



Introdução a arquitetura e montagem de ambiente de desenvolvimento para plataforma RISC-V

Lenin Cristi

CMCC – Universidade Federal do ABC (UFABC)
Santo André – SP – Brasil

lenin.cristi@aluno.ufabc.edu.br

Resumo. *O presente artigo traz uma breve introdução histórica da arquitetura RISC-V, um panorama dela frente as alternativas de mercado atuais, um comparativo de arquiteturas e um manual de provisionamento de ambiente de desenvolvimento para software nesta plataforma.*

Sumário

| | |
|--------------------------------------------------------------------------------------|----|
| 1. Introdução | 3 |
| 2. Objetivos do trabalho | 3 |
| 3. Conceitos e fundamentos | 3 |
| 3.1. O que é uma ISA | 3 |
| 3.2. RISC e CISC | 3 |
| 4. Como surgiu a RISC-V (uma revisão histórica) | 4 |
| 4.1. Projetos anteriores e nascimento da arquitetura | 4 |
| 4.2. Projetos atuais e derivados | 5 |
| 4.3. ARM | 5 |
| 5. Comparativo de arquiteturas | 5 |
| 5.1. Pontos chave de diferenciação CISC & RISC | 5 |
| 6. Estado do mercado e perspectivas | 7 |
| 6.1. O caso Nvidia | 7 |
| 6.2. A importância de uma plataforma aberta | 8 |
| 7. Provisionamento de um Ambiente de desenvolvimento | 8 |
| 7.1. Passos preliminares (Windows somente) | 8 |
| 7.2. Dependências do toolchain | 9 |
| 7.3. Build do RISC-V Toolchain, emulador e proxy | 9 |
| 7.4. Teste da instalação e IDE's | 10 |
| 8. Conclusão | 10 |
| 9. Referências bibliográficas | 11 |
| 9.1. Artigos | 11 |
| 9.2. Referências gerais: Definição | 11 |
| 9.3. Referências gerais: Apple, Qualcomm e nuvens com chips customizados | 12 |
| 9.4. Referências gerais: O caso Nvidia e declarações da câmara de comercio americana | 13 |
| 9.5. Referências técnicas | 13 |

1. Introdução

RISC-V é uma arquitetura aberta e livre de royalties de conjunto de instruções ISA para construção de processadores. O termo RISC significa “*Reduced Instruction Set Computing*” numa contraposição a arquiteturas CISC “*Complex Instruction Set Computer*”, a letra “V” indica a quinta iteração do projeto original.

Essa linha de arquitetura favorece um conjunto reduzido e simples de instruções que tenham aproximadamente o mesmo tempo de execução e sejam usadas com uma média de frequência mais alta que as instruções mais complexas encontradas em arquiteturas CISC.

É uma arquitetura mantida pela fundação sem fins lucrativos suíça “*RISC-V Foundation*” derivada da arquitetura RISC original, que também originou a popular arquitetura ARM mas por ser completamente aberta para uso acadêmico e comercial, tem ganhado destaque entre fabricantes para uso em desenvolvimento de chips de computação móvel, pessoal e especializados.

2. Objetivos do trabalho

Os objetivos do trabalho incluem apresentar brevemente conceitos chave para entendimento das arquiteturas apresentadas, um breve histórico centrado na criação e evolução da arquitetura RISC-V, um comparativo das arquiteturas RISC e CISC, um panorama de mercado e das oportunidades de uso para a arquitetura RISC-V e por fim o provisionamento de um ambiente de desenvolvimento e emulação de uma plataforma RISC-V capaz de gerar código executável para um chip RISC-V.

3. Conceitos e fundamentos

3.1. O que é uma ISA

ISA ou “*Instruction Set Architecture*” se refere à arquitetura do conjunto de instruções de um processador ou de uma família de processadores, e define o conjunto e formato de instruções que um processador pode executar. Ela atua como uma interface entre software e hardware fornecendo um conjunto estável e previsível de instruções que podem ser usadas na confecção de um software que será executado naquela família de processadores.

3.2. RISC e CISC

RISC e CISC são as duas principais categorias de ISA’s atualmente:

RISC ou “*Reduced Instruction Set Computing*” onde o conjunto de instruções é projetado para ser reduzido ou simples e executar operações básicas de forma mais eficiente. Os processadores RISC tendem a ter um número menor de instruções, cada uma executando aproximadamente num único ciclo de clock. Isso simplifica o design do hardware e, muitas vezes, resulta em um desempenho e eficiência energética mais previsível.

CISC ou “*Complex Instruction Set Computing*” onde o conjunto de instruções é mais extenso e complexo, o que permite a realização de operações mais sofisticadas em uma única instrução. Embora esse conjunto de instruções mais completo permita uma facilidade de desenvolvimento, como as extensões SSE e AVX que aceleram a operação de produtos multimídia como o Adobe Premiere, ou as extensões AES que aceleram a operação de soluções de criptografia “*data-at-rest*” como o bitlocker do Windows ou o LUKS do Linux, os processadores CISC podem levar mais ciclos de clock para completar

uma instrução e ter parte do seu conjunto de instruções subutilizado para determinados cenários e cargas de trabalho.

4. Como surgiu a RISC-V (uma revisão histórica)

4.1. Projetos anteriores e nascimento da arquitetura

O termo RISC data por volta de 1980, mas antes disso já havia estudos em curso no sentido de que computadores mais simples poderiam ser mais efetivos, notadamente no trabalho de John Cocke, notório cientista da computação e pesquisador da IBM, laureado com diversos prêmios entre eles o Turing e o Von Neumann e considerado o pai da arquitetura RISC.

A formalização destes estudos, no entanto veio no trabalho *“The case for the reduced instruction set computer”* [1] de David Patterson e David Ditzel que questionava entre outras coisas a presunção de que arquiteturas mais complexas teriam custo mais efetivo.

No ano seguinte, em 1981 D. Patterson publicou as duas primeiras especificações RISC-I e RISC-II liderando o projeto da universidade de Berkley intitulado *“Berkley RISC”* que era um de dois projetos RISC em curso financiados pela DARPA (acrônimo de *“Defense Advanced Research Projects Agency”*), o outro projeto RISC em curso estava na universidade vizinha de Stanford e originou a arquitetura MIPS e este por sua vez se estendeu de 1981 a 1984.

O projeto de Berkley no entanto foi tão bem sucedido que mesmo a arquitetura MIPS desenvolvida em Stanford depois veio a ser reconhecida como um tipo de RISC e o projeto *“Berkeley RISC”* assumiu o termo RISC para si. Este projeto originou as arquiteturas SPARC da Sun Microsystems e mais tarde a própria arquitetura ARM.

O projeto *“Berkeley RISC”* chegou a 1984 com a especificação RISC-III e em 1988 com a especificação RISC-IV que a esta altura já contava com software, simuladores e propostas de designs de chips.

Em 2010, o pesquisador Krste Asanović também de Berkeley recebeu o requisito de desenvolver uma plataforma aberta de computação tanto para uso acadêmico como comercial, David Patterson se juntou ao projeto como desenvolvedor original do *“Berkley RISC”* e o termo *“RISC-V”* foi cunhado como menção a longa cadeia de desenvolvimento até ali.

A especificação ISA em si (ou a codificação de um conjunto de instruções) foi publicada como aberta em 2011 e depois colocada em licença *“Creative Commons”* o que permitiria seu aprimoramento por contribuidores externos.

A fundação RISC-V surgiu em 2015 da necessidade de estabilizar o conjunto ISA da RISC-V para que pudesse ser usado para fins comerciais, e ela é a responsável por manter e publicar material e propriedade intelectual relativa à arquitetura RISC-V. Os autores originais passaram os direitos de propriedade para a fundação na sua criação. Em novembro de 2019 a fundação moveu sua sede para a suíça como uma organização sem fins lucrativos por temores de que restrições do departamento de estado americano pudessem afetar o caráter aberto e colaborativo do projeto. No mesmo ano a fundação publicou documentos que definiram a arquitetura e permitiram uso irrestrito tanto para software quanto hardware na arquitetura.

Em 2022 restrições impostas pelo departamento de estado americano a Nvidia quanto a exportação de chips capazes de acelerar desenvolvimento de IA em determinados países (incluindo a China) mostraram que os temores que motivaram a realocação da fundação não eram infundados.

4.2. Projetos atuais e derivados

Existem vários projetos proprietários e abertos atuais ou em desenvolvimento utilizando a plataforma RISC-V, entre eles:

Proprietários: Módulos de segurança Titan M2 da Google usados a partir do Pixel 6; Em 2020 a Seagate anunciou o desenvolvimento de dois chips de uso geral para sua linha de storages; Em 2020 O departamento de defesa indiano começou o uso de chips 65-bits desenvolvidos na IIT-Madras e fabricados pela intel. Em 2016 a Nvidia anunciou planos de usar a arquitetura RISC-V para substituir processadores Falcon nas placas gráficas da linha GeForce, em 2022 esta tecnologia já era utilizada na sua linha Tesla para datacenters.

Abertos: Chips Rocket (computação pessoal de baixo consumo) e BOOM (pessoal e supercomputação) da universidade de Berkeley; Chip SERV de Olof Kindgren, o menor chip RISC-V construído; XuanTie um chip de 2019 com 64-bits e 16 cores a 2,5GHz do Alibaba Group, e que foi tornado aberto em 2021; XiangShan um chip de alta performance da Chinese Academy of Sciences.

4.3. ARM

O mais notório projeto derivado da arquitetura RISC atualmente é o ARM (originalmente Acorn RISC Machine, e depois Advanced RISC Machine) desenvolvido pela empresa britânica ARM holdings e licenciado para empresas que os implementam em seus próprios produtos.

Inicialmente teve uma adoção ampla em dispositivos móveis por características como em geral precisarem de uma quantidade menor de transistores, permitirem um consumo e custo mais efetivo e com maior eficiência e menor dissipação de calor que chips CISC. Essa adoção em dispositivos moveis garantiu a ARM o posto de arquitetura mais usada no mundo com 230 bilhões de chips produzidos até 2022.

Atualmente a Qualcomm ensaia sua entrada no mercado de PCs domésticos com uma lista ampla de sistemas operacionais populares já portados para a plataforma. Entre eles o RTOS, BSD, Android, Fedora, Debian, Ubuntu, iOS, Windows 8, 10 e 11.

A Apple em 2020 na sua conferência de desenvolvedores anunciou o fim da parceria de 15 anos com a Intel e o início de sua fabricação independente de chips baseados em ARM, que viria a se tornar a família de chips M1, M2 e M3 que são empregados em seus notebooks e desktops mais recentes. Este movimento de fabricação de chips ARM foi seguido por outras empresas como Nvidia, AMD, Amazon e Microsoft.

5. Comparativo de arquiteturas

5.1. Pontos chave de diferenciação CISC & RISC

Instruções mais simples ou reduzidas. Arquiteturas RISC são voltadas a um conjunto menor de instruções desenhadas para uma operação básica. Isto reduz a quantidade de decodificação e

processamento necessário para o processador executar cada instrução, acelerando o tempo de execução.

Load-store. Como toda arquitetura RISC, a arquitetura RISC-V é do tipo “*load-store*” em contraposição a “*register-memory*” das arquiteturas CISC.

Numa arquitetura “*load-store*” as operações de carga (*load*) e armazenamento (*store*) envolvem transferências de dados entre os registradores da CPU e a memória e as instruções aritméticas e lógicas geralmente operam nos registradores, e não diretamente na memória (*register to register*) o que aumenta a necessidade de registradores mas libera transistores que podem ser usados nestes registradores adicionais.

Já na arquitetura “*register-memory*” as operações aritméticas e lógicas podem ser realizadas diretamente na memória, sem a necessidade de transferir dados para registradores primeiro e o chip pode ter instruções que operam diretamente em dados localizados na memória (*register to register, register to memory ou memory to memory*), evitando a etapa de carga prévia o que diminui a necessidade de registradores mas faz uso de transistores adicionais para o armazenamento nas instruções complexas.

Uma diferença imediata dessa característica é que a arquitetura do chip é simplificada numa arquitetura RISC e o hardware se torna mais eficiente e de desempenho mais consistente para efetuar operações lógicas, isso por outro lado pode exigir mais instruções para realizar um processo com transferência frequente entre a memória e os registradores, o acesso a memória pode ser também mais lento que arquiteturas “*register-memory*”.

Uso frequente de pipelines. Processadores RISC tipicamente lançam mão de pipelines, uma técnica onde múltiplas instruções são executadas simultaneamente em diferentes estágios do pipeline, o que aumenta a eficiência de execução.

Instruções de tamanho fixo. Instruções RISC tipicamente têm um tamanho fixo de comprimento (*fixed-length*) o que torna a decodificação e execução também mais rápida.

Necessidade reduzida de acesso a memória. Processadores RISC reduzem a quantidade de acesso a memória necessário pelo armazenamento em cache de instruções mais frequentes o que permite ao processador acessá-las mais rapidamente sem necessidade de recorrer a memória principal.

Número reduzido de instruções de controle de fluxo. Processadores RISC tipicamente necessitam de menos instruções de controle e fluxo, o que reduz a complexidade do conjunto de instruções e aumenta a performance.

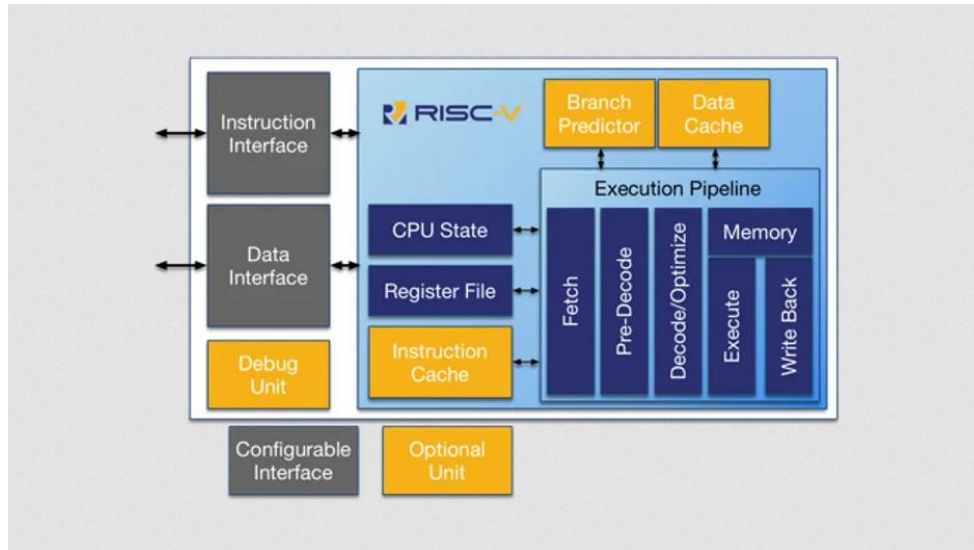


Diagrama de blocos da arquitetura RISC-V. Adaptado de wevolver.com e roallogic.com

6. Estado do mercado e perspectivas

6.1. O caso Nvidia

Nos últimos 5 anos, mais notadamente do período 2020 a 2023, a Nvidia se tornou de uma empresa voltada a fabricação de placas gráficas para o público gamer numa gigante de IA avaliada em mais de um trilhão de dólares. Esta subida sem precedentes de uma fabricante de chips se deu pela evolução, implementação e ampla adoção da arquitetura CUDA para aceleração de fluxos de trabalho de IA.

Presente nos frameworks mais populares entre eles PyTorch e TensorFlow e nas placas tanto do segmento doméstico quanto de servidores, a arquitetura fechada rapidamente se tornou e ainda hoje é o padrão de hardware acelerador de IA. Isso fez com que ela atingisse a marca histórica de ações, mas esse domínio também gerou efeitos adversos:

- Acendeu um alerta de dependência tecnológica de chips em empresas como Microsoft, Amazon e Google, que sem exceção utilizam estas placas do segmento de servidor tanto como aluguel direto para cargas de trabalho quanto para habilitar serviços de IA para seus clientes.
- Ocasinou uma escassez de produtos tanto da linha doméstica quanto em especial, da linha de servidores, que está com produção completamente vendida até o fim de 2024.
- Gerou um movimento do governo americano no sentido de impedir suas exportações para uma lista de países não alinhados, entre eles a China, por ver nesta arquitetura um bem de valor estratégico para a segurança nacional com a própria supremacia em IA sendo vista como ativo estratégico de defesa.

Estes fatores fizeram com que algumas destas empresas iniciassem o desenvolvimento de chips próprios para tarefas de IA como os chips Maia e Cobalt (um tipo de ARM) da Microsoft para a nuvem

Azure, a linha Tritanium da Amazon para a nuvem AWS e as unidades TPU (um ASIC até onde se sabe) da Google para sua nuvem GCP.

6.2. A importância de uma plataforma aberta

O movimento da câmara de comércio americana em especial, validou as preocupações e posterior relocação da fundação RISC-V para fora dos Estados Unidos, pois a arquitetura RISC-V que é flexível já está sendo portada para uma próxima geração de aplicações de IA [2][3][4], o que torna absolutamente plausível que estes mesmos chips e a arquitetura em si entrem nesse conjunto de proibições tanto de produção quanto exportação dadas as declarações recentes da câmara de comércio americana na pessoa da secretária de comércio Gina Raimondo quanto os chips Nvidia:

"Nós (os EUA) não permitiremos que a China tenha acesso a esses chips, ponto final (...). Nós negaremos a eles o acesso a toda a nossa tecnologia de ponta.

"Eu sei que há certos CEOs de companhias de chips na plateia que não gostaram do que fiz, porque estão perdendo dinheiro (...). É a vida. Proteger nossa segurança nacional é mais importante do que lucro a curto prazo."

E sobre manobras recentes da Nvidia no sentido de contornar essas restrições:

"Se você (Huang) redesenhar um chip, contornando um limite particular que permita a eles (a China) desenvolver soluções de IA, eu vou restringi-lo no dia seguinte."

Essa disputa em torno dos chips aceleradores de IA e o próprio campo de IA ter sido colocado como estratégico pelos Estados Unidos e passível de restrições deixa clara a importância de uma arquitetura aberta, flexível e de alto desempenho para produção de chips tanto de uso geral quanto específico.

7. Provisionamento de um Ambiente de desenvolvimento

O acesso a hardware RISC-V para experimentação não é comum e plataformas x86_x64 são bastante populares, isso motivou a inclusão de uma sessão com passos para um ambiente de experimentação possível de ser utilizado em computadores com sistema Ubuntu ou Debian ou mesmo sistemas Windows 10+ com o recurso WSL (Windows subsystem for Linux) ativado. É essencial que a tecnologia de virtualização de hardware disponível no processador esteja ativada na BIOS (pode estar sob os nomes SVM, Virtualization, Virtualization Technology).

7.1. Passos preliminares (Windows somente)

Ativar e atualizar o WSL, setar o WSL para utilizar a versão 2

```
wsl --install  
wsl --update  
wsl --set-default-version 2
```

Escolher a distribuição Linux desejada no WSL2

Para Ubuntu:

```
wsl --install Ubuntu  
wsl --setdefault Ubuntu
```


Para Debian:

```
wsl --install Debian  
wsl --setdefault Debian
```

Para iniciar o WSL digite no prompt:

```
wsl
```

7.2. Dependências do toolchain

Bibliotecas necessárias para o toolchain e compilador

```
sudo apt-get install autoconf automake autotools-dev curl python3 libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc zlib1g-dev libexpat-dev
```

Device Tree Compiler

```
sudo apt install device-tree-compiler
```

7.3. Build do RISC-V Toolchain, emulador e proxy

Crie uma pasta para o clone e build das ferramentas

```
mkdir RISCv  
cd RISCv
```

Clone do toolchain, do proxy de Kernel e do emulador

```
git clone --recursive https://github.com/riscv/riscv-gnu-toolchain  
git clone https://github.com/riscv/riscv-pk  
git clone https://github.com/riscv/riscv-isa-sim
```

Setagem da variável de ambiente RISCv (preferencialmente utilize o arquivo .bashrc para mantê-la ativa)

```
export RISCv=/caminho/de/home/para/RISCv  
export PATH=$PATH:$RISCv/bin
```

Build do toolchain (a partir da pasta RISCv)

```
cd riscv-gnu-toolchain  
mkdir build  
cd build  
../configure --prefix=$RISCv --with-arch=rv32i  
make
```

Build to proxy para o Kernel (a partir da pasta riscv-gnu-toolchain)

```
cd ../riscv-pk  
mkdir build  
cd build  
../configure --prefix=$RISCv --host=riscv32-unknown-elf  
make  
make install
```

Build do emulador Spike (a partir da pasta riscv-pk)

```
cd ../riscv-isa-sim  
mkdir build
```

```
cd build
../configure --prefix=$RISCV --enable-histogram
make
make install
```

7.4. Teste da instalação e IDE's

Programa simples de teste (a partir da pasta RISCV)

```
cd ..
touch hello.c
```

Edite o arquivo (utilizando o nano por exemplo)

```
nano hello.c
```

Adicione o seguinte conteúdo

```
#include <stdio.h>
int main() {
    printf("Hello world!\n");
}
```

Salve e execute na pasta o seguinte comando

```
riscv32-unknown-elf-gcc hello.c /caminho/para/riscv-isa-sim/build/spike --isa=RV32IMAC
/caminho/para/riscv-pk/build/pk -o hello
```

Para executar programas usando Spike e proxy

```
spike pk hello
```

Para executar programas bare-metal

```
spike hello
```

8. Conclusão

RISC-V é uma arquitetura ISA em desenvolvimento, aberta e flexível para implementação de hardware e software de uso geral e especializado. Traz as vantagens de arquiteturas RISC que tornaram as arquiteturas ARM as mais produzidas na história e mais populares para dispositivos móveis, como eficiência energética e complexidade baixa de hardware comparada a arquiteturas CISC enquanto mantém proteções para mantê-la livre no sentido de royalties de hardware e software e relativamente segura do ponto de vista de embargos enquanto pesquisas de sua implementação em soluções voltadas a IA avança.

9. Referências bibliográficas

9.1. Artigos

[1] PATTERSON, David; DITZEL, David

The case for the reduced instruction set computer. CS at UC Berkeley, 1980.

DOI:10.1145/641914.641917

<https://inst.eecs.berkeley.edu/~n252/paper/RISC-patterson.pdf>

[2] WANG, Jiulong; WU, Ruopu; CHEN, Guokai; CHEN, Xuhao; LIU, Boran; ZONG, Jixiang; ZHAO, Di

RISC-V Toolchain and Agile Development based Open-source Neuromorphic Processor. 2022.

<https://arxiv.org/abs/2210.00562>

[3] CONTI, Francesco; PAULIN, Gianna; GAROFALO, Angelo; ROSSI, Davide; DI MAURO, Alfio; RUTISHAUSER, Georg; OTTAVI, Gianmarco; EGGIMANN, Manuel; OKUHARA, Hayate; BENINI, Luca

Marsellus: A Heterogeneous RISC-V AI-IoT End-Node SoC with 2-to-8b DNN Acceleration and 30%-Boost Adaptive Body Biasing. 2023.

<https://arxiv.org/abs/2305.08415>

[4] RUTISHAUSER, Georg; HUNZIKER, Robin; DI MAURO, Alfio; BIAN, Sizhen; BENINI, Luca; MAGNO, Michele

ColibriES: A Milliwatts RISC-V Based Embedded System Leveraging Neuromorphic and Neural Networks Hardware Accelerators for Low-Latency Closed-loop Control Applications. 2023.

<https://arxiv.org/abs/2302.07957>

9.2. Referências gerais: Definição

RISC-V is an open standard Instruction Set Architecture (ISA) enabling a new era of processor innovation through open collaboration

<https://riscv.org/>

RISC-V Instruction Set Manual

<https://github.com/riscv/riscv-isa-manual>

RISC-V vs ARM: A Comprehensive Comparison of Processor Architectures

<https://www.wevolver.com/article/risc-v-vs-arm-a-comprehensive-comparison-of-processor-architectures>

John Cocke

[https://en.wikipedia.org/wiki/John_Cocke_\(computer_scientist\)](https://en.wikipedia.org/wiki/John_Cocke_(computer_scientist))

Reduced instruction set computer

https://en.wikipedia.org/wiki/Reduced_instruction_set_computer

Complex Instruction Set Computer

<https://pt.wikipedia.org/wiki/CISC>

RISC-V

<https://en.wikipedia.org/wiki/RISC-V>

CISC and RISC Processor Architecture

<https://electronicsforu.in/cisc-and-risc-processor-architecture/>

Creative Commons

https://en.wikipedia.org/wiki/Creative_Commons

9.3. Referências gerais: Apple, Qualcomm e nuvens com chips customizados

Apple Exploring RISC-V, Hiring RISC-V 'High Performance' Programmers

<https://www.tomshardware.com/news/apple-looking-for-risc-v-programmers>

Apple's new M3 chips have big GPU upgrades focused on gaming and pro apps

<https://www.theverge.com/2023/10/30/23938676/apple-m3-chip-gpu-upgrade-hardware-accelerated-ray-tracing-gaming-specs-release-date>

What is RISC-V, and why we're unlocking its potential

<https://www.qualcomm.com/news/onq/2023/09/what-is-risc-v-and-why-were-unlocking-its-potential>

Microsoft is finally making custom chips — and they're all about AI

<https://www.theverge.com/2023/11/15/23960345/microsoft-cpu-gpu-ai-chips-azure-maia-cobalt-specifications-cloud-infrastructure>

Amazon unveils new chips for training and running AI models

<https://techcrunch.com/2023/11/28/amazon-unveils-new-chips-for-training-and-running-ai-models/>

Enabling next-generation AI workloads: Announcing TPU v5p and AI Hypercomputer

<https://cloud.google.com/blog/products/ai-machine-learning/introducing-cloud-tpu-v5p-and-ai-hypercomputer>

9.4. Referências gerais: O caso Nvidia e declarações da câmara de comercio americana

Nvidia: The chip maker that became an AI superpower
<https://www.bbc.com/news/business-65675027>

Why Nvidia is suddenly one of the most valuable companies in the world
<https://www.washingtonpost.com/technology/2023/05/25/nvidia-ai-stock-gpu-chatbots/>

Após veto dos EUA, Nvidia prepara lançamento de novos chips para China
<https://investnews.com.br/negocios/apos-veto-dos-eua-nvidia-prepara-lancamento-de-novos-chips-para-china/>

EUA para Nvidia: "pare de vender para a China!"
<https://meiobit.com/462951/nvidia-puxao-orelha-dpto-comercio-eua-vendas-ia-china/>

9.5. Referências técnicas

Spike RISC-V ISA Simulator: Spike, the RISC-V ISA Simulator, implements a functional model of one or more RISC-V harts. It is named after the golden spike used to celebrate the completion of the US transcontinental railway.
<https://github.com/riscv-software-src/riscv-isa-sim>

RISC-V: A Baremetal Introduction using C++.
<https://philmulholland.medium.com/modern-c-for-bare-metal-risc-v-zero-to-blink-part-1-intro-def46973cbe7>

RISC-V Guide: A guide covering the RISC-V Architecture including the applications, libraries and tools that will make you a better and more efficient developer with the RISC-V ISA.
<https://github.com/mikeroyal/RISC-V-Guide>

Compiling C/C++ Code for RISC-V
<https://www.linkedin.com/pulse/compiling-cc-code-risc-v-sibraintech/>

Spike Tutorial for Linux
<http://acsa.ustc.edu.cn/ics/download/riscv/spike-tutorial-linux.pdf>

How to Enable Virtualization?
<https://www.simplilearn.com/enable-virtualization-windows-10-article>