UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE TELEINFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

PEDRO LUCAS FALCÃO LIMA

Projeto de um IP Soft Core para Detecção de Ataques DDoS

Fortaleza, Ceará

2017

PEDRO LUCAS FALCÃO LIMA

PROJETO DE UM IP SOFT CORE PARA DETECÇÃO DE ATAQUE DDOS

Monografia submetida à Coordenação de Teleinformática e à Coordenadoria do Curso de Bacharelado em Engenharia de Computação da Universidade Federal do Ceará - Campus do PICI, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação.

Área de pesquisa: Utilização de FPGA's

Orientador: PROF. MSC. JARDEL NUNES DA SILVEIRA

Fortaleza
2017

UNIVERSIDADE FEDERAL DO CEARÁ
COORDENAÇÃO DE ENGENHARIA DE COMPUTAÇÃO

PEDRO LUCAS FALCÃO LIMA

Esta Monografia foi julgada adequada para a obtenção do Grau de Bacharel em Engenharia da Computação, sendo aprovada pela Coordenadoria de Teleinformática e pela Coordenadoria do curso de Bacharelado em Engenharia de Computação do Campus do Pici da Universidade Federal do Ceará e pela banca examinadora:

Orientador: Prof. Msc. Jardel Nunes da Silveira
Universidade Federal do Ceará - UFC

Prof. Dr. Jarbas Aryel Nunes da Silveira
Universidade Federal do Ceará - UFC

Prof. Dr. Otávio Alcântara de Lima Júnior
Instituto Federal do Ceará - IFCE

Fortaleza, 18 de Dezembro de 2017

Dedico este trabalho ...

Agradecimentos

“A mente que se abre a uma nova idéia jamais voltará ao seu tamanho original”.

Albert Einstein

Resumo

Este trabalho apresenta...

Abstract

This work presents...

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

Z	variavel aleatoria
$\mathbb{R}$	conjunto dos números reais
t	tempo contínuo
n	tempo discreto
$f(z)$	função densidade de probabilidade
$F(z)$	função de distribuição acumulada
σ	desvio padrão
μ	média ou esperança matemática
$\lvert\cdot\rvert$	operador magnitude
∇	operador gradiente

Lista de Abreviacoes

fdp	Função densidade de probabilidade
fda	Função de distribuição acumulada
EMQ	Erro médio quadrático

CAPÍTULO 1

INTRODUÇÃO

O ataque do tipo DoS (Denial Of Service, em inglês), também conhecido como ataque de negação de serviço, é uma tentativa de fazer com que aconteça uma sobrecarga em um servidor ou computador comum para que recursos do sistema fiquem indisponíveis para seus utilizadores. Esses ataques se tornam cada vez mais comuns, principalmente pelo surgimento de ataques do tipo DDoS(um tipo de ataque DoS de grandes dimensões), por isso, vale ressaltar a importância da detecção desse tipo de ataques. Tendo como principal objetivo detecção de ataques DDoS em tempo real, foi escolhido uma solução de detecção em hardware, pois soluções baseadas em software são ineficientes para aplicações de tempo real, uma vez que essas exigem grande quantidade de ciclos de CPU, utilizados em todo o software. Uma arquitetura de hardware dedicada para o módulo de detecção de ataque DDoS é considerada para implementação em FPGAs. Os recentes dispositivos FPGA oferecem alto desempenho e estão aptos a receber lógicas grandes e complexas. Além disso, os FPGAs oferecem adaptabilidade dinâmica, que é importante para aplicações que requerem mudanças freqüentes em suas configurações, como a detecção de ataques DDoS que evoluem com freqüência. No trabalho será utilizado, uma solução híbrida para detecção de ataques, para a implementação da variável de correlação proposta. Por isso, os módulos do pré-processador e do gerenciador de segurança são implementados separadamente usando software. As máquinas que implementam esses módulos e o FPGA podem comunicar usando as interfaces de E / S de alta velocidade suportadas pelos FPGAs modernos, como PCeI e Gigabit Ethernet. O módulo de detecção de ataque recebe a instância de tráfego do módulo pré-processador. Além disso, ele recebe o perfil normal e um valor limiar do banco de dados do perfil criado pelo gerenciador de segurança. Cada uma das instâncias de tráfego e o perfil padrão são vetores que consistem em três recursos de tráfego. O módulo de detecção de ataque calcula primeiro o NaHiD VERC (variável de correlação de acordo com a principal referência de estudo, “Real-time DDoS attack detection using FPGA”) entre a instância de tráfego de entrada e o perfil normal. O valor de

correlação calculado é comparado com o limite para classificar a ocorrência de tráfego recebido como ataque ou normal. O resultado da classificação é armazenado no banco de dados Log para análise off-line pelo gerenciador de segurança. Além disso, um alarme é gerado no caso de a instância ser classificada como um ataque. A 2/3 nível de arquitetura, para separar controle e computação, a unidade de controle e o datapath são desenvolvidos separadamente. No entanto, eles são tão intrinsecamente dependentes. O datapath consiste nos operadores para os cálculos e os registros para armazenar as entradas e os resultados da computação intermediária. A unidade de controle envia os sinais de seleção do multiplexador usados pelas operações e os registros para selecionar a entrada. Eles são as principais partes do módulo da FPGA para a detecção, porém existem outros elementos que vão ser de suma importância para realizar operações complexas e necessárias.

1.1 Motivação e objetivos

1.2 Contribuicoes

1.3 Producao cientifica

1.4 Organizacao da tese

Capitulo 2: descricao...

Capitulo 3: descricaoo...

Capitulo 4: descricao...

Capitulo 5: descricao...

CAPÍTULO 2

REFERENCIAL TEÓRICO

2.1 Ataques

Ataques à uma rede de computadores sãos ações maliciosas em que softwares são utilizados para de alguma forma prejudicar, interromper uma ação ou invadir uma rede ou máquina, afim de se beneficiar com tal ação. Todavia o ataque, deve ser caracterizado dessa forma uma vez que se comprova que ele não é apenas ameaça e sim um agente malicioso. Por isso, vale ressaltar que o ataque é a ação propriamente dita, entretanto, uma ameaça à um sistema é algo que possa afetar ou atingir o seu funcionamento, operação, disponibilidade e integridade. Podemos dizer que um ataque ocorre quando uma ameaça intencional é realizada 1. Outra definição de ataque seria relacionada à seria o tráfego não desejado, que seria qualquer tipo de tráfego de rede não requisitado e/ou inesperado, cujo único propósito é consumir recursos computacionais da rede, desperdiçar tempo e dinheiro dos usuários e empresas e que pode gerar algum tipo de vantagem ou benefício (lucro) para seus criadores 2.

2.2 DoS

Um dos tipos mais comuns de ataque é o ataque DoS (Denial Of Service), que é um tipo de ataque no qual uma ou mais máquinas buscam afetar uma vítima e tentam evitar que a vítima faça algum tipo de trabalho “útil” 3. Uma característica importante desses tipos de ataques é que o objetivo principal desse tipo de ataque não é a invasão em busca de dados ou informações, mas a negação do serviço que a vítima está utilizando ou oferecendo. Uma das estratégias mais utilizadas para a realização desse tipo de ataque é o envio de múltiplas requisições a vítima em questão, de forma que ela não suporta tantas requisições e começa a negar serviços, que antes ela era capaz de realizar 4.

2.3 DDoS

Segundo 4 DDoS (Distributed Denial of Service) é um tipo de ataque DoS de grandes dimensões, ou seja, que utiliza até milhares de computadores para atacar uma determinada máquina, distribuindo a ação entre elas. Trata-se de uma forma muito utilizada, já que é o tipo de ataque mais comum na internet. Outra definifição seria "Um ataque DDoS usa muitos computadores para lançar um ataque DoS coordenado contra um ou mais alvos. Usando a tecnologia cliente / servidor, o atacante é capaz de multiplicar a eficácia do DoS significativamente, aproveitando os recursos de vários computadores cúmplices involuntários, que servem como plataformas de ataque "5.

2.4 Detecção

A detecção de um ataque DDoS é um trabalho relativamente complexo, uma vez que esse tipo de ataque ocorre em tempo real, e muitas vezes é difícil ser identificado a máquina em que realmente está sendo o atacante principal, por isso utilizamos o conceito de “tráfego não desejado”. Como visto anteriormente, antes da ação do ataque acontecer, existem as ameaças aquela rede ou máquina, por isso tráfegos que não desejados (que podem ser vistos como ameaça anteriormente a ação do ataque) devem ser identificados e o sistema de segurança podem tomar as devidas medidas. Segundo o 6 que é a referência de nosso trabalho, várias técnicas de detecção de ataques DDoS já foram propostas, muitas na área da estatística, porém essas soluções não proporcionam alta precisão de detecção DDoS ao usar um pequeno conjunto de recursos de tráfego, ou seja, com poucos dados de tráfego a detecção se torna inviável.

2.5 Correlação

Uma medida estatística que é muito utilizado em detecções em geral, seria a correlação, que é uma medida que mede o relacionamento entre duas variáveis (similaridade, linearidade e direção). Uma das maiores vantagens da correlação é que ela não necessita de uma grande quantidade de variáveis para chegar a uma conclusão de relacionamento entre dois conjuntos. Portanto para uma detecção de

ataques consistentes é necessária uma medida de correlação efetiva para classificar ataques DDoS em tempo real mesmo quando usa um pequeno número de recursos de trafego. Uma correlação de forma resumida, é um conjunto de cálculos que a partir das variáveis de entrada, retorna o relacionamento entre as variáveis em um valor entre -1 e 1, então para ela ser efetivada basta se ter os valores de entrada e um módulo que realize esses cálculos e retorne o resultado da correlação. Segue abaixo a fórmula da correlação proposta pelo trabalho para ataques DDoS.

$$NaHiD(X,Y) = 1 - \frac{1}{n}\sum_{i=1}^{n} \frac{(|X(i) - Y(i)|)}{||\mu X - sX| - X(i)| + ||\mu Y - sY| - Y(i)|} \tag{2.1}$$

onde

- μX: Média aritmética do objeto de tráfego X.
- μX: Média aritmética do objeto de tráfego Y.
- sX: Desvio padrão do objeto de tráfego X.
- sY: Desvio Padrão do objeto de tráfego Y.

2.6 Soluções em Hardware

Quando é necessário realizar um considerável número de cálculos em um dado sistema, pode-se utilizar duas abordagens soluções utilizando software e soluções utilizando hardware. As utilizações de soluções em softwares possuem duas grandes vantagens: Podem ser utilizadas para soluções de propósitos gerais e são bem precisas em seus resultados. Já as soluções em hardware podem acumular uma lógica grande e complexa, além de ter um alto desempenho 6. Na abordagem de Ataques DDoS, utilizamos um sistema em tempo real, que precisa de velocidade e uma relativa precisão, como os softwares em geral exigem grande quantidade de ciclos de CPU (“É o período de tempo no qual um computador lê e processa uma instrução em linguagem de máquina da sua memória ou a sequência de ações que a CPU realiza para executar cada instrução em código de máquina num programa”7), o que traz uma grande perca em sistemas de tempo real. Entretanto, umas arquiteturas baseadas em solução de hardware com algumas adaptações podem trazer desempenho e precisão uma vez que as medidas e ciclos são implementadas de acordo com o problema específico.

2.7 Técnicas Utilizadas

Nota-se que a correlação proposta por 6, necessita de uma certa precisão, pois a mesma é uma medida de detecção. Para tal é necessário que os resultados de suas operações aritméticas tenham uma certa precisão. Basicamente seria necessário ver cada componente dessa formulação e realizar a implementação em hardware de forma acertiva, para que o sistema posso realizar conclusões e medidas precisas. Serão descritas as operações e as formas em hardware de realizar esses cálculos:

2.7.1 Soma:

A soma pode ser implementada, com componentes do tipo somador, ja bem conhecidos pela comunidade. Vale ressaltar que a soma pode ser utilizada no módulo, média e desvio padrão.

2.7.2 Módulo:

O módulo pode ser facilmente implementador em hardware com componentes do tipo somador. Vale ressaltar que o módulo pode ser utilizado no desvio padrão.

2.7.3 Divisão:

A divisão de dois números quaisquer possuem uma certa complexidade em Hardware, por não utilizar em suas operações apenas o conjunto dos números naturais, tendo que utilizar os conjuntos dos números fracionários. Para isso do utilizado um IP da plataforma VIVADO da Xinlix, chamado Divider Generator, esse IP realiza possue uma implementação de divisão de dois números e retorna um resultado em uma representação de números fracionados com uma precisão escolhida de forma personalizada {Divider Generator v5.1 LogiCORE IP Product Guide}.Esse IP configura a latência (quantos ciclos de clock, são necessários para um termino de operação) da computação da de divisão.

Ponto Fixo

Para representação de números fracionários, tem-se duas abordagens mais conhecidas: Ponto fixo e Ponto Flutuante. A representação em ponto flutuante, divide a palavra binária em duas partes (parte que representa o número inteiro e parte que representa o número decimal), essa divisão é feita dinamicamente, ou seja, uma hora a representação do número inteira tem um certo tamanho, outra hora pode ter outro tamanho, podendo ter uma maior precisão ou não de acordo com o a especificação do problema. Já a representação em ponto fixo divide a palavra binária em duas partes (parte que representa o número inteiro e parte que representa em decimal), porém com a divisão fixa , deixando a precisão sempre estática. A vantagem da representação em ponto fixo é a velocidade no qual essa representação é computada frente a representação em ponto flutuante, gerando um bom desempenho e uma certa precisão nos resultados 8, o que para um problema em tempo real é essencial.

2.7.4 Média aritmética:

A média aritmética é um tipo de soma de variáveis seguido por um divisão, como foi dito uma divisão de dois números em hardware não é algo trivial. Por isso, uma alternativa é utilizar nessa divisão aritmética binária, de forma que com pequenos ajustes em uma operação (ajustes que não influenciem no resultado dessa operação),gere um resultado aproximado. No caso da média , seriam uma operação de soma de todos os elementos em questão, divido pelo número de elementos, segundo 4 a divisão por números pares são aproximadas do resultado apenas pelo deslocamento de algum bit, desprezando a parte fracionada. Vale ressaltar que a Média Aritmética pode ser utilizado no desvio padrão.

2.7.5 Desvio padrão:

O Desvio padrão é a variável em questão subtraída da média, após isso elevado ao quadrado. Após essa série de cálculos, calcula-se a raiz quadrada, tendo então a o desvio padrão. Então é necessário um multiplicador para realizar esse potenciação e uma raíz quadrada. Um multiplicador é facilmente implementado em hardware, pois existem componentes do tipo multiplicador. Já a raiz quadrada sua computação é um pouco mais complexa,por isso é necessário a utilização de um Ip core que faça o

calculo de uma raiz. Para isso pode-se utilizar um IP da plataforma VIVADO da Xinlix, chamado CORDIC CORDIC v6.0 , que realiza implementação de raízes quadradas. Basicamente esse IP recebe uma variável como entrada e em sua saída retorna a raiz aproximada do número de entrada .Esse IP configura a latência (quantos ciclos de clock, são necessários para um termino de operação) da computação de raiz.

2.8 FPGA

Uma solução em Hardware que possui certas especificações, são totalmente voltadas parar sistemas embarcados. E quanto maior a especificação de um dado problema, será mais viável a escolha de que tipo de sistema embarcado que se deve utilizar para acomodar a lógica do problema em questão. Como foi pontuado acima o problema da detecção de ataques em tempo real, requer duas características principais desempenho e precisão. Para realizar esse detecção poderíamos utilizar dois tipos de unidade de processamento mais conhecidas ASICs(Circutos integrados para uma aplicação específica) e as FPGAS(Arranjo de Portas Programáveis em Campo). Porém segundo 6, as FPGAs oferecem adaptabilidade dinâmica, que é importante para aplicações que requerem mudanças freqüentes em suas configurações, como a detecção de ataques DDoS que evoluem com freqüência. Além disso segundo 9 O tempo de projetos baseados em FPGA é normalmente conhecido com muita precisão, o que é essencial para um sistema em tempo real.

Para o módulo de correlação em hardware receber as instâncias de tráfego no trabalho 6 foi implementado um módulo chamado de pré-processador e ao final da computação o módulo de hardware envia para outro módulo que irá realizar algum tratamento no sistema, chamado de gerenciador de segurança. Nota-se que os módulos do pré-processador e do gerenciador de segurança são implementados separadamente usando software. As máquinas que implementam esses módulos e o FPGA podem comunicar usando as interfaces de E / S de alta velocidade suportadas pelos FPGAs modernos, como PCI e Ethernet. O módulo de detecção de ataque recebe a instância de tráfego do módulo pré-processador. Além disso, ele recebe o perfil normal e um valor limiar do banco de dados do perfil criado pelo gerenciador de segurança. Cada uma das instâncias de tráfego e o perfil normal são vetores que consistem em três recursos de tráfego. O módulo de detecção de ataque calcula primeiro o NaHiD VERC entre a instância de tráfego de entrada e o perfil

normal. O valor de correlação calculado é comparado com o limite para classificar a ocorrência de tráfego recebido como ataque ou normal. O resultado da classificação é armazenado no banco de dados Log para análise off-line pelo gerenciador de segurança. Além disso, um alarme é gerado no caso de a instância ser classificada como um ataque.

2.9 FPGA'S Xinlix de série 7

As FPGAs da série Xilinx 7 compreendem quatro famílias de FPGA que abordam uma gama completa de requisitos nos sistemas como: baixo custo, pequenos tamanhos, sensíveis ao custo, aplicações de alto volume de largura para altas conectividades, capacidade lógica e capacidade de processamento de sinal para aplicações de alto desempenho.

Artix (Lowest Power and Cust) , Kintex (Industry's Best Price/Performance), Virtex (Industry's Highest Performance) e ZYNQ (All programmable SoC). A diferença entre essas famílias está basicamente em sua performance e custo , consequentemente para qual finalidade elas deverão ser usadas. Pra corroborar com isso nota que cada família tem variações em relação máxima capacidade dos recursos que ambas possuem, como Células Lógicas, Blocos de RAM , pinos I/O e etc . Cada bloco lógico configurável (CLB) é composto por dois Slices que podem ser do tipo M (SLICEM) , podem ser utilizados para memória e lógica, ou do tipo L (SLICEL), podem ser utilizados apenas para lógica e aritmética. Cada Slice pode ter como recursos gerenciáveis 4 LUTs(Look-up Tables) de 6 entradas , multiplexadores , carry chains e 4 flip-flops/latches. As interfaces I/O, em geral, trabalham pra proporcionar altas velocidades de resposta sem tentar perder integridade no sinal. Além de serem projetadas para diferentes padrões (Voltagens, larguras e protocolos). Nas famílias da 7 series elas podem ser caracterizados por dois tipos : High Range (HR), suportando padrões I/O de tensões de ate 3.3 V, e High Performance (HP), suportando somente padrões I/O de tensões de até 1.8V e projetado para alta performance, por isso dependente da família. Todos os membros das famílias de série 7 tem o mesmo bloco RAM/FIFO(First in first out) , operação totalmente síncrona, muitas opções de configurações (true dual port, simple dual-port, single-port) e etc.

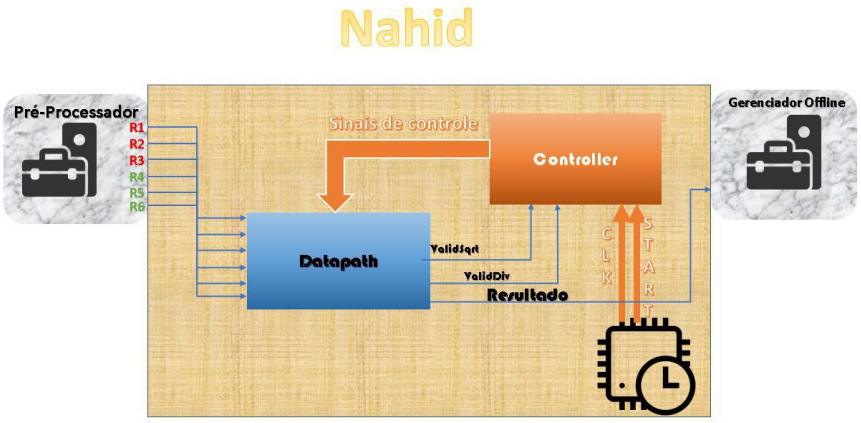
CAPÍTULO 3

MÓDULO NAHID

O módulo de detecção em hardware é basicamente a implementação da formulação dos passos da correlação proposta, utilizando uma FPGA. Para isso, é necessário adequar as operações para os componentes que são disponíveis e escolhidos para realizar a computação de uma dada operação. Além disso, para otimizar o tempo de resposta é necessário uma organização dessas operações em ciclos de clock, para ter uma referência de tempo para cada computação realizada e assim organizar a sequência das operações. Com isso , a arquitetura de detecção de ataques é proposta com o nome de Nahid. Essa arquitetura possui componentes de diversos níveis, nos quais estão dispostos dependendo da operação que esteja sendo feita no momento, vale ressaltar que essas operações são aritméticas, mudança no tamanho da palavra, registro e seleção.

3.1 Módulo Nahid

Nahid é o componente de mais alto nível, sendo ele quem recebe as entradas(perfil normal e instâncias de tráfego em análise) do módulo pré-processador e envia a saída (resultado da análise) para gerenciador de segurança. O Nahid é composto basicamente por dois componentes internos, esses são: Datapath e Controller. Além dos vetores de entrada citados anteriormente, o componente Nahid recebe os sinais de controle clock e start (que serão os sinais que indicaram início da detecção e as mudanças de ciclos). Esse componente abstrai toda a combinação lógica que foi implementado nos componentes internos, porém é de suma importância para realizar a junção de entradas e saídas entre componentes internos, módulos e gerenciadores.



3.1.1 Datapath

O componente responsável por alocar todos os componentes que realizam as operações da detecção é o Datapath. Por isso, o mesmo comporta no mínimo um dos componentes de mais baixo nível, que serão descritos posteriormente. O Datapath recebe os vetores de entradas(perfil normal e instâncias de tráfego em análise) do Nahid, pois esses dados são selecionados, tratados e registrados pelos componentes internos do Datapath. Porém para isso é necessário que seja indicado ao componente quais são as entradas a serem processadas num dado ciclo, por isso o Datapath recebe como entrada seletores provindos do Controller . Entretanto , o Datapath possui saídas para o controller, pois em alguns ciclos é necessário de alguma confirmação de algum componente interno ao Datapath, além disso a saída do sistema será um resultado registrado num componente interno e será em enviado ao componente de mais alto nível (Nahid).

Extend

Esse componente possui a função de estender o tamanho de uma palavra de bits de qualquer tamanho (menor que 23), para uma palavra de mesmo conteúdo com tamanho de 23 bits. Esse componente é de suma importância nas operações de soma, uma vez que o componentes de soma (Adder) possuem entradas de tamanho de 23 bits. O Datapath possui 12 componentes do tipo extend. Esse componente é do tipo de mudança no tamanho da palavra e em um ciclo ele completa sua computação.

Reduce

Esse componente possui a função de reduzir o tamanho de uma palavra de bits de qualquer tamanho (maior que 11), para uma palavra de mesmo conteúdo com tamanho de 11 bits. Esse componente é de suma importância nas operações de multiplicação, uma vez que o componentes de multiplicação(Mul) possuem entradas de tamanho de 11 bits. O Datapath possui 4 componentes do tipo reduce. Esse componente é do tipo de mudança no tamanho da palavra e um em ciclo ele completa sua computação.

Mux

Esse componente possui a função de ter na entrada vários opções, porém a cada iteração, existe um seletor que indica que entrada será utilizada num dado momento, levando para saída do componente essa escolha. Esse componente é de suma importância para reutilização de componentes, deixando a arquitetura mais enxuta e coesa, sendo muito utilizado no Controller, pois para cada ciclo existem determinadas escolhas nesses multiplexadores. É importante ressaltar, que no Datapath existem mux de diversos tamanhos (2 entradas,4 entradas e 6 entradas), isso está diretamente relacionado com o componente que recebe a saída do mux, pois quanto mais ele pode ser utilizado por entradas diferentes, maior será o número de entradas no multiplexador combinado a esse componente. Devido aos multiplexadores terem grande importância na reutilização de componentes, existem duas frentes de alocação de mux no Datapath. Nas operações aritméticas(Multiplicadores e Somadores) e de registro(Registradores). Esse componente é do tipo de seleção de dados e em um ciclo ele completa sua computação.

Mul

Esse componente possui a função de receber duas entradas, e realizar a multiplicação aritmética das mesmas, gerando uma saída do resultado. Foi utilizado um Mul que recebe entradas de 11 bits podendo gerar até 22 bits no resultado dessa multiplicação. Existem 3 multiplicadores no Datapath, que em nosso módulo realizam operação de quadrado de um número, ou seja as multiplicações tem as mesmas entradas num dado momento. Esse componente é do tipo operações aritméticas e em ciclo ele completa sua computação e em um ciclo ele completa sua computação.

Adder

Esse componente possui a função de receber duas entradas, e realizar a soma aritmética das mesmas, gerando uma saída do resultado. Foi utilizado um adder que recebe entradas de 23 bits podendo gerar até 24 bits no resultado dessa soma. Vale ressaltar que o componente possui 4 modos de operação, o que caracteriza diferentes formas de somar as entradas. Esses modos são:

- O modo 0: A soma padrão de dois números positivos.
- O modo 1: A soma de dois números positivos, com o resultado dividido por 4.
- O modo 2: O módulo de dois números positivos
- O modo 3: O módulo de dois números positivos, com o resultado dividido por 4.

Esses modos de operação, são necessários pelos diversos “cálculos” que são necessários na formulação da correlação que o módulo implementa, por isso algum componente é necessário implementar essas adições, sendo escolhido o Somador. Existem 5 somadores no Datapath, que no módulo realizam operações de adição necessárias. Vale ressaltar que o seletor de operação, também é uma entrada do Adder. Esse componente é do tipo operações aritméticas e em um ciclo ele completa sua computação.

Divider

Esse componente possui a função de receber duas entradas, e realizar a divisão aritmética das mesmas, gerando uma saída do resultado. Esse componente possui certas peculiaridades, pois uma divisão em hardware é mais custosa, pela existência de números fracionários nos resultados de divisões decimais, que interferem diretamente nas operações e no resultado do módulo. Por isso existe a necessidade da utilização de alguma representação numérica que possa trazer resultados fracionários, foi utilizado nesse caso representação em ponto fixo como falado anteriormente. O componente Divider utilizado foi o IP core da xilinx “Divider Generator”, que foi configurado da seguinte forma:

- Dividendo: 12 bits
- Divisor: 12 bits

- Quociente: 12 bits
- Parte Fracionária: 8 bits

Em outras palavras, temos 2 entradas de 12 bits e uma saída de 20 bits, porém na representação de ponto fixa, tem-se 12.8(12 números na parte inteira e 8 na decimal). Além dessas instâncias, existem dois sinais de controle no módulo, o" valid in" (responsável por indicar que o componente está pronto para receber as entradas e iniciar a computação) e o "valid out" (responsável por indicar que o componente acabou de realizar a operação por completo e já tem o resultado), esses sinais são de suma importância para a organização do módulo e regulação dos ciclos. Existe apenas um componente do tipo do Divider, que realiza divisões que não possuem divisor diferente de potências de 2 (pois pode-se utilizar mecanismos mais simples, para realizar essas divisões, mantendo o resultado no universo dos inteiros). Esse componente é do tipo operações aritméticas e em 22 ciclos ele completa sua computação.

Sqrt

Esse componente possui a função de receber uma entrada, e realizar a raiz quadrada aritmética da mesma, gerando uma saída do resultado. Esse componente possui resultados inteiros, porém o resultado é um inteiro aproximado para números que não possuem raízes fechadas, pois é possível os números de entradas não serem quadrados perfeitos e o resultado da raíz ser número com casas decimais. Para realizar a busca e aproximação de resposta, é necessário algum algoritmo de busca dessa raiz, gerando um certo custo de ciclos, como no divisor. Para tal o componente Sqrt utilizado foi o IP core da xilinx "Cordic" (função Raíz), que foi configurado da seguinte forma:

- Entrada: 22 bits
- Saída: 12 bits

Logo, temos 1 entrada de 22 bits e uma saída de 12 bits. Além dessas instâncias, existem dois sinais de controle no módulo, o" valid in" (responsável por indicar que o componente está pronto para receber a entrada e iniciar a computação) e o "valid out" (responsável por indicar que o componente acabou de realizar a operação

por completo e já tem o resultado), esses sinais são de suma importância para a organização do módulo e regulação dos ciclos. Existe apenas um componente do tipo do Sqrt, esse componente é do tipo operações aritméticas e em 6 ciclos ele completa sua computação.

Register

Esse componente possui a função de receber uma entrada, e armazenar o valor recebido a partir da próxima subida do clock e atualizar o valor que estiver na entrada na próxima subida do clock. De forma que pelomenos durante um ciclo o registrador terá o valor recebido num dado momento, por isso podemos considerar o conjunto de registradores como a memória do módulo. Os registradores são de suma importância para a realizar as operações, uma vez que não para realizar todos os passos da correlação, “guardam-se” variáveis de uma operação para utilizar nas próximas operações. O Registrador implementado por padrão recebe entradas de 24 bits e a saída tem o mesmo tamanho, porém existem registradores específicos que possuem tamanhos diferentes do padrão, por serem usados para em fins específicos. Além da entrada, o Registrador recebe o sinal “enable” (responsável para habilitar ou não o registro do componente, no próximo ciclo de clock) e “clr” (responsável por zerar o registro do componente, no próximo ciclo de clock). Existem 11 registradores no Datapath, que realizam o regsitro necessários no módulo. Esse componente é do tipo registro e em um ciclo ele completa sua computação.

3.1.2 Controller

O componente responsável por organizar os ciclos das operações do Módulo Nahid é o Controller. Quais componentes do Datapath que serão utilizados num determinado ciclo de computação e em que ciclo teremos os resultados de uma dada operação, são as principais funções desse componente. O Controller recebe o clk (clock do sistema) e o start (sinal que indica o início do módulo) do Nahid, para que o componente garante que o sistema está síncrono. Além dessas entradas, como dito anteriormente, recebe os sinais “valid out” dos componentes Divider e Sqrt. O Controller possue saídas para o Datapath, para indicar a esse componente o que será utilizado num dado ciclo. A estrutura do controller pode ser dividida em dois cases:

1. Case de operações: Esse case é responsável por indicar todas as operações

que serão feitos em todos os ciclos.

2. Case de transições de ciclos: Esse case é responsável por indicar quando haverá as transições de um ciclo para outro.

CAPÍTULO 4

RESULTADOS EXPERIMENTAIS

Tabela 1: Relatório de Utilização

Tipo	Usado	Fixo	Disponível	Utilização%
CLB LUTs*	1302	0	216960	0.60
LUT as Logic	1301	0	216960	0.60
LUT as Memory	1	0	99840	<0.01
LUT as Distributed RAM	0	0		
LUT as Shift Register	1	0		
CLB Registers	1180	0	433920	0.27
Register as Flip Flop	1122	0	433920	0.26
Register as Latch	58	0	433920	0.01
CARRY8	78	0	27120	0.29
F7 Muxes	0	0	108480	0.00
F8 Muxes	0	0	54240	0.00
F9 Muxes	0	0	27120	0.00

CAPÍTULO 5

CONCLUSÃO E TRABALHOS FUTUROS

APÊNDICE A – Título do Apêndice

APÊNDICE B – Exemplo do pacote Algorithm

Algoritmo 1 Estimador ML otimizado.

Inicializar o contador: $j \leftarrow 1$;
Fixar o limiar de variação das estimativas: $e_{\text{out}} \leftarrow 10^{-4}$;
Fixar o número máximo de iterações: $N \leftarrow 1000$;
Computar o ponto inicial: $\hat{\gamma}(0)$;
Determinar o limiar inicial: $e_1 \leftarrow 1000$;
Estabelecer o valor inicial de α: $\hat{\alpha}(0) \leftarrow -10^{-6}$;
enquanto $e_j \geq e_{\text{out}}$ e $j \leq M$ **fazer**
    Solucionar $\hat{\alpha}_j \leftarrow \arg\max_{\alpha} l_1(\alpha; \gamma_{j-1}, \mathbf{z}, n)$;
    Solucionar $\hat{\gamma}_j \leftarrow \arg\max_{\gamma} l_2(\gamma; \alpha_j, \mathbf{z}, n)$;
    $j \leftarrow j + 1$
    Computar o critério de convergência: e_j;
fim enquanto