Date: Oct 18, 2022

Teoria

FPGA Design Workshop

Introdução sobre FPGAs

A linha entre engenharia de software e hardware é mais tênue do que parece. Dispositivos chamados field-programmable gate arrays (FPGAs), cujos atributos físicos podem ser manipulados através do uso de linguagens de descrição de hardware (HDLs), preenchem a lacuna entre programação de software e programação de hardware.

Abordaremos o básico das FPGAs neste artigo, como como eles funcionam e por que são usados.

O que é FPGA e por que é usado?

Uma FPGA é um circuito integrado (IC) equipado com blocos lógicos configuráveis (CLBs), ou também chamados de elementos lógicos (LEs) a depender da tecnologia do fabricante, e outros recursos que podem ser programados e reprogramados por um usuário. O termo “programável em campo” indica que as habilidades da FPGA são ajustáveis e não são programadas pelo fabricante como em outros CIs.

FPGAs são circuitos integrados que se enquadram no guarda-chuva de dispositivos lógicos programáveis (PLDs). A funcionalidade fundamental da tecnologia FPGA é construída em hardware adaptável, que tem a capacidade única de ser modificado após a fabricação. Matrizes de blocos de hardware, cada um configurável, podem ser conectados conforme necessário, permitindo que arquiteturas específicas de domínio altamente eficientes sejam construídas para qualquer aplicação.

Essa adaptabilidade de hardware é um diferencial exclusivo de CPUs e GPUs.

As CPUs são altamente flexíveis, mas seu hardware subjacente é fixo. Uma vez que uma CPU é fabricada, o hardware não pode ser alterado. Ele depende do software para dizer qual operação específica (função aritmética) executar, em quais dados na memória. O hardware deve ser capaz de realizar todas as operações possíveis, que são chamadas usando instruções de software e geralmente só podem executar uma instrução por vez. FPGAs, em contraste, podem processar grandes quantidades de dados em paralelo. O benefício do hardware adaptável sobre as CPUs varia de acordo com o aplicativo, dependendo em grande parte da natureza da computação e de sua capacidade de paralelização, mas não é incomum ver uma melhoria de desempenho de 20 vezes versus uma implementação de CPU de funções que podem ser altamente paralelizadas.

As GPUs abordam uma grande desvantagem das CPUs – a capacidade de processar uma grande quantidade de dados em paralelo e operar em conjuntos de dados muito amplos. Fundamentalmente, as GPUs são semelhantes à CPU porque possuem hardware fixo e operam usando instruções de software. Uma única instrução pode processar milhares de dados ou mais, tornando-os adequados para domínios específicos, como aceleração gráfica, computação de alto desempenho, processamento de vídeo, certas formas de aprendizado de máquina e muito mais. Fundamentalmente, no entanto, a arquitetura básica e o fluxo de dados de uma GPU são corrigidos antes da fabricação.

As FPGAs oferecem aos programadores e designers a capacidade de adaptar e atualizar a arquitetura de computação com maior flexibilidade — resultando em arquiteturas específicas de domínio que são mais específicas para seus requisitos. FPGAs não são novidade, mas estão se tornando mais necessários devido à velocidade de inovação em áreas como inteligência artificial. A primeira FPGA de uso comercial foi inventada em 1985 pela Xilinx, que domina 60%-70% do mercado atual de FPGAs.

Usos e aplicações de FPGA

As aplicações para FPGAs são vastas. Hoje, são usados em data center, engenharia aeroespacial, defesa, inteligência artificial (IA), IoT industrial (internet das coisas), redes com e sem fio, automotivo e inúmeras outras indústrias. Esses dispositivos geralmente estão em ambientes onde os usuários precisam de informações em tempo real. Por exemplo, uma câmera de segurança residencial precisa transmitir imagens instantâneas para os dispositivos inteligentes do proprietário — com alta resolução e latência mínima. Essas expectativas só aumentarão à medida que os consumidores se tornarem mais dependentes de informações instantâneas na ponta dos dedos.

FPGAs também auxiliam na aceleração de funções que de outra forma seriam feitas em software. Isso torna as FPGAs uma ferramenta útil para descarregar tarefas de alto desempenho, como inferência de redes neurais profundas (DNN) para inteligência artificial.

FPGAs e aceleração de hardware

A arquitetura das FPGAs as torna uma solução eficiente para aceleração de hardware. Dispositivos como ASICs e GPUs usam um método antiquado de pular entre programação e memória. Eles também não acomodam aplicativos em que são necessárias informações em tempo real, pois a alta quantidade de energia necessária para tarefas de armazenamento e recuperação causa atrasos no desempenho.

Ao contrário dos ASICs e GPUs, as FPGAs não precisam alternar entre memória e programação, o que torna o processo de armazenamento e recuperação de dados mais eficiente. E como a arquitetura FPGA é mais flexível, você pode personalizar a quantidade de energia que deseja que um FPGA utilize para uma tarefa específica.

Essa flexibilidade pode ajudar a descarregar as tarefas que consomem energia para uma, ou para várias FPGAs, provenientes de uma CPU convencional ou outro dispositivo. E como FPGAs podem ser reprogramadas, você pode implementar atualizações e ajustes em um sistema de aceleração de hardware.

Como funciona a programação FPGA?

A programação de FPGA usa um HDL para manipular circuitos dependendo de quais recursos você deseja que o dispositivo tenha. O processo é diferente de programar uma GPU ou CPU, pois você não está escrevendo um programa que será executado sequencialmente. Em vez disso, você está usando um HDL para criar circuitos e alterar fisicamente o hardware, dependendo do que você deseja que ele faça.

O processo é semelhante ao software de programação em que você escreve um código que é transformado em um arquivo binário e carregado no FPGA. Mas o resultado é que o HDL faz alterações físicas no hardware, em vez de otimizar estritamente o dispositivo para executar o software.

Um programa em um FPGA reúne elementos de baixo nível, como portas lógicas e blocos de memória, que trabalham em conjunto para concluir uma tarefa. Como você está manipulando o hardware desde o início, os FPGAs permitem uma grande flexibilidade. Você pode ajustar funções básicas, como memória ou uso de energia, dependendo da tarefa.

Linguagens usadas para programar FPGAs

Pode parecer que as FPGAs estejam principalmente no domínio dos projetistas de chips, em vez de engenheiros especializados em desenvolvimento de software. Afinal, a maioria das HDLs usadas para escrever código FPGA são linguagens de baixo nível com as quais os engenheiros de hardware provavelmente estão mais familiarizados do que os engenheiros de software. Mas alguns HDLs são mais parecidos com linguagens de software comuns do que você imagina.

Quando usamos a palavra “programação” em relação aos FPGAs, não é exatamente o mesmo que criar software, devido à forma como o programa é configurado e como é executado. Mas o uso desse termo abrange a ideia de que escrever e executar código FPGA é semelhante ao processo de criação de um algoritmo de software. A velha maneira de pensar era que FPGAs só poderiam ser programados por engenheiros de hardware projetando no nível do circuito. Hoje, não é mais assim.

Com a ajuda de plataformas de software unificadas, os desenvolvedores de software podem usar suas linguagens preferidas para programar FPGAs sem serem treinados em HDLs. Isso elimina o estresse de ter que migrar para uma nova linguagem de programação e pode ajudar os desenvolvedores de software a se concentrarem em conceitos em vez de hardware. Essas plataformas funcionam essencialmente traduzindo linguagens de alto nível para linguagens de baixo nível para que um FPGA possa executar a função desejada.

As linguagens que podem ser usadas com plataformas de software unificadas para programar FPGAs incluem:

AI Framework como TensorFlow e Pytorch - Com o Vitis AI, os cientistas de IA agora podem pegar diretamente seus modelos de aprendizado profundo treinados do TensorFlow ou Pytorch e compilar para aceleração de FPGA. Isso não apenas elimina a necessidade de programação de hardware de baixo nível, mas também alcança um tempo de compilação extremamente rápido em minutos, combinando com a experiência típica de compilação de software usando CPUs e GPUs.

C e C++ - Graças à síntese de alto nível (HLS), linguagens baseadas em C agora podem ser usadas para design de FPGA. Especificamente, o compilador Xilinx® Vivado® HLS fornece um ambiente de programação que compartilha tecnologia chave com processadores padrão e especializados para a otimização de programas C e C++. Isso permite que os engenheiros de software otimizem o código sem ter que lidar com o obstáculo de espaço de memória limitado ou recursos computacionais.

Python - Os designers podem usar a linguagem e as bibliotecas Python para criar aplicativos de alto desempenho e programar FPGAs com PYNQ - um projeto de código aberto da Xilinx que facilita o uso das plataformas Xilinx.

Há também uma série de HDLs convencionais que são usados principalmente na programação de FPGA hoje. Aqui está um breve resumo sobre seus nomes e atributos principais:

Lucid - Essa linguagem foi feita especificamente para FPGAs e supera algumas das armadilhas de linguagens mais arcaicas, como o Verilog.

VHDL - Um acrônimo para VHSIC (Very High Speed Integrated Circuits) Hardware Description Language, esta linguagem apareceu pela primeira vez na década de 1980 e foi baseada em Ada e Pascal.

Verilog - O primeiro HDL já criado, o Verilog hoje é usado principalmente para análise e verificação de testes. O núcleo desta linguagem foi baseado em C.

Nota: esse texto foi uma tradução livre do artigo "Programming an FPGA: An Introduction to How It Works", publicado pela Xilinx e que pode ser acessado em: https://www.xilinx.com/products/silicon-devices/resources/programming-an-fpga-an-introduction-to-how-it-works.html

Outras recomendações de leitura:

- **Greg Martin**. "Adaptive Computing: Technology Overview". Xilinx (2021). https://www.xilinx.com/content/dam/xilinx/publications/technology-briefs/adaptive-computing-technology-overview.pdf

- **Javier Serrano**. "Introduction to FPGA Design", CERN (2009). http://cds.cern.ch/record/1100537