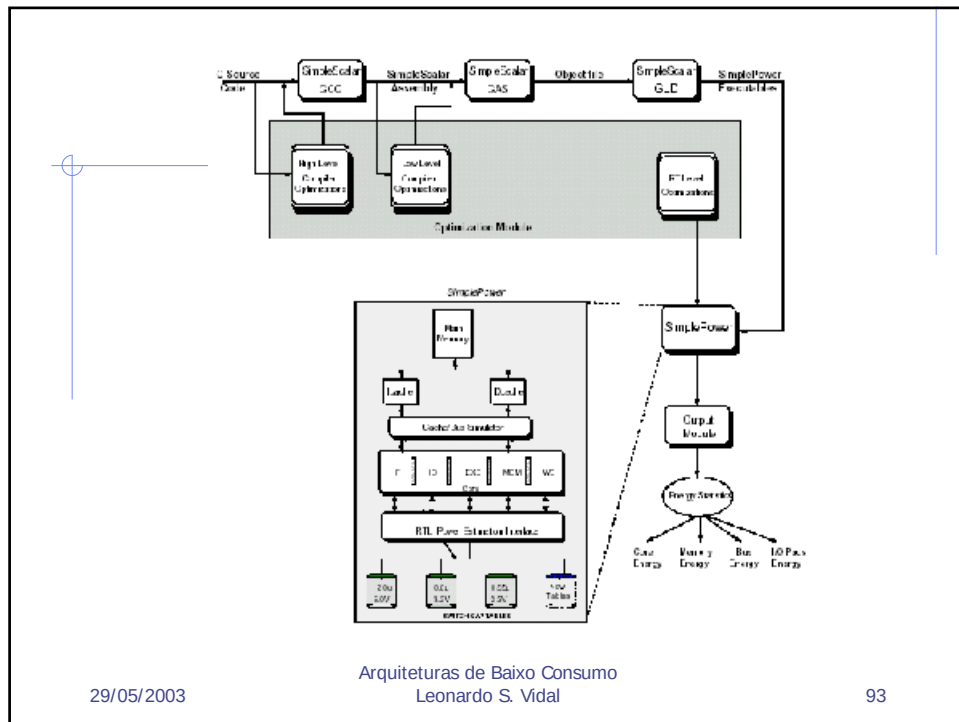
SimplePower

- ◆ O tamanho das unidades é alterado fornecendo novas tabelas
- ◆ Precisão de 15%

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

92



PowerAnalyzer

- ◆ Modela um processador ARM no SimpleScalar
- ◆ Permite a modelagem de dispositivos embutidos
- ◆ Permite a simulação da aplicação, do sistema operacional e dos drivers de dispositivos
- ◆ Inclui preditores de desvio, BTB e TLB
- ◆ Executa as instruções em ordem

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

94

Bibliografia

Bibliografia

- ◆ L. Benini e G De Michelit, "System-Level Power Optimization: Techniques and Tools", In ACM Transactions on Design Automation of Electronic Systems, V5, N2, 2000.
- ◆ D. Brooks e M. Martonosi, "Dynamically Exploiting Narrow Width Operands to Improve Processor Power and Performance", In HPCA-5, January 1999.
- ◆ T. Pering, T. Burd e R. Brodersen, "The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms", In IEEE Symposium on Low Power Electronics, 1995.

Bibliografia

- ◆ A. Sinha, A. P. Chandrakasan, "Energy Aware Software", in Proceedings of the 13th International Conference on VLSI Design, January 2000.
- ◆ J. Montanaro et al., "A 160MHz, 32b, 0.5W CMOS RISC Microprocessor", IEEE Journal of Solid-State Circuits, volume 31, number 11, November 1996, pp. 1703-1714.
- ◆ R. Gonzalez and M. Horowitz, "Energy Dissipation in General-Purpose Microprocessors," IEEE J. Solid-State Circuits, IEEE Press, Piscataway, N.J., 1996, pp. 1277-1284.

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

97

Bibliografia

- ◆ S. Segars, "Low Power Design Techniques for Microprocessors", ISSCC, Feb 4th 2001.
- ◆ A. Klaiber, "The Technology Behind Crusoe™ Processors", Transmeta Corporation, January 2000.
- ◆ G. Marcyk e R. Chau, "New Transistors for 2005 and Beyond", Intel Corporation.
- ◆ J. R. Lorch e A. J. Smith, "Improving Dynamic Voltage Scaling Algorithms with PACE", In Sigmetrics 2001, Cambridge, MA, June 2001.

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

98

Bibliografia

- ◆ T. Pering, T. Burd e R. Brodersen, "Voltage Scheduling in the IpARM Microprocessor System", In Proceedings of the 2000 International Symposium on Low Power Design, August 2000.
- ◆ K. Govil, E. Chany e H. Wassermanz, "Comparing Algorithms for Dynamic Speed_Setting of a Low_Power CPU", In Proceedings of The First ACM International Conference on Mobile Computing and Networking, Berkeley, CA, November, 1995.
- ◆ J. A. Butts e G. S. Sohi, "A static power model for architects". In Micro-33, pp 191--201, December 2000.

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

99

Bibliografia

- ◆ D. Grunwald, P. Levis, C. Morrey III, M. Neufeld e K. Farkas, "Policies for Dynamic Clock Scheduling", In OSDI, San Diego, CA, October 2000.
- ◆ M. Weiser, B. Welch, A. Demers e S. Shenker, "Scheduling for reduced cpu energy", In First Symposium on Operating Systems Design and Implementation, pp 13--23, November 1994.
- ◆ S. Manne, D. Grunwald e A. Klauser, "Pipeline Gating: Speculation Control for Energy Reduction", In ISCA-25: ACM/IEEE International Symposium on Computer Architecture, June 1998.

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

100

Bibliografia

- ◆ V. Agarwal, M.S. Hrishikesh, S. W. Keckler, and D. Burger. Clock rate versus ipc: The end of the road for conventional microarchitectures. In the 27th Annual International Symposium on Computer Architecture, May 2000.
- ◆ G. Z. N. Cai, C. H. Lim, "Microarchitectural Power Analysis for CPU Power/Performance Optimization", MS JF3-202 MS RA1-311 Intel Corp, Intel Corp.
- ◆ W. Ye, N. Vijaykrishnan, M. Kandemir, and M. Irwin, "The Design and Use of SimplePower: A cycle-accurate energy estimation tool", In Design Automation Conference, June 2000.

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

101

Bibliografia

- ◆ N. Vijaykrishnan, M. Kandemir, M. J. Irwin, H. S. Kim e W. Ye, "Energy-Driven Integrated Hardware-Software Optimizations Using SimplePower", In ISCA 00 Vancouver, British Columbia, Canada, 2000.

29/05/2003

Arquiteturas de Baixo Consumo
Leonardo S. Vidal

102

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.