RISC-V: Ambiente de montagem e simulação

Orientador: Ricardo Pezzuol Jacobi

Aluno: Matheus Y. Matsumoto

Introdução

- Desenvolvido por Krste Asanoviü, Andrew Waterman, e Yunsup Lee
- Universidade da Califórnia, Berkeley
- colaboração de David Patterson.
- ISA para estudar, implementar com liberdade para fins acadêmicos, porém muito complexas, ou muito específicas, ou simplesmente fechadas.

Introdução

- Por que RISC-V ?
- Open-Source, compatibilidade, extensibilidade, propósito geral, redução de custos, inovação.
- MIPS?
  - Branch delay slot, Position-independent-code, 16-bit imediatos, arquitetura especial para multiplicação e divisão.
- ARM?
  - Não havia suporte 64-bit, thumb, contador de programa é endereçável, enorme quantidade de instruções

Introdução

- Colaboradores:
  - Google, Nvidia, Qualcomm, Samsung, HP, HUAWEI, IBM, Cadence, Raspberry PI, entre vários outros

Fonte:https://riscv.org/members-at-a-glance/

Introdução

- Ambiente de desenvolvimento
  - Este projeto consiste em uma plataforma para aprender e exercitar conhecimentos de programação em assembly RISC-V
  - Consiste em três módulos principais:
    - Editor de texto
    - Montador
    - Simulador

Fundamentação Teórica

- Arquitetura RISC-V
  - ISA RISC *(“Reduced Instruction Set Computing”)*
  - Objetivos: Compacta, completa, simples, eficiênciente, extensível e compatível.
  - Implementações padrões:
    - **RV32I**, RV64I, RV128I

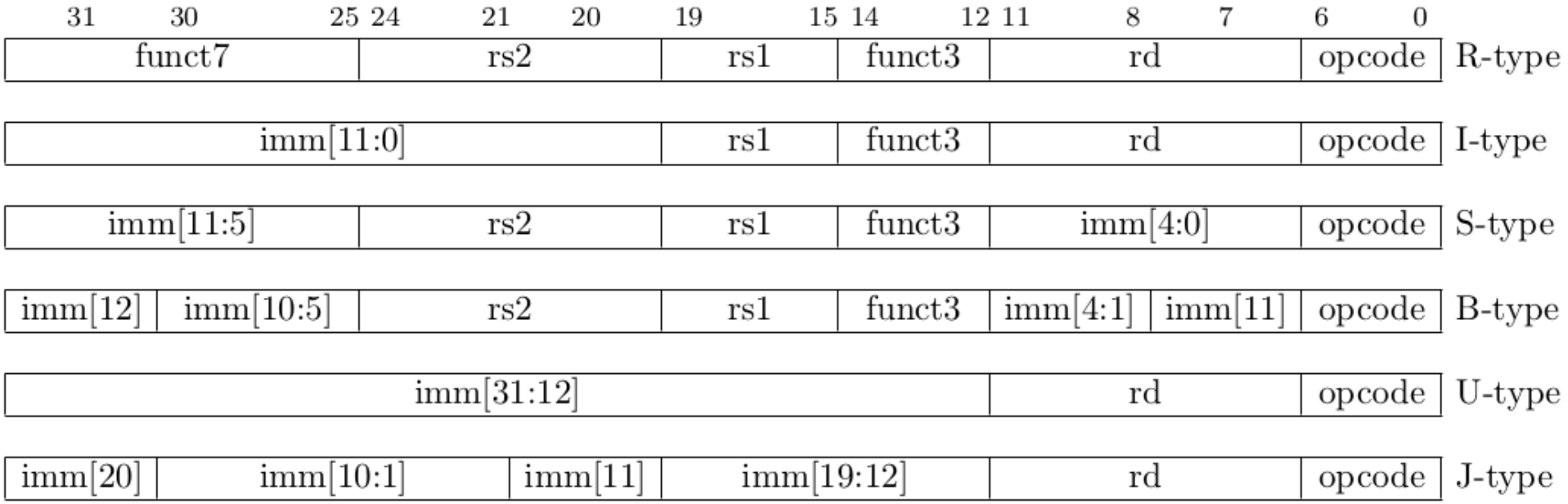
Fundamentação Teórica

- Módulos
  - Módulos padrões:
    - Não possuem conflitos entre si
    - De propósito geral
    - **IMAFD**QL**C**BJTPVN
  - Módulos não-padrões:
    - Propósito específico
    - Podem ser conflitantes

Fundamentação Teórica

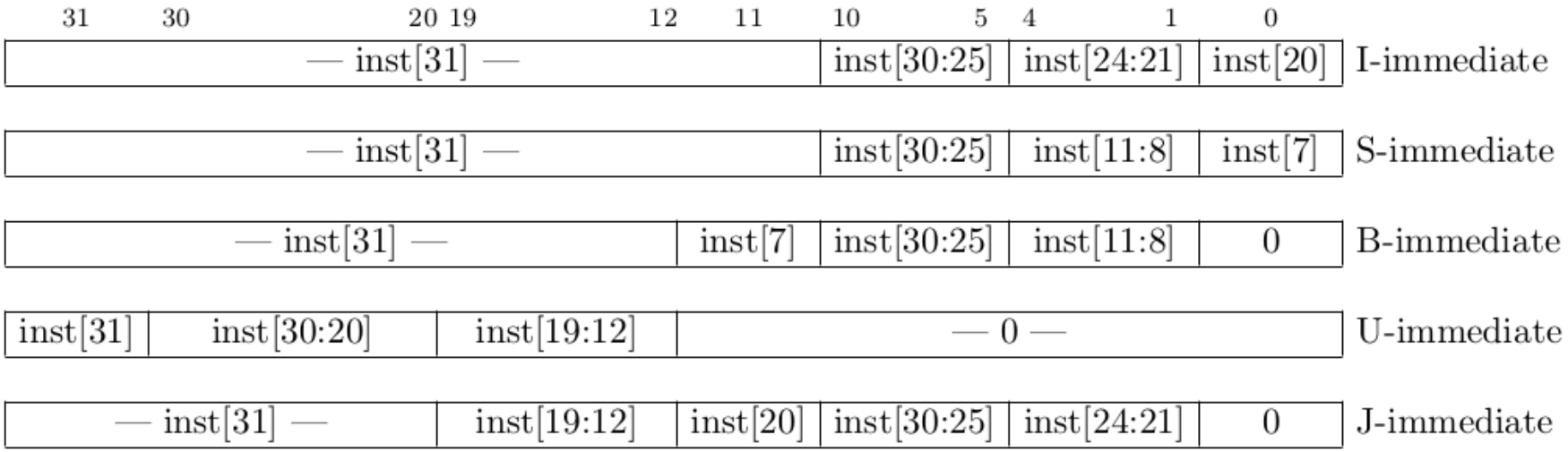
- Instruções
  - RV32I: 47 instruções básicas
    - 38 instruções retirando instruções de controle de estado, chamadas de sistema, sincronização de memória, e substituindo por uma SYSTEM geral.
    - 6 tipos básicos de instrução

Fundamentação Teórica



Fonte: The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 2.2

Fundamentação Teórica



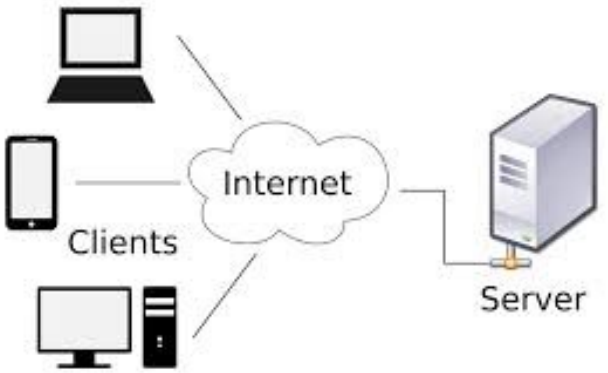
Fonte: The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 2.2

Fundamentação Teórica

- Montador
  - Algoritmo de duas passagens
    - Análise léxica
    - Análise sintática
    - Análise semântica
    - Geração de código

Fundamentação Teórica

- Arquitetura de software:
  - Aplicação WEB modelo cliente-servidor
  -



Fonte: https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Client-server-model.svg/250px-Client-server-model.svg.png

Ambiente Proposto

- O ambiente proposto consiste em três módulos principais:
  - Editor de texto
  - Montador
  - Simulador

Ambiente Proposto

- Single Page Application
  - Todo o sistema é renderizado em uma chamada GET
  - Ações são carregadas dinâmicamente através de requisições GET/POST pelo Javascript

Ambiente Proposto

- Front-end
  - riscv_flow
    - Fluxo de navegação
      - Ex: editor → montador
  - riscv_functions
    - Chamadas ao back-end
      - Ex: assemble()

Ambiente Proposto

- Back-end
  - Consiste dos dois módulos restantes
    - Montador
    - Simulador

Ambiente Proposto

- Montador
  - Como todo o resto do sistema, foi implementado como um pacote python
  - Algoritmo de duas passagens
  - Entrada: Código assembly RISC-V
  - Saídas: Código montado em diferentes formatos e erros para códigos mal formados

Ambiente Proposto

- Simulador
  - Pacote python
  - Recebe o código montado como uma lista JSON do código em binário
  - Ações: RUN, STEP, AUTO-RUN, RESET
  - Saída: Memória, Registradores, Mensagens na console, bitmap da memória, código montado

Resultados e Avaliação do Sistema

- Demonstração
- http://riscv.ymmatheus.com/

Conclusões

- Neste trabalho implementamos:
  - Ambiente de desenvolvimento
  - Montador
  - Simulador

Conclusões

- Vantagens
  - Não necessita instalação
  - Multiplataforma
- Desvantagens
  - Performance
  - Indisponibilidade

Conclusões

- Melhorias futuras
  - Expansão para 64 bits
  - Novos módulos (novas instruções)
  - Contador de instruçõse
  - Adição de instruções e pseudo instruções
  - Usabilidade
  - Interface Mobile