Processadores para computação de alto desempenho

Aleardo Manacero Jr.

DCCE/UNESP

Grupo de Sistemas Paralelos e Distribuídos

Arquitetura do Conjunto de Instruções

- Tópicos a serem abordados:
  - Métricas das ISAs
  - Classificação de ISAs
  - Diferenciando ISAs
    - Acesso à memória
      - Big-endian x little-endian
      - Modos de endereçamento
    - Tipos de instruções
  - Facilitando a compilação

Métricas das ISAs

- Densidade de instruções
  - É uma medida de quanto espaço é ocupado pelas instruções necessárias para executar uma tarefa. Importante em ambientes com restrição de espaço de armazenamento.
- Quantidade de instruções
  - É uma medida do número de instruções para executar uma tarefa.
- Complexidade de instruções
  - Mede como são as instruções no que diz respeito ao número e tipo de operandos.
- Tamanho de instruções
  - Instruções de tamanho constante ou variável?

Classificação de ISAs

- Memória/memória
  - Podem ter até três operandos na memória
  - Costumam ter poucos registradores
- Registrador/memória
  - Um operando em um registrador e outro na memória
  - Geralmente usa apenas dois operandos
- Carga/armazenamento
  - Apenas transferências entre memória e registradores
  - ULA opera somente com os registradores
  - Típico arranjo em processadores RISC

Classificação de ISAs

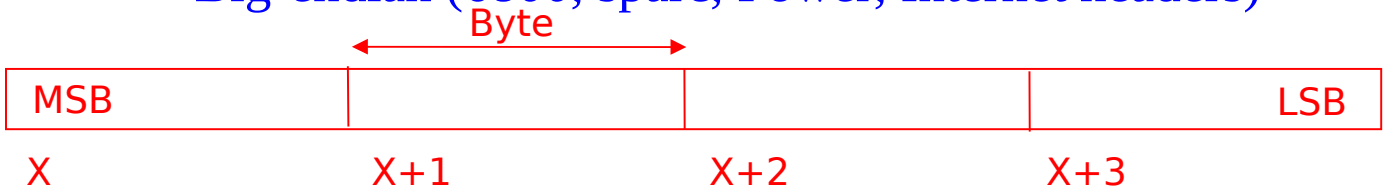
- Acumulador/memória
  - Um operando está no acumulador e outro na memória
  - Instruções movem dados entre o acumulador e a memória
- Baseadas em pilhas
  - Todos os operandos ficam no topo de uma pilha
  - Instruções movem dados entre a pilha e a memória
  - JVM é baseada em pilhas
- Estas classes são menos comuns

Diferenciando ISAs

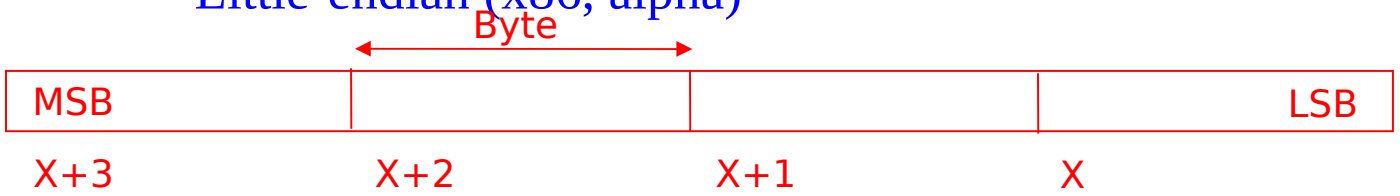
- Registradores são mais rápidos que memória
- Compiladores otimizam o uso de registradores
  - Armazenamento temporário
    - Endereços
    - Variáveis
    - Parâmetros
- Classificação
  - Segundo o número de operandos da ULA
  - Segundo o número de operandos na memória

Acesso à memória

- Quanto à forma
  - Big-endian (6800, sparc, Power, Internet headers)



  - Little-endian (x86, alpha)



  - Versões recentes de sparc e PowerPC são bi-endian
  - Não há vantagem intrínseca em qualquer alternativa
  - Redes costumam transmitir dados no formato big-endian

Modos de endereçamento

- Imediato
  - ADD R4, #7
- Direto em registrador
  - ADD R4, R5, R6
- Direto na memória
  - ADD R4, M[X]
- Indireto
  - ADD R4, M[R1]
- Indexado
  - ADDI R4, M[R1]

Modos de endereçamento

- Relativo
  - Indexado usando o contador de programa como base
- Implícito
  - Usa o topo da pilha como endereço
- Auto-incremento
  - Incrementa registrador base
- Indireções múltiplas

Modos de endereçamento

- Maior variedade de modos
  - Menos instruções
  - Maior densidade de código
  - Implementação mais complexa
  - Pode aumentar o número de CPI (ciclos/instrução)
  - Pode reduzir tempo de execução
- Menor variedade de modos
  - Mais instruções
  - ISA menor, mais simples e de decodificação mais fácil
  - Pode reduzir o número de CPI
  - Pode reduzir tempo de execução

Modos de endereçamento

- Conjunto mínimo
  - Imediato
    - Acesso a constantes (<16bits)
    - Endereços de memória (~32bits)
  - Direto em registrador
    - Acesso a registradores
    - Máquinas baseadas em pilhas são exceção
  - Indireto
    - Acesso a memória

Tipos de instruções

- Instruções básicas
  - Aritméticas
    - Soma, subtração, multiplicação ...
  - Lógicas
    - E, OU, negação, complemento ...
  - Movimento de dados
    - Transferências entre registradores e memória.
    - Deslocamentos de dados em registradores
  - Controle
    - Saltos, chamadas a sub-rotinas ...

Tipos de instruções

- Instruções opcionais
  - Suporte ao sistema
    - Chamadas ao sistema operacional
    - Suporte à memória virtual
- Outras instruções
  - Operações gráficas e vetoriais
- As instruções básicas devem ser rápidas!

Instruções de controle de fluxo

- Tipos de instruções de controle de fluxo
  - Saltos condicionais (>80% do total)
  - Saltos incondicionais
  - Chamadas e retornos de procedimentos
- Endereçamento para o controle de fluxo
  - Relativo
    - código relocável; usado para saltos
  - Indireto
    - usado para retorno de procedimentos e saltos incondicionais
  - Direto na memória
    - usado para acesso à memória

Tamanho das instruções

- Tamanho variável (CISC)
  - Compõem as instruções em pedaços: tipo e especificadores
  - Empacotam mais instruções num mesmo espaço
  - Decodificação mais difícil
  - Necessário buscar instruções desalinhadas
- Tamanho fixo (RISC)
  - Operação e modos de endereçamento fixos no opcode
  - Suporta poucos modos de endereçamento
  - Decodificação mais simples
  - Busca muitas instruções antecipadamente
  - Código menos denso

Facilitando a compilação

- >99% do código é produzido por compiladores (ou interpretadores)
- Passos de um compilador
  - Analisador léxico
  - Analisador sintático
  - Analisador semântico
  - Otimização de código intermediário
  - Geração de código objeto

Facilitando a compilação

- O conjunto de instruções deve ser regular
  - Operações, tipos de dados e modos de endereçamento devem ser ortogonais
- As soluções devem ser primitivas
  - Permita ao compilador construir suas próprias sequências de código.
- Simplificar escolhas
  - Não obrigue o compilador escolher entre 20 alternativas
- Mantenha as coisas o mais simples possível!

Facilitando a compilação

- Não projete uma ISA para uma HLL específica
- Lembre que o programa típico não existe
- Não ceda à tentação de incluir aquela instrução “manera”.
- Não existe ISA sem falhas
- ISAs com falhas podem ser bem sucedidas
- Leva um tempo para que uma ISA fique obsoleta