Arquitetura de Computadores

Reduced Instruction Set Computer: RISC-V

Rodolfo Labiapari Mansur Guimarães

Última Atualização: 31 de outubro de 2016

rodolfolabiapari@gmail.com
Lattes: `http://goo.gl/MZv4Dc`
Departamento de Computação – Universidade Federal de Ouro Preto
Ouro Preto - MG – Brasil

Introdução

Introdução ao RISC-V

- Desenvolver um CPU requer experiência ao design em várias especialidades:
  - Eletrônica Digital, Compiladores e Sistemas Operacionais.
- Quando um projeto baseado de uma série de projetos acadêmicos de design de computadores.
- Foi desenvolvido por profissionais e voluntários entusiastas obtêm-se o resultado um moderno e de alta qualidade conjunto de instruções para computador de propósito geral, o RISC-V.
- Para projetar tal ISA, os autores possuíam vasta experiência em pesquisa na área além de simulação e validação dos design do projeto.
- RISC-V é um novo conjunto de instruções de arquitetura com **propósito geral**.

UFOP

Introdução ao RISC-V

- Produzido pelo Computer Science Divison na Universidade da Califórnia, Berkeley, é um *Instruction-Set Architecture* (ISA) *open-source* (BSD)
  - Possui várias colaborações de voluntários entusiastas.
- É um conjunto **limpo**, **modular** com inteiros bases em 32, 64 e 128 bits e várias opções de extensão de instruções como ponto-flutuante, multiplicadores, etc.
- Cada usuário demanda que os designers considerem a performance e eficiência energética quando projetar processador
  - É um conjunto de fácil implementação comparado a outras alternativas e o projeto possui grande aceitação na indústria de semicondutores.
- A meta do projeto RISC-V é criar um conjunto de instruções 'universal' que é livre e aberto para todos os usuários, provendo tudo que é necessário para suportar perfeitamente qualquer projeto comercial.

UFOP

Introdução ao RISC-V

- Em contraste com a maioria dos ISA disponíveis atualmente, RISC-V é disponibilizado livremente para todos usarem bem como quiserem.
  - Incluindo fazer *design*, fabricar e vender os chips e *software* RISC-V.
- Não é o primeiro projeto desse porte *open-souce* a aparecer, mas seu projeto é modelado para ser útil nos dispositivos computacionais mais modernos que vão desde servidores, celulares *high-end* até projetos embarcados de pequeno porte.
- Em contraste disso, chips da ARM e MIPS Technologies necessitam de licença para uso de suas patentes
  - Também requerem acordos de confidencialidade para uso de seus documentos que citam as vantagens de seus design e conjunto de instruções.
- O ISA RISC-V tem sido desenvolvido com o mais pequeno, rápido e mínimo gasto de energia de implementações existente hoje
  - Sem sobrecarregar outras partes do seu sistema.

Introdução ao RISC-V - Exemplificação

- Consideremos um smartphone moderno:
  - Possui dúzias de *cores* com diferentes pilhas de *softwares*
    - Uma aplicação ARM no CPU;
    - Uma GPU;
    - 3 à 5 DSP;
    - Um *core* de gerenciamento de energia.
  - Em teoria, todos usam uma variante de um ISA simples e muitos casos, reutilizando *hardware* e *software*.
  - RISC-V poderia oferecer uma opção prática para unificar todos esses *cores* num único local.

UFOP

Introdução ao RISC-V - Propósito

- Seu propósito de mercado é amplo ao ponto de ser suportado para executar em:
  - Microcontroladores que processem imagens, gráficos; ou até mesmo
  - Processadores de servidores.
- Assim, o ISA deve ser consistente em microarquiteturas. É direcionado para ser adequado para quase todo tipo de implementação.
  - Desde design Scalar in-order até design out-of-order.
  - Similarmente, é projetado para ser adequado a quase todas implementações. Desde macros sintetizações em FPGA até projetos totalmente customizados.
- Por causa do crescente interesse em aceleração em processamento, **extensibilidade** do RISC-V é parte essencial da universalidade
  - Podendo adicionar módulos, temos uma arquitetura que atende a todos os projetos de *hardware*.
- Para ativar as extensões, porções do espaço de codificação de instruções já foram reservados para uso futuro.

Design do RISC-V

O RISC-V em Termos Gerais

- RISC-V possui 32 registradores inteiros, e 32 registradores opicionais para ponto flutuante.
  - Também existe uma variante RISC-V com 16 registradores inteiros.
- Sua memória possui endereçamento de 1 byte.
- O assembler ussa o registrador x0 como um espaço reservado para fazer manuleio de algumas operações, por exemplo o `move rx to ry` seria
  1. Soma rx com 0: `add r0 to rx`;
  2. Salva o resultado em ry: `store in ry`.
- Registradores de controle e status também, mas somente o nível de privilégio mais alto (user) que poderá acessá-los para medição de performance.

UFOP

O RISC-V em Termos Gerais

- Como todos os outros design de RISCs, RISC-V é um máquina load-store.
  - Somente estas duas instruções que acessam a memória principal;
  - Todas as operações lógico-aritméticas ocorrem entre registradores.
- Diferentemente de outros projetos RISC acadêmicos otimizados para simplicidade, o conjunto de instruções do RISC-V é desenvolvido para a praticidade de implementações, com características que aumentam a velocidade computacional enquanto reduz seu curso e energia.
- Várias otimizações foram feitas como
  - Colocar os bits mais significantes numa posição fixa;
  - Disposição de bits para reduzir o número de multiplexadores no CPU;

O RISC-V em Termos Gerais

- Foi projetado para alcançar altos e baixos níveis de velocidade com pouco gasto de energia e eletrônicos.
- Todas suas instruções possuem 32 bits. Isso gera simplicidade, mas códigos com mais instruções.
- RISC-V, intencionalmente, não possui códigos condicionais, nem mesmo bit de *carry*
  - Afirmam que isso pode simplificar o desenvolvimento do CPU, minimizando interações entre instruções.
  - Construíram operandos de comparação dentro dos *jumps* condicionais.
  - Não possui *carry* de operações aritméticas complicadas.
  - Não tem detector ou flag para erros aritméticos, incluindo *overflow*, *underflow* e divisão por 0.

O RISC-V em Termos Gerais

- Foi desenvolvido para suportar sistema de memória e instruções inteira e ponto flutuante de 32, 64 e 128 bits.
- Suas funções load e store pode realizar operações com 16 e 8 bits, mas não operações aritméticas.
    - Já as instruções de 64 bits, incluem aritméticas de 32-bits.

Introdução ao RISC-V - Propósito

- RISC-V provê garantia de funcionamento para:
  - Instruções de inteiros tamanho 32, 64 e 128 bits;
  - Uma família de extensões opcionais/predefinidas.
- Embora alguns processadores disponíveis hoje no mercado sejam amplamente utilizados, eles são:
  - Complexos; e
  - É difícil de ser utilizado para experimentação e uso acadêmico.
- Com isso, tendo construído uma arquitetura **simples** e **clara**, tem-se então implementações finais **simplificadas**.

UFOP

De Projeto Acadêmico à Item Comercial

UFOP

O Projeto RISC-V

- Projeto iniciado em 2010, foi disponibilizado em 2015 e já é utilizado em indústrias de grande porte como Google, Mellanox e Oracle além de grandes centros acadêmicos.
- Com isso vários projetos funcionais online já estão disponíveis, usufruindo da permissão de licença BSD.
- Um exemplo é o escalar de 5 estágios chamado RISC-V Rocket[1] em Scala e para sintetização em chips[2].

[1] Disponível em: `https://github.com/ucb-bar/rocket`
[2] Disponível em: `https://github.com/ucb-bar/rocket-chip`

ISA Modular

RISC-V - Extensões

Base ISA	Instructions	Description
RV32I	47	32-bit address space and integer instructions
RV32E	47	Subset of RV32I, restricted to 16 registers
RV64I	59	64-bit address space and integer instructions, along with several 32-bit integer instructions
RV128I	71	128-bit address space and integer instructions, along with several 64- and 32-bit instructions
Extension	**Instructions**	**Description**
M	8	Integer multiply and divide
A	11	Atomic memory operations, load-reserve/store conditional
F	26	Single-precision (32 bit) floating point
D	26	Double-precision (64 bit) floating point; requires F extension
Q	26	Quad-precision (128 bit) floating point; requires F and D extensions
C	46	Compressed integer instructions; reduces size to 16 bits

Modelo de Instruções

Base ISA	Instructions	Description
RV32I	47	32-bit address space and integer instructions
RV32E	47	Subset of RV32I, restricted to 16 registers
RV64I	59	64-bit address space and integer instructions, along with several 32-bit integer instructions
RV128I	71	128-bit address space and integer instructions, along with several 64- and 32-bit instructions
Extension	**Instructions**	**Description**
M	8	Integer multiply and divide
A	11	Atomic memory operations, load-reserve/store conditional
F	26	Single-precision (32 bit) floating point
D	26	Double-precision (64 bit) floating point; requires F extension
Q	26	Quad-precision (128 bit) floating point; requires F and D extensions
C	46	Compressed integer instructions; reduces size to 16 bits

- O modelo de programação **RV32I** é escasso. Nele é contido:
  - *Program Counter* e 32 registradores inteiros (x0 - x31).
  - Não contém registrador de retorno sendo este o x1.
- As instruções possuem 32 bits possuindo várias combinações de imediatos, operandos e outra especificações.
  - O opcode e operando ficam em locais fixos, facilitando o processo de decodificação.

Modelo de Memória

- Sua memória é endereçada por byte e utiliza-se *little-endian*.
- Enquanto outros processadores utilizam complexos instruções e modos de endereçamento, RISC-V usa somente *base+offset*.
- Podem operar em dados de tamanho 8 (byte), 16 (half-word) ou 32 (word) bits além das opções de sinalização.

UFOP

Variantes e Extensões

RISC-V - Variantes

- Todas as outras variantes do RISC-V são baseadas do **RV32I**.
- **RV32E**:
  - Possui a meta para ser satisfatório para implementações de **pequeno porte**.
  - Sendo um projeto menor, ocupa-se uma pequena porção de área *die*.
    - Aumenta o custo/benefício de produção de chips;
    - Melhora a dissipação de calor por se tratar de um chip com menos área de circuito;
    - Isso reduz registradores para somente o *Program Counter* e x0-x15.
  - De resto, RV32E e RV32I são idênticos, com exceção de:
    - Suas instruções continuam com tamanho de 32 bits. Os bits de maior índices são 0.
- **RV64I** e **RV128I**:
  - Simples extensões de 64 e 128 bits.
  - Aumentam o espaço de endereço e estende os registradores de 32 para o tamanho apropriado.
  - Essa extensão introduz novas instruções que operam com dados menores como 32 e 64 bits.

RISC-V - Extensões - Multiplicação

Base ISA	Instructions	Description
RV32I	47	32-bit address space and integer instructions
RV32E	47	Subset of RV32I, restricted to 16 registers
RV64I	59	64-bit address space and integer instructions, along with several 32-bit integer instructions
RV128I	71	128-bit address space and integer instructions, along with several 64- and 32-bit instructions
Extension	**Instructions**	**Description**
M	8	Integer multiply and divide
A	11	Atomic memory operations, load-reserve/store conditional
F	26	Single-precision (32 bit) floating point
D	26	Double-precision (64 bit) floating point; requires F extension
Q	26	Quad-precision (128 bit) floating point; requires F and D extensions
C	46	Compressed integer instructions; reduces size to 16 bits

- **Multiplicação M**:
  - Adiciona 4 instruções de multiplicação, duas de divisão, e duas de manipulação de restos.
  - A base é indicada pela base do ISA.
  - A divisão por 0, por exemplo, não causa interrupção, mantendo a simplicidade.

RISC-V - Extensões - Sincronização

Base ISA	Instructions	Description
RV32I	47	32-bit address space and integer instructions
RV32E	47	Subset of RV32I, restricted to 16 registers
RV64I	59	64-bit address space and integer instructions, along with several 32-bit integer instructions
RV128I	71	128-bit address space and integer instructions, along with several 64- and 32-bit instructions
Extension	**Instructions**	**Description**
M	8	Integer multiply and divide
A	11	Atomic memory operations, load-reserve/store conditional
F	26	Single-precision (32 bit) floating point
D	26	Double-precision (64 bit) floating point; requires F extension
Q	26	Quad-precision (128 bit) floating point; requires F and D extensions
C	46	Compressed integer instructions; reduces size to 16 bits

- **Sincronização A**:
  - Adiciona 11 instruções de sincronização visando consistência e atomicidade da operação.
  - São divididos em dois grupos: *Load* Reserved e *Store* Conditional para operações atômicas na memória.

RISC-V - Extensões - Ponto Flutuante

Base ISA	Instructions	Description
RV32I	47	32-bit address space and integer instructions
RV32E	47	Subset of RV32I, restricted to 16 registers
RV64I	59	64-bit address space and integer instructions, along with several 32-bit integer instructions
RV128I	71	128-bit address space and integer instructions, along with several 64- and 32-bit instructions
Extension	**Instructions**	**Description**
M	8	Integer multiply and divide
A	11	Atomic memory operations, load-reserve/store conditional
F	26	Single-precision (32 bit) floating point
D	26	Double-precision (64 bit) floating point; requires F extension
Q	26	Quad-precision (128 bit) floating point; requires F and D extensions
C	46	Compressed integer instructions; reduces size to 16 bits

- Possui três extensões diferentes:
  - Precisão simples (**F**), Precisão Dupla (**D**) e quádrupla precisão (**Q**).
- A extensão F é pré-requisito para D, que é pré-requisito para Q.
- Introduz também 32 registradores de ponto flutuante (f0-f31) de tamanho 32 bits.
- Registrador de 5 bits para exceções.
  - Exceções não geram interrupções. Devem ser verificadas por meio de consultas.
- As instruções *load-store* usam o mesmo endereçamento *base+offset*.

RISC-V - Extensões - Compressão de Tamanho de Código

Base ISA	Instructions	Description
RV32I	47	32-bit address space and integer instructions
RV32E	47	Subset of RV32I, restricted to 16 registers
RV64I	59	64-bit address space and integer instructions, along with several 32-bit integer instructions
RV128I	71	128-bit address space and integer instructions, along with several 64- and 32-bit instructions
Extension	**Instructions**	**Description**
M	8	Integer multiply and divide
A	11	Atomic memory operations, load-reserve/store conditional
F	26	Single-precision (32 bit) floating point
D	26	Double-precision (64 bit) floating point; requires F extension
Q	26	Quad-precision (128 bit) floating point; requires F and D extensions
C	46	Compressed integer instructions; reduces size to 16 bits

- A última extensão **C** não adiciona nenhum outra função, mas, ao invés disso, **codifica** as instruções inteiras para **salvar espaço** e com isso reduzir o tamanho do *footprint*.
- É disponível para bases inteiras, bem como *load* e *store* para pontos flutuantes.
- Cria-se instruções compressadas em 16 bit
  - Basicamente, cada função compactada é mapeada diretamente à instrução real.
- Possui algumas restrições para a compressão sobre o formato dos operandos.

RISC-V - Extensões - Compressão de Tamanho de Código

Base ISA	Instructions	Description
RV32I	47	32-bit address space and integer instructions
RV32E	47	Subset of RV32I, restricted to 16 registers
RV64I	59	64-bit address space and integer instructions, along with several 32-bit integer instructions
RV128I	71	128-bit address space and integer instructions, along with several 64- and 32-bit instructions
Extension	**Instructions**	**Description**
M	8	Integer multiply and divide
A	11	Atomic memory operations, load-reserve/store conditional
F	26	Single-precision (32 bit) floating point
D	26	Double-precision (64 bit) floating point; requires F extension
Q	26	Quad-precision (128 bit) floating point; requires F and D extensions
C	46	Compressed integer instructions; reduces size to 16 bits

- O propósito de reduzir o tamanho do código binário, energia e custo para pequenos computadores, é visando sistemas embarcados.
- Pesquisas com **C** mostram que um código 20% menor que um x86 e MIPS Comprimido e 2% maior que um ARM Thumb-2.

Níveis de Privilégios

RISC-V - Níveis de Privilégios

- Existe 4 níveis de privilégios:
  - User (**U**);
  - Supervisor (**S**);
  - Hypervisor (**H**); e
  - Machine (**M**).
- Para reduzir o custo de uma implementação mínima, somente o *machine* é requerido.
  - Entretanto, todos os 4 privilégios são suportados.
  - Quando suportados, é possível até executar o Linux.

UFOP

O Ecossistema Open-Source

RISC-V - O Ecossistema do Projeto

- O projeto ainda está incompleto, mas as especificações já mostram que é bastante promissor.
- Sua arquitetura é mais fácil de ser implementada do que um ARM, por exemplo:
  - Pela sua simplicidade em:
    - Decodificar códigos;
    - Nos modos de endereçamento.
  - Omite instruções complexas;
  - Tudo isso com **metade** do tamanho de projeto de um ARM *core*.
- É considerado como a técnica mais **promissora** em flexibilidade de extensões customizadas combinado com arquitetura de propósito geral.

UFOP

RISC-V - O Ecossistema do Projeto

- Extensões potenciais do RISC-V incluem novos tipos como
  - Extensões Cray-style vector;
  - Memória Transacional; e
  - Manipulação de bit.
- Outras extensões são disponibilizadas para obter melhor desempenho, mas requerem cadeias de ferramentas e limitam a portabilidade de *software*.

RISC-V - O Ecossistema do Projeto

- O ecossistema RISC-V ainda está nos estágios iniciais
  - Um **Linux** 4.1 foi sido portado para o projeto RISC-V
    - E um Linux Embarcado (**Yocto**) também foi disponibilizado.
  - Várias ferramentas do RISC-V incluem **compiladores** GCC, LLVM e Clang, GDB, uma suite de **verificação** e **simuladores**.
- Oferece todos os recursos básicos de um RISC com implementações simplificadas, reduzindo a *die area* e potencialmente, energia consumida por ele.
- Comparado os dois modelos mais populares (ARM e x86)
  - Oferece uma considerável área de circuito não utilizada em placa deixando o projeto menos custoso;
  - Permite a adição de extensões customizadas ou não.
  - É um projeto *open-source*.

Adotantes do Projeto

RISC-V - Adotantes do Projeto

- Um número de organizações comerciais planejam suportar o RISC-V Foundation
  - Bluespec, Inc., Google, Hewlett Packard Enterprise (HPE), Lattice Semiconductor, Mellanox Technologies, Microsemi, Oracle e Rambus Cryptography Research.
- O Instituto Indiano de Tecnologia Madras estão desenvolvendo 6 RISC-V *open-source* para 6 tipos de usuários diferentes.
  - De um pequeno CPU 32 bit para IoT até largos computadores de 64 bit concebido para computadores em escala de de armazenamento.
- lowRISC é um projeto sem fins lucrativos que visa implementar um sistema *open-source* num *System on Chip* (SoC) baseado num RISC-V de 64 bits.

RISC-V - Adotantes do Projeto

- O Laboratório de Computação, na Universidade de Cambridge, em colaboração com o FreeBSD Project, tem suportado o sistema operacional FreeBSD para o RISC-V 64-bit.
- ETH Zurich e a Universidade de Bologna têm desenvolvido em cooperação um System on Chip (SoC) de baixa energia usando RISC-V.
- Um dos fundadores do Adapteva planeja usar RISC-V como sucessor para o seu produto acelerador multicore.
- Nvidia planeja usá-lo para substituir a linha de processadores Falcon em suas placas gráficas GeForce.

Arquitetura de Computadores

Reduced Instruction Set Computer: RISC-V

Rodolfo Labiapari Mansur Guimarães

Última Atualização: 31 de outubro de 2016

rodolfolabiapari@gmail.com
Lattes: `http://goo.gl/MZv4Dc`
Departamento de Computação – Universidade Federal de Ouro Preto
Ouro Preto - MG – Brasil

Arquitetura de Computadores

Reduced Instruction Set Computer: RISC-V

Rodolfo Labiapari Mansur Guimarães

Última Atualização: 31 de outubro de 2016

rodolfolabiapari@gmail.com
Lattes: `http://goo.gl/MZv4Dc`
Departamento de Computação – Universidade Federal de Ouro Preto
Ouro Preto - MG – Brasil