Conjuntos de Instruções: MIPS, Registradores, Palavras e tipos de instruções

Yuri Kaszubowski Lopes

UDESC

Anotações

Instruções de Máquina

- Para nos comunicar com o processador precisamos “falar a sua língua”
- Alguns exemplos:
  - O seu computador pessoal:
    - x86, AMD64 (x64)
  - O seu Smartphone:
    - ARM
  - Microcontroladores:
    - MIPS, PIC instruction SET
- O Conjunto de instruções está diretamente relacionado com o hardware:
  - Como o hardware interpreta as instruções
  - O quão complexa é a interpretação
  - A quantidade de instruções disponíveis
  - Como as instruções são armazenadas e requisitadas da memória
  - ...

Anotações

MIPS

- Focamos no MIPS de 32 bits (MIPS32)
  - Discutido em Patterson e Henessy (2014)
  - Microprocessor without Interlocked Pipeline Stages
- Desenvolvido por, entre outros pesquisadores, Patterson e Henessy
  - Turing Award de 2017
  - Ideias do MIPS da década de 80 possibilitaram a criação de processadores extremamente eficientes, como os do seu smartphone
  - Diversos processadores de hoje que utilizam a arquitetura MIPS atual
- Conjunto de instruções relativamente simples
- Aprenda um conjunto/arquitetura e migrar para outro conjunto de instruções será (quase) fácil

Anotações

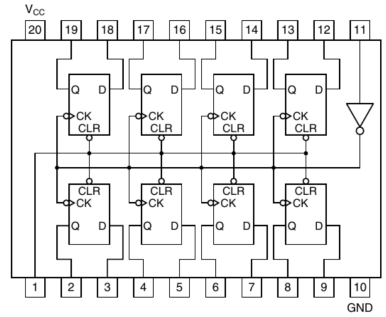
Registradores

- A vasta maioria das arquiteturas atuais (x86-64, MIPS, ARM) operam somente na CPU
- Precisamos carregar os dados para os registradores da CPU
  - Porções de memória na CPU as quais podemos utilizar para realizar operações
- Os registradores são visíveis ao programador
  - Ao menos quando programamos em baixo nível
  - Existem registradores não visíveis

Anotações

Registradores

- Registradores geralmente são construídos com flip-flops



Exemplo de um Circuito Integrado com 8 flip-flops, formando uma memória de 8 bits (Tanenbaum, 2007)

Anotações

Registradores

- Registradores são os dispositivos de memória mais rápidos disponíveis no computador
- Enquanto temos uma abundância relativa de memória principal, os registradores são escassos
  - No MIPS, por exemplo, temos 32 registradores de 32 bits cada
- Quantos registradores seu processador x86 possui?
  - O seu processador x86 tem apenas 8 registradores que usamos em nossos programas
  - 16 registradores no x86-64
  - Os microcontroladores PIC 16F62... possuem apenas um registrador geral (W)
- Cada registrador precisa ter um endereço. Quantos bits são necessários para endereçar todos os registradores do MIPS?

Anotações

Registradores

- São necessários 5 bits para endereçar os registradores do MIPS (2^5 = 32).

Número (Decimal)	Nome Registrador	Descrição
0	$zero,$r0	Sempre contém zero
1	$at	Utilizado para o assembler (montador)
2 e 3	$v0 e $v1	Valores de retorno
4,...,7	$a0,...$a3	Argumentos de função
8,..,15	$t0,...,$t7	Para cálculos temporários (não salvos)
16,...,23	$s0,...,$s7	Registradores salvos (entre chamadas de função)
24 e 25	$t8 e $t9	Mais registradores temporários
26 e 27	$k0 e $k1	Reservados para o Kernel (S.O.)
28	$gp	Apontador de memória global
29	$sp	Ponteiro de pilha
30	$fp	Ponteiro de quadro
31	$ra	Endereço de retorno

Anotações

Registradores

- Focamos inicialmente nos registradores gerais 8 a 15 (não salvos), e 16 a 23 (salvos)
- A máquina entende somente zeros e uns (Linguagem de Máquina)
  - Difícil enxergar que o valor 10001_2 em uma instrução se referencia ao registrador 17_{10}
  - Por essa razão programamos em linguagem de montagem — Assembly
  - Nos referenciamos aos registradores (e operações) por seus nomes
- Os nomes dos registradores em assembly do MIPS começam com `$`
  - e.g., o registrador `$s0` é o registrador 16_{10} ou 10000_2
  - O montador (Assembler) simplesmente traduz de `$s0` para 10000_2 em linguagem de máquina

Anotações

Tamanho da palavra – Word size

- O tamanho “natural” dos dados que um processador lida é denominado word (palavra)
- O tamanho da palavra (word) do MIPS32 é de 32 bits
- No MIPS32, os registradores suportam 32 bits, e as operações **geralmente** lidam com 32 bits
- Processadores diferentes possuem palavras de tamanhos diferentes
  - x86-64 possui uma palavra de 16 bits
    - Mesmo que a arquitetura suporte registradores de 32-bits (x86) e 64-bits (x64); sua palavra de dados remete ao tamanho original de 16-bits
    - byte, word (16-bits), dword (32-bits), qword (64-bits)
  - Os PICs da família 16F62x possuem uma palavra de 8 bits

Anotações

Instruções

- Todas instruções no **MIPS** ocupam **32 bits**
- A consistência facilita o projeto
- x86 por exemplo possui instruções de tamanhos variados
  - Mais flexível, mas o hardware se torna muito mais complexo (e muitas vezes lento)

Exemplo de uma instrução no MIPS

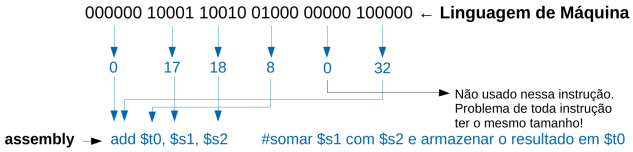
$$\underbrace{00000010001100100100000000100000}_{32\ \text{bits}}$$

- Um tanto difícil interpretar e criar um programa utilizando diretamente as instruções de máquina
  - Esse é um dos motivos de programarmos em Assembly
  - O montador (assembler) consegue traduzir diretamente de Assembly para a linguagem de máquina, e vice-versa
  - Diferente de um compilador, que precisa fazer uma "reinterpretação do código" para transformá-lo em linguagem de máquina

Anotações

Instruções

- No assembly, utilizamos **mnemônicos** ao invés dos bits diretamente para representar uma instrução

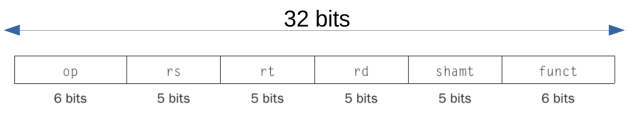


- Para entender como a CPU interpreta a instrução, e como podemos transformar de assembly para linguagem de máquina (e vice-versa), vamos começar a entender a arquitetura MIPS
- A instrução MIPS possui campos com larguras pré-definidas
  - Quais campos são utilizados em quais instruções depende do formato da instrução

Anotações

Instruções (Tipo-R)

- **op**: código básico da instrução, tradicionalmente chamado de opcode
- **rs**: registrador do primeiro operando (fonte)
- **rt**: registrador do segundo operando (fonte)
- **rd**: registrador destino
- **shamt**: Shift Ammount (quantidade de deslocamento)
- **funct**: variante da operação



Anotações

Instruções (Tipo-R)

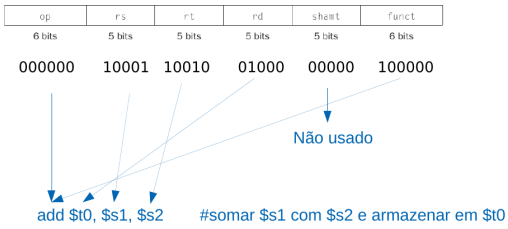
- Qual seria o problema no MIPS se tivéssemos mais de 32 registradores?
  - Os campos **rs**, **rt** e **rd** precisariam de mais bits
  - Sacrificaríamos outros campos, ou então ocuparíamos mais bits com as instruções

Anotações

Instruções (Tipo-R) – exemplo

- `add reg1, reg2, reg3`
  - Some `reg2` + `reg3` e armazene o resultado em `reg1`

```
1   add $t0, $s1, $s2
```



Anotações

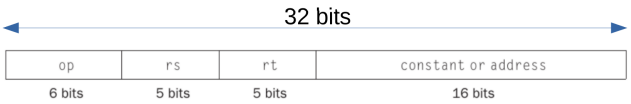
Instruções (Tipo-R vs Tipo-I)

- Instruções do **tipo-R** são fundamentais para lidarmos diretamente com registradores
- Mas e se precisarmos carregar um valor "fixo" para dentro de um registrador?
  - e.g., colocar o valor 2855_{10} em `$s0`
  - Poderíamos utilizar um opcode diferente para especificar que **rs** ou **rt** se referem ao valor a ser carregado, e não o endereço do registrador
  - Qual o problema?
    - Temos apenas 5 bits nesses campos
    - A maior constante que podemos especificar seria 32_{10}
    - Se considerarmos valores com sinal em complemento a 2, nosso intervalo é entre -16 e +15

Anotações

Instruções (Tipo-I)

- Instruções do **tipo-I** servem para (dentre outras coisas) carregar constantes, denominadas valores imediatos, e para acessar a memória
  - **tipo-I**mediato
- Não temos os campos **rd**, **shamt** e **func**
  - Esses campos viram um único campo de 16 bits, onde colocamos o imediato
  - Agora podemos inserir constantes de +/-2^{15} (complemento a dois)
- **op** e **rs** possuem os mesmos significados do tipo-R
- No **tipo-I**, o campo **rt** especifica o destino ou a fonte, dependendo da instrução

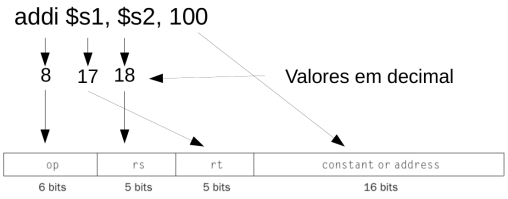


Anotações

Instruções (Tipo-I) – exemplo

- `addi reg1, reg2, imediato`
  - Some `reg2` + `imediato` e armazene o resultado em `reg1`

```
1    addi $s1, $s2, 100
```



Anotações

Referências

- D. Patterson; J. Henessy. **Organização e Projeto de Computadores: Interface Hardware/Software**. 5a Edição. Elsevier Brasil, 2017.
- Andrew S. Tanenbaum. **Organização estruturada de computadores**. 5. ed. São Paulo: Pearson, 2007.
- Harris, D. and Harris, S. **Digital Design and Computer Architecture**. 2a ed. 2012.
- courses.missouristate.edu/KenVollmar/mars/

Anotações

Conjuntos de Instruções: MIPS, Registradores, Palavras e tipos de instruções

Yuri Kaszubowski Lopes

UDESC

Anotações

Anotações

Anotações