FACULDADE DE TECNOLOGIA DA UNIVERSIDADE ESTADUAL DE CAMPINAS

Implementação das funções Low-PHY da estrutura de rede do 5G em um SmartNIC acelerado por FPGA utilizando a linguagem e recursos da OpenCL (Open Computing Language)

ALUNO: YOUSSEF HASSAN GHARIB

ORIENTADOR: PROFESSOR DR. RANGEL ARTHUR

COAUTOR: GABRIEL PEDRO PAIÃO

Palavras-chave: FPGA, 5G, Low-PHY, OpenCL.

LIMEIRA

2024

1. BREVE DESCRIÇÃO

O projeto consistiu no teste das funções da camada Low-PHY da unidade distribuída, do inglês Distributed Unit (DU), do 5G em um acelerador baseado em hardware, aproveitando os recursos da Open Computing Language (OpenCL), compatível com ambientes computacionais heterogêneos, permitindo o processamento paralelo.

Os sinais de radiofrequência (RF) chegam ao FPGA e são processados conforme as funções *Low-PHY*. No *downlink*, o sinal chega no domínio da frequência e sofre a transformada rápida inversa de fourier, do inglês *Inverse Fast Fourier Transform (IFFT)*, e é inserido um prefixo cíclico, do inglês *Cyclic Prefix (CP)*, para evitar a interferência intersimbólica. No *Uplink*, o sinal no domínio do tempo tem o prefixo cíclico retirado e passa pela transformada rápida de fourier, do inglês *Fast Fourier Transform (FFT)*, que o deixa no domínio da frequência.

Como a linguagem de programação OpenCL permite que apenas um programa escrito pelo host seja executado em diferentes plataformas heterogêneas, foi pensado em fazer com que certas funções fossem executadas no hardware acelerador, aproveitando do seu poder de processamento paralelo, e outras no CPU, diminuindo a carga nos processadores deste. Assim, atingindo uma melhor performance geral do sistema.

2. OBJETIVOS

Propor um acelerador programável baseado em hardware (FPGA) que acelera a função *Low-PHY* na Unidade Distribuída (DU) da arquitetura de rede da Nova Rádio (NR) do 5G, utilizando os recursos da linguagem OpenCL.

Apresentar uma reformulação no sistema de processamento de sinais e transmissão de dados na estrutura do 5G, que oferecerá um processamento mais rápido dos sinais IQ (em fase e quadratura) no *midhaul*, contribuindo com o desenvolvimento das características marcantes do 5G: eMBB (*Enhanced Mobile Broadband*, Banda larga móvel aprimorada), URLLC (*Ultra-Reliable Low Latency Communication*, Comunicação de baixa latência ultra confiável) e mMTC (*Massive Machine Type Communication*, Comunicação do tipo de máquina massiva). Isso sem nenhuma modificação de caráter significativo no sistema completo.

Contribuir para o início do desenvolvimento da área de redes 5G e o uso do OpenCL para programação de aceleradores baseados em hardware, na divisão de Telecomunicações da Faculdade de Tecnologia, com um projeto de pesquisa pioneiro utilizando essas duas tecnologias.

3. RESULTADOS OBTIDOS

Conforme mostrado nas Figuras 1 e 2, a análise de desempenho da IFFT usando as ferramentas do clFFT tanto em CPU quanto em GPU mostra diferenças significativas, destacando os potenciais benefícios de descarregar certas tarefas para um hardware mais especializado. Os testes realizados em um CPU Intel(R) Core(TM) i7-10510U e em uma GPU NVIDIA GEFORCE® MX110 demonstram que a GPU supera consistentemente a CPU em termos de tempo de execução, independentemente do tamanho do *batch* (lote), que define quantas transformadas serão calculadas de uma vez. Isso é particularmente notável, dado que a CPU normalmente é responsável por gerenciar uma multiplicidade de operações em segundo plano simultaneamente, o que pode impactar sua capacidade de executar tarefas computacionalmente intensivas de maneira eficiente.

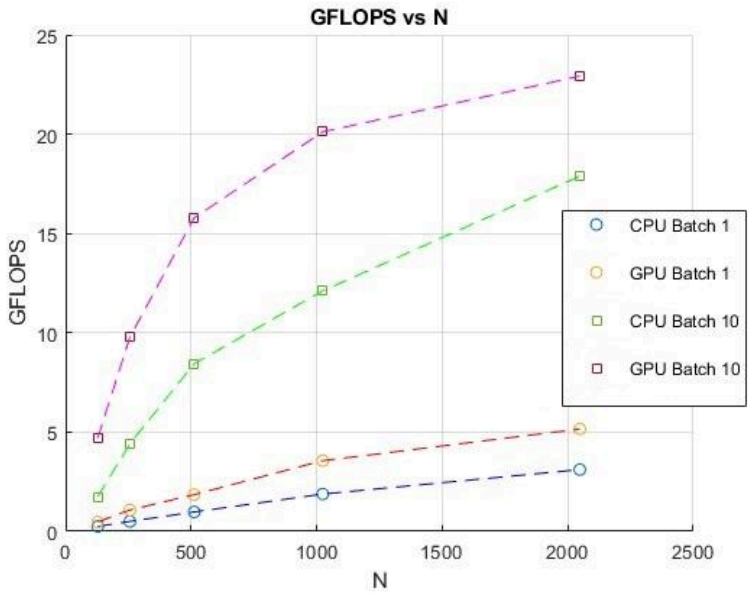

Figura 1 -IFFT GFLOPS VS TAMANHO N para CPU e GPU com tamanhos *Batch Size* 1 e 10.

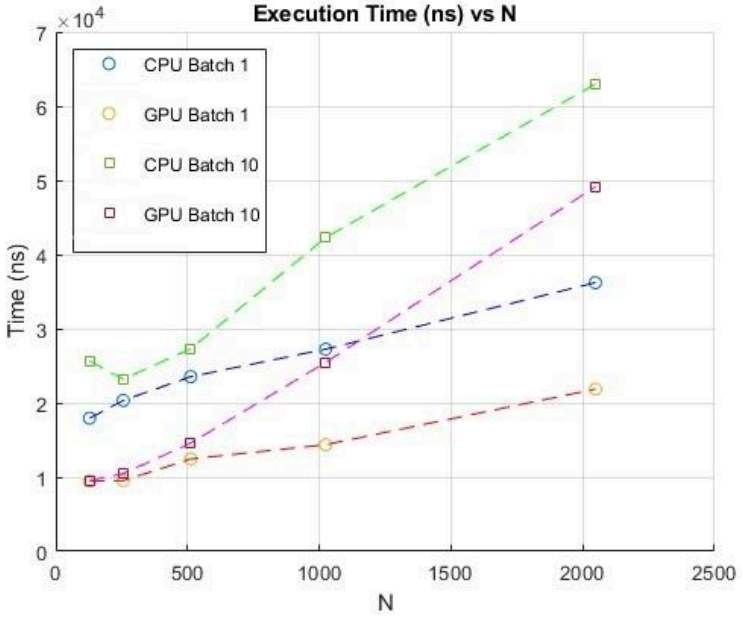

Figura 2 - TEMPO DE EXECUÇÃO IFFT VS TAMANHO N para CPU e GPU com tamanhos *Batch Size* 1 e 10.

Para esta implementação, o número de pontos utilizados são 128, 256, 512, 1024 e 2048. Todos sendo potência de dois, considerando a aplicação do algoritmo radix-2, ou uma combinação de radix-2 com radix-4.

Para o tamanho de *batch* 1, a GPU mostra melhorias notáveis de desempenho em relação à CPU. Por exemplo, com N = 128, a GPU atinge um tempo de execução de 9.461 ns em comparação aos 17.944 ns da CPU. À medida que o valor de N aumenta, a GPU continua a manter sua vantagem, com o tempo de execução para N = 2048 sendo de 21.859 ns na GPU contra 36.211 ns na CPU. As métricas de GFLOPS também apoiam esses resultados, onde a GPU consistentemente alcança maiores GFLOPS em todos os valores testados de N. Isso

indica que a GPU é capaz de lidar com mais operações de ponto flutuante por segundo, um fator crítico em aplicações de computação de alto desempenho.

Ao aumentar o tamanho do *batch* para 10, tanto a CPU quanto a GPU exibem tempos de execução aumentados, mas a GPU ainda mantém uma vantagem clara. Por exemplo, em N = 128, o tempo de execução da GPU é de 9.528 ns, enquanto a CPU leva 25.655 ns. Mesmo em N = 2048, a GPU completa a tarefa em 49.144 ns em comparação aos 62.977 ns da CPU. Notavelmente, os GFLOPS para a GPU com tamanho de lote 10 chegam a 22,92 para N = 2048, demonstrando a superior capacidade da GPU de escalar o desempenho com cargas de trabalho aumentadas.

Uma observação chave dos testes é a relação entre GFLOPS e tempo de execução. Apesar do aumento no tempo de execução com tamanhos de *batch* maiores, os GFLOPS alcançados tanto pela CPU quanto pela GPU também aumentam, com a GPU mostrando uma melhoria mais pronunciada. Isso sugere que, embora operações individuais possam demorar um pouco mais com tamanhos de lote maiores, o rendimento geral (número de operações concluídas por segundo) melhora significativamente. Isso é particularmente relevante para aplicações que requerem alto rendimento computacional, como processamento de sinal em redes 5G.

No contexto das funções de 5G, como a IFFT no OFDM (parte das funções Low-PHY), descarregar a computação para um dispositivo OpenCL como uma GPU ou FPGA poderosa pode melhorar o desempenho geral do sistema. Ao reduzir a carga computacional na CPU, que já está lidando com outras funções críticas, o sistema pode alcançar maior eficiência e tempos de processamento mais rápidos. Essa abordagem está alinhada com a tendência mais ampla de utilizar um hardware especializado para lidar com tarefas específicas em sistemas de alto desempenho e em tempo real, garantindo a utilização otimizada dos recursos e melhores métricas de desempenho.