
# DESENVOLVIMENTO DE UM ACELERADOR INT8 BASEADO EM SYSTOLIC ARRAY EM FPGA INTEGRADO A UM SYSTEM-ON-CHIP

Thiago Fernandes

Itajaí – SC
2025

Thiago Fernandes

# Desenvolvimento de um Acelerador INT8 Baseado em Systolic Array em FPGA Integrado a um System-on-Chip

Estudo acadêmico desenvolvido de forma independente como parte do aprofundamento em arquitetura de computadores, sistemas embarcados e aceleração de hardware em FPGA, no âmbito do curso de Engenharia da Computação da Universidade do Vale do Itajaí – UNIVALI.

Itajaí – SC
2025

# RESUMO

O aumento da complexidade computacional em aplicações modernas tem impulsionado o uso de aceleradores de hardware como alternativa aos processadores de propósito geral, especialmente em domínios caracterizados por elevado paralelismo, como processamento digital de sinais e aprendizado de máquina. Nesse contexto, dispositivos FPGA destacam-se por permitir a implementação de arquiteturas paralelas customizadas com elevada eficiência.

Este estudo apresenta o desenvolvimento de um acelerador de operações aritméticas em ponto fixo de 8 bits (INT8), baseado em uma arquitetura de *systolic array* e implementado em linguagem VHDL. O acelerador foi integrado a um System-on-Chip por meio do barramento Avalon Memory-Mapped, utilizando um wrapper dedicado que encapsula a lógica de computação e permite sua interação com um processador softcore.

A validação funcional e a análise de desempenho foram realizadas por meio de simulação RTL, permitindo a medição de latência e throughput em nível de ciclos de clock. Os resultados demonstram que a arquitetura proposta apresenta baixa latência e elevada taxa de processamento quando comparada, de forma analítica, a um processador de propósito geral executando a mesma operação de maneira sequencial. Dessa forma, o trabalho evidencia o potencial de aceleradores dedicados em FPGA para aplicações computacionalmente intensivas. **Palavras-chave**: FPGA. Acelerador de hardware. Systolic Array. System-on-Chip. VHDL.

# SUMÁRIO


**1 Introdução . . . . . 4**
**2 Fundamentação Teórica . . . . . 5**
2.1 Aceleradores de Hardware em FPGA . . . . . 5
2.2 Arquitetura Systolic Array . . . . . 5
2.3 Aritmética em Ponto Fixo INT8 . . . . . 5
**3 Metodologia . . . . . 6**
**4 Desenvolvimento do Acelerador . . . . . 7**
**5 Arquitetura do System-on-Chip . . . . . 8**
5.1 Componentes do SoC . . . . . 8
5.2 Modelo de Comunicação . . . . . 9
5.3 Fluxo de Dados do Sistema . . . . . 9
**6 Verificação Funcional e Análise de Desempenho . . . . . 10**
6.1 Ambiente de Simulação . . . . . 10
6.2 Inicialização do Sistema . . . . . 10
6.3 Escrita e Controle via Barramento Avalon . . . . . 10
6.4 Registradores Internos do Acelerador . . . . . 11
6.5 Execução do Acelerador . . . . . 11
6.6 Análise de Latência . . . . . 11
6.7 Análise de Throughput . . . . . 12
6.8 Discussão dos Resultados . . . . . 12
**7 Comparação com Processador de Propósito Geral . . . . . 13**
7.1 Modelo Computacional Avaliado . . . . . 13
7.2 Modelo Analítico do Processador . . . . . 13
7.3 Desempenho do Acelerador . . . . . 14
7.4 Comparação de Desempenho . . . . . 14
7.5 Análise de Speedup . . . . . 15
7.5.1 Speedup de Latência . . . . . 15
7.5.2 Speedup de Throughput . . . . . 15
7.5.3 Resumo dos Resultados de Speedup . . . . . 16
7.6 Discussão . . . . . 16
**8 Conclusão . . . . . 17**
**Referências . . . . . 19**

# 1 INTRODUÇÃO

O avanço das aplicações computacionais nas últimas décadas tem intensificado a demanda por maior desempenho e eficiência energética, especialmente em áreas como processamento digital de sinais, visão computacional e aprendizado de máquina. Essas aplicações envolvem, em sua maioria, operações matemáticas repetitivas e altamente paralelizáveis, que não são exploradas de forma eficiente por arquiteturas baseadas exclusivamente em processadores de propósito geral [1].

Nesse cenário, aceleradores de hardware surgem como uma alternativa eficaz, permitindo a execução de tarefas específicas em unidades dedicadas. De acordo com Kuon, Tessier e Rose [2], dispositivos FPGA oferecem uma combinação única de flexibilidade e paralelismo, possibilitando a implementação de arquiteturas customizadas com elevado desempenho.

Arquiteturas baseadas em *systolic arrays* têm sido amplamente utilizadas em aceleradores modernos devido à sua escalabilidade, eficiência computacional e elevado reuso de dados. Essas arquiteturas são particularmente adequadas para operações de multiplicação e acumulação, fundamentais em algoritmos de aprendizado de máquina [4].

Diante desse contexto, este estudo tem como objetivo o desenvolvimento de um acelerador INT8 baseado em systolic array, bem como sua integração em um System-on-Chip utilizando FPGA, explorando conceitos de modularização, encapsulamento e integração hardware/software.

# 2 FUNDAMENTAÇÃO TEÓRICA

## 2.1 ACELERADORES DE HARDWARE EM FPGA

Aceleradores de hardware são unidades especializadas projetadas para executar funções específicas de forma mais eficiente do que processadores de propósito geral. Segundo Hennessy e Patterson [1], essa abordagem permite aumentar significativamente o desempenho ao reduzir o número de instruções executadas em software.

Em dispositivos FPGA, aceleradores podem ser implementados diretamente na lógica reconfigurável, explorando paralelismo espacial e pipelines profundos. Estudos demonstram que, apesar de apresentarem menor eficiência em área quando comparados a ASICs, os FPGAs oferecem flexibilidade e tempo de desenvolvimento reduzido [3].

## 2.2 ARQUITETURA SYSTOLIC ARRAY

A arquitetura *systolic array*, introduzida por Kung e Leiserson [4], consiste em uma rede de unidades de processamento que operam de forma síncrona, com dados fluindo ritmicamente entre os elementos. Esse modelo reduz acessos à memória externa e favorece o reuso de dados intermediários.

Devido a essas características, systolic arrays são amplamente empregados em aceleradores para multiplicação matricial, convoluções e inferência em redes neurais.

## 2.3 ARITMÉTICA EM PONTO FIXO INT8

A utilização de aritmética em ponto fixo de 8 bits (INT8) tem se tornado comum em aceleradores de aprendizado de máquina. Trabalhos recentes demonstram que, com técnicas adequadas de quantização e acumulação, é possível obter desempenho elevado com baixo impacto na precisão [6]. Além disso, a aritmética INT8 reduz significativamente o consumo de energia e a área ocupada no FPGA.

# 3 METODOLOGIA

O desenvolvimento do acelerador seguiu uma abordagem incremental. Inicialmente, foi projetada a unidade elementar de processamento responsável pelas operações de multiplicação e acumulação em formato INT8. Em seguida, essas unidades foram interconectadas em um systolic array unidimensional.

Após a validação funcional em nível RTL, foi desenvolvido um wrapper compatível com o barramento Avalon Memory-Mapped, conforme especificado pela Intel [8]. Por fim, o acelerador foi integrado a um System-on-Chip utilizando o Intel Platform Designer [9], juntamente com um processador softcore, memória on-chip e controladores de clock e reset.

# 4 DESENVOLVIMENTO DO ACELERADOR

O acelerador desenvolvido neste estudo foi implementado integralmente em linguagem VHDL e estruturado de forma modular, visando facilitar sua compreensão, reutilização e expansão futura. A arquitetura adotada baseia-se em unidades elementares de processamento interconectadas em um systolic array unidimensional.

Cada unidade elementar de processamento realiza operações de multiplicação entre um dado de entrada e um peso, ambos representados em formato INT8. O resultado da multiplicação é acumulado em um registrador de 32 bits, garantindo segurança contra overflow durante operações sucessivas. A unidade é síncrona e utiliza pipeline para permitir operação em frequências mais elevadas.

O systolic array é formado pela interconexão sequencial dessas unidades, permitindo que os dados de entrada sejam propagados entre os elementos enquanto os pesos permanecem estacionários. Essa abordagem reduz a necessidade de acessos à memória e favorece o paralelismo computacional. O resultado final da acumulação é disponibilizado na saída do array para leitura pelo processador.

# 5 ARQUITETURA DO SYSTEM-ON-CHIP

A arquitetura do sistema em chip desenvolvido neste estudo segue um modelo clássico de SoC baseado em barramento, no qual um processador central coordena a execução das aplicações e o controle dos periféricos especializados. O acelerador INT8 foi integrado como um periférico customizado, acessível por meio do barramento Avalon Memory-Mapped.

A Figura 1 apresenta uma visão geral da arquitetura do sistema.

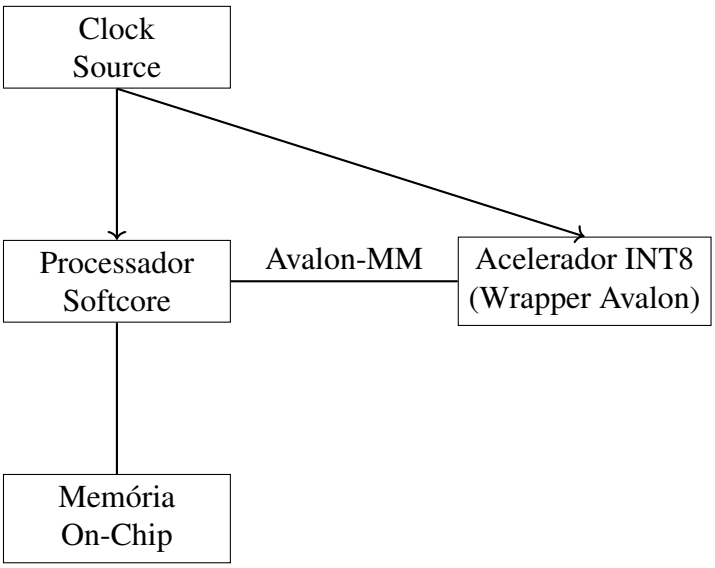


Figura 1 – Arquitetura do System-on-Chip com acelerador INT8 integrado

## 5.1 COMPONENTES DO SOC

O SoC é composto pelos seguintes blocos principais:

- **Fonte de Clock**: responsável pela geração do sinal de clock global do sistema.
- **Processador Softcore**: executa o software de controle e realiza acessos ao barramento Avalon-MM.
- **Memória On-Chip**: armazena o código do programa e dados de execução.
- **Controlador de Reset**: garante a inicialização síncrona e consistente dos módulos.
- **Acelerador INT8**: periférico customizado responsável pelas operações de multiplicação e acumulação.

## 5.2 MODELO DE COMUNICAÇÃO

A comunicação entre o processador e o acelerador ocorre exclusivamente por meio de acessos ao barramento Avalon Memory-Mapped. O acelerador é mapeado em um espaço de endereçamento específico e controlado por registradores, permitindo uma integração simples e eficiente.

Essa abordagem favorece a modularidade do sistema, pois o acelerador pode ser substituído ou expandido sem a necessidade de alterações na arquitetura global do SoC.

## 5.3 FLUXO DE DADOS DO SISTEMA

O fluxo de dados do sistema inicia-se no software executado pelo processador, que escreve os valores de entrada e os pesos nos registradores do acelerador por meio de acessos de escrita no barramento Avalon-MM. Após a configuração dos dados, o software ativa o acelerador por meio de um registrador de controle.

Internamente, o wrapper converte os acessos do barramento em sinais de controle e dados para o systolic array. Os dados de entrada são propagados sequencialmente entre as unidades de processamento, enquanto os pesos permanecem estacionários em cada elemento do array. O resultado acumulado é então disponibilizado em um registrador de saída, que pode ser lido pelo processador.

Esse modelo de operação permite desacoplar completamente o software da implementação interna do acelerador, reforçando o encapsulamento e a reutilização do hardware.

# 6 VERIFICAÇÃO FUNCIONAL E ANÁLISE DE DESEMPENHO

Este capítulo apresenta o processo de verificação funcional do acelerador INT8 integrado ao System-on-Chip, bem como a análise de desempenho obtida por meio de simulação temporal. As validações foram realizadas em nível RTL utilizando o ModelSim, permitindo observar o comportamento interno do sistema e medir métricas como latência e throughput.

## 6.1 AMBIENTE DE SIMULAÇÃO

A verificação funcional do sistema foi realizada por meio de um testbench desenvolvido em VHDL, responsável por gerar os sinais de clock, reset e estímulos do barramento Avalon Memory-Mapped. O clock do sistema foi configurado com período de 10 ns, correspondente a uma frequência de 100 MHz.

O testbench simula o comportamento do software embarcado, realizando escritas nos registradores do acelerador, ativando sua execução e monitorando os sinais de saída.

## 6.2 INICIALIZAÇÃO DO SISTEMA

A Figura 2 apresenta os sinais de clock e reset do sistema. Observa-se que o reset é aplicado durante os ciclos iniciais da simulação, garantindo a inicialização adequada de todos os registradores internos do acelerador e do System-on-Chip.


Figura 2 – Sinais de clock e reset durante a inicialização do sistema

## 6.3 ESCRITA E CONTROLE VIA BARRAMENTO AVALON

A comunicação entre o processador e o acelerador ocorre por meio do barramento Avalon Memory-Mapped. A Figura 3 ilustra as operações de escrita realizadas pelo software nos registradores internos do acelerador, incluindo a escrita dos dados de entrada, dos pesos e do registrador de controle responsável por iniciar a execução.


Figura 3 – Escrita de dados e controle no acelerador via barramento Avalon-MM

## 6.4 REGISTRADORES INTERNOS DO ACELERADOR

A Figura 4 apresenta o comportamento dos registradores internos do acelerador durante a simulação. Observa-se o correto armazenamento dos valores de entrada, dos pesos e do registrador de controle, validando o funcionamento do wrapper Avalon e o mapeamento de memória adotado.

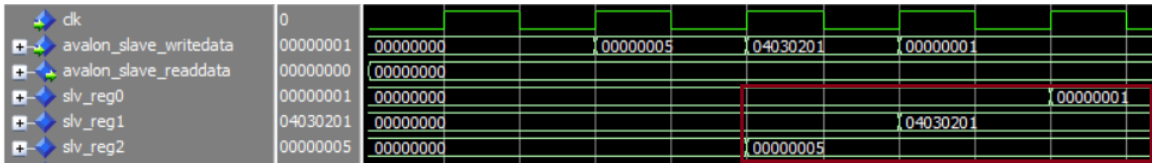


Figura 4 – Estado dos registradores internos do acelerador

## 6.5 EXECUÇÃO DO ACELERADOR

Após a ativação do sinal de início de execução, o acelerador realiza as operações de multiplicação e acumulação em formato INT8. A Figura 5 apresenta o sinal de saída do acelerador, evidenciando a geração do resultado final após o processamento interno.


Figura 5 – Execução do acelerador e geração do resultado

## 6.6 ANÁLISE DE LATÊNCIA

A latência do acelerador foi medida como o número de ciclos de clock entre a ativação do sinal *start* e a disponibilização do resultado válido na saída. Conforme ilustrado na Figura 6, observa-se que o resultado final é produzido após três ciclos de clock, sendo identificado um valor intermediário em dois ciclos.

Considerando o período de clock de 10 ns, a latência total do acelerador é de aproximadamente 30 ns.

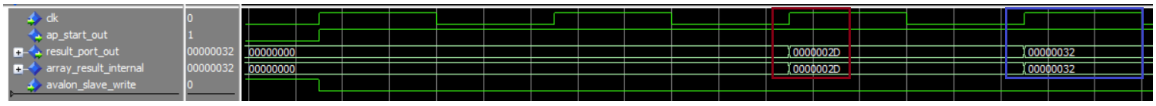


Figura 6 – Latência do acelerador medida em ciclos de clock

## 6.7 ANÁLISE DE THROUGHPUT

A análise de throughput foi realizada observando-se a capacidade do acelerador de produzir resultados em execuções consecutivas. A Figura 7 mostra que, após o preenchimento do pipeline, novos resultados são produzidos a cada dois ciclos de clock.

Esse comportamento evidencia a eficiência do paralelismo estrutural adotado na arquitetura do systolic array, permitindo alta taxa de processamento com baixa latência incremental.


Figura 7 – Throughput do acelerador após preenchimento do pipeline

## 6.8 DISCUSSÃO DOS RESULTADOS

Os resultados obtidos demonstram o correto funcionamento do acelerador INT8 integrado ao System-on-Chip, validando tanto a lógica de computação quanto a interface de comunicação via barramento Avalon-MM.

A baixa latência e o alto throughput observados confirmam a eficácia da arquitetura baseada em systolic array, evidenciando o potencial de aceleradores dedicados em FPGA para aplicações computacionalmente intensivas.

# 7 COMPARAÇÃO COM PROCESSADOR DE PROPÓSITO GERAL

Este capítulo apresenta uma análise comparativa entre o acelerador INT8 baseado em *systolic array* desenvolvido neste estudo e um processador de propósito geral, representado por uma arquitetura *in-order* típica, como aquelas utilizadas em softcores embarcados em FPGA. A comparação é realizada por meio de um modelo analítico baseado em contagem de ciclos de clock, abordagem amplamente adotada na literatura para avaliação de desempenho arquitetural [1].

A opção por um modelo analítico justifica-se pela ausência de uma plataforma física para execução de software embarcado, bem como pelo fato de que a análise em nível de ciclos permite uma comparação justa e independente de otimizações específicas de compilador.

## 7.1 MODELO COMPUTACIONAL AVALIADO

A operação avaliada corresponde a um cálculo de multiplicação e acumulação (*Multiply-Accumulate* – MAC), amplamente utilizada em aplicações de processamento digital de sinais e aprendizado de máquina. Considerando um conjunto de quatro pesos, a operação pode ser descrita conforme a Equação 7.1.

$$y = \sum_{i=0}^{3} x \cdot w_i \tag{7.1}$$

No acelerador proposto, essa operação é executada de forma paralela por meio de quatro unidades de processamento interconectadas em um systolic array. Em contrapartida, em um processador de propósito geral *in-order*, a mesma operação é realizada de forma sequencial, por meio de um laço de instruções em software.

## 7.2 MODELO ANALÍTICO DO PROCESSADOR

Para a estimativa de desempenho do processador, considera-se uma arquitetura simples, *in-order*, sem execução fora de ordem ou múltiplas unidades de execução, conforme descrito por Hennessy e Patterson [1]. A Tabela 1 apresenta uma estimativa conservadora do número de ciclos necessários para cada operação elementar.

Tabela 1 – Estimativa de ciclos por operação em processador *in-order*

| **Operação** | **Ciclos estimados** |
|---|---|
| Leitura do dado de entrada | 1 |
| Leitura do peso | 1 |
| Multiplicação | 1 |
| Acumulação (soma) | 1 |
| Controle de laço | 1 |
| **Total por iteração** | **5 ciclos** |

Considerando quatro iterações para o cálculo completo, o tempo total estimado é de aproximadamente 20 ciclos, acrescido de ciclos adicionais para inicialização e finalização, resultando em uma latência total aproximada de 25 a 30 ciclos por operação.

## 7.3 DESEMPENHO DO ACELERADOR

O desempenho do acelerador foi obtido por meio de simulação RTL no ModelSim, conforme apresentado no capítulo anterior. O clock do sistema foi configurado com período de 10 ns (100 MHz). As medições realizadas indicaram:

- Latência de 3 ciclos de clock entre o sinal de início e a disponibilização do resultado final;
- Produção de novos resultados a cada 2 ciclos de clock após o preenchimento do pipeline;
- Execução paralela de quatro operações de multiplicação e acumulação.

Esse comportamento é característico de arquiteturas baseadas em systolic arrays, nas quais o paralelismo estrutural e o pipeline permitem elevado throughput [4].

## 7.4 COMPARAÇÃO DE DESEMPENHO

A Tabela 2 apresenta a comparação direta entre o acelerador proposto e o processador de propósito geral, considerando a mesma frequência de clock.

Tabela 2 – Comparação de desempenho entre acelerador e processador

| **Métrica** | **Acelerador** | **Processador** |
|---|---|---|
| Frequência de clock | 100 MHz | 100 MHz |
| Latência (ciclos) | 3 | $\approx$ 25–30 |
| Latência (tempo) | 30 ns | 250–300 ns |
| Throughput | 1 resultado / 2 ciclos | 1 resultado / $\approx$ 25 ciclos |
| Modelo de execução | Paralelo | Sequencial |
| Tipo de aritmética | INT8 dedicada | INT8 via software |

A partir dos dados apresentados, observa-se que o acelerador apresenta uma redução de latência de aproximadamente uma ordem de grandeza quando comparado ao processador. Além

disso, o throughput do acelerador é significativamente superior, evidenciando sua capacidade de processar múltiplas operações de forma eficiente.

## 7.5 ANÁLISE DE SPEEDUP

A fim de quantificar de forma objetiva os ganhos de desempenho proporcionados pelo acelerador INT8 desenvolvido, realizou-se uma análise de *speedup* baseada em contagem de ciclos de clock. O *speedup* é uma métrica amplamente utilizada na literatura de arquitetura de computadores para expressar a razão entre o tempo de execução de uma solução de referência e o tempo de execução de uma solução otimizada [1].

Neste estudo, o processador de propósito geral *in-order* foi adotado como referência, enquanto o acelerador baseado em *systolic array* representa a solução otimizada. Considerando que ambos operam na mesma frequência de clock, o tempo de execução pode ser diretamente relacionado ao número de ciclos necessários para a conclusão da operação avaliada.

### 7.5.1 Speedup de Latência

O *speedup de latência* considera o tempo necessário para a produção de um único resultado completo. Conforme discutido anteriormente, a execução da operação de multiplicação e acumulação (MAC) envolvendo quatro pesos requer aproximadamente 25 ciclos de clock quando realizada em software no processador de propósito geral. Em contrapartida, o acelerador proposto disponibiliza o resultado final após apenas três ciclos de clock.

O *speedup de latência* pode, portanto, ser calculado conforme a Equação 7.2:

$$\text{Speedup}_{\text{latência}} = \frac{25}{3} \approx 8{,}33 \tag{7.2}$$

Esse resultado indica que o acelerador é aproximadamente oito vezes mais rápido que o processador para a execução de uma única operação MAC, evidenciando o impacto positivo do paralelismo estrutural e da lógica dedicada.

### 7.5.2 Speedup de Throughput

Além da latência, é relevante avaliar o desempenho do sistema em cenários de execução contínua, nos quais múltiplas operações são processadas de forma sequencial. Nessa situação, o *speedup de throughput* reflete a taxa de produção de resultados após o preenchimento do pipeline.

Enquanto o processador produz um resultado aproximadamente a cada 25 ciclos de clock, o acelerador gera novos resultados a cada dois ciclos de clock após o estágio inicial de pipeline. Dessa forma, o *speedup de throughput* é dado pela Equação 7.3:

$$\text{Speedup}_{\text{throughput}} = \frac{25}{2} = 12{,}5 \tag{7.3}$$

Esse valor demonstra que, em aplicações caracterizadas por fluxo contínuo de dados, o acelerador apresenta um ganho de desempenho ainda mais expressivo, sendo mais de doze vezes superior ao processador de propósito geral.

### 7.5.3 Resumo dos Resultados de Speedup

A Tabela 3 apresenta um resumo dos valores de *speedup* obtidos nesta análise.

Tabela 3 – Resumo dos valores de speedup obtidos

| **Métrica** | **Speedup** |
|---|---|
| Speedup de latência | 8,33 |
| Speedup de throughput | 12,5 |

Os resultados apresentados confirmam que o uso de um acelerador dedicado baseado em *systolic array* proporciona ganhos substanciais de desempenho, especialmente em cenários com elevado grau de paralelismo e reutilização de dados. Esses ganhos são consistentes com estudos prévios da literatura e reforçam a adequação de aceleradores em FPGA para aplicações computacionalmente intensivas.

## 7.6 DISCUSSÃO

Os resultados obtidos corroboram a literatura, que aponta aceleradores dedicados como soluções eficazes para aplicações com alto grau de paralelismo [2]. Enquanto o processador executa as operações de forma sequencial, o acelerador explora paralelismo espacial, reduzindo significativamente o número de ciclos necessários para a computação.

Adicionalmente, o uso de aritmética em ponto fixo INT8 contribui para a redução da complexidade da lógica, do consumo energético e da latência, conforme discutido em estudos recentes sobre quantização em aprendizado de máquina [6].

Dessa forma, a comparação evidencia que, para aplicações específicas como operações de multiplicação e acumulação, o uso de aceleradores dedicados em FPGA pode proporcionar ganhos expressivos de desempenho quando comparado a processadores de propósito geral.

# 8 CONCLUSÃO

Este trabalho apresentou o projeto, a implementação e a avaliação de um acelerador aritmético INT8 baseado em arquitetura de *systolic array*, integrado a um System-on-Chip em FPGA por meio do barramento Avalon Memory-Mapped. O objetivo principal foi investigar o impacto do uso de arquiteturas dedicadas na aceleração de operações computacionalmente intensivas, em especial multiplicação e acumulação, amplamente empregadas em aplicações de processamento digital de sinais e aprendizado de máquina.

A arquitetura desenvolvida foi implementada integralmente em VHDL e estruturada de forma modular, o que facilitou sua validação funcional e integração ao sistema. A adoção de um systolic array unidimensional permitiu explorar paralelismo estrutural e pipeline, resultando em uma solução capaz de produzir resultados com baixa latência e elevado throughput. As simulações em nível RTL indicaram uma latência de apenas três ciclos de clock entre a ativação do acelerador e a disponibilização do resultado final, além da produção de novos resultados a cada dois ciclos após o preenchimento do pipeline.

A integração do acelerador ao System-on-Chip, realizada por meio de um wrapper compatível com o barramento Avalon Memory-Mapped, demonstrou a viabilidade de encapsular aceleradores dedicados como periféricos acessíveis por software. Esse modelo de comunicação permite desacoplar o controle do acelerador de sua implementação interna, favorecendo a modularidade e a reutilização do hardware.

A comparação analítica com um processador de propósito geral, baseada em contagem de ciclos de clock, evidenciou ganhos expressivos de desempenho. Enquanto o processador executa a operação de forma sequencial, demandando dezenas de ciclos de clock, o acelerador explora paralelismo espacial, reduzindo significativamente a latência e aumentando o throughput. Esses resultados estão alinhados com a literatura, que aponta aceleradores dedicados em FPGA como soluções eficazes para aplicações com alto grau de paralelismo.

Como limitação deste estudo, destaca-se o fato de que as análises foram conduzidas exclusivamente por meio de simulação RTL, não contemplando medições em hardware físico, como consumo energético e utilização efetiva de recursos do FPGA. Ainda assim, a abordagem adotada é amplamente aceita para avaliação arquitetural preliminar e permite conclusões consistentes em nível de projeto.

Como trabalhos futuros, propõe-se a implementação do acelerador em uma plataforma FPGA real, possibilitando a obtenção de métricas adicionais, como potência dissipada e eficiência energética. Além disso, a execução de software embarcado em um processador softcore para comparação direta de desempenho e a expansão da arquitetura para suportar arrays de maior dimensão ou operações mais complexas representam extensões naturais deste trabalho.

Dessa forma, conclui-se que o acelerador desenvolvido atende aos objetivos propostos,

demonstrando o potencial de arquiteturas baseadas em *systolic arrays* como alternativas eficientes para a aceleração de operações aritméticas em sistemas embarcados baseados em FPGA.

# REFERÊNCIAS


[1] HENNESSY, John L.; PATTERSON, David A. *Computer Architecture: A Quantitative Approach*. 6. ed. San Francisco: Morgan Kaufmann, 2019.

[2] KUON, Ian; TESSIER, Russell; ROSE, Jonathan. FPGA Architecture: Survey and Challenges. *Foundations and Trends in Electronic Design Automation*, v. 2, n. 2, p. 135–253, 2008.

[3] KUON, Ian; ROSE, Jonathan. Measuring the gap between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 26, n. 2, p. 203–215, 2007.

[4] KUNG, H. T.; LEISERSON, C. E. Systolic Arrays for VLSI. In: *Sparse Matrix Proceedings*. Academic Press, 1979.

[5] CHEN, Yu-Hsin et al. Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks. *IEEE Journal of Solid-State Circuits*, v. 52, n. 1, p. 127–138, 2017.

[6] JACOB, Benoit et al. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] MEHER, Pramod Kumar et al. Efficient FPGA and ASIC realizations of digital signal processing systems. *IEEE Transactions on Circuits and Systems I*, v. 59, n. 6, p. 1200–1213, 2012.

[8] INTEL CORPORATION. *Avalon Interface Specifications*. Intel FPGA Documentation, 2024.

[9] INTEL CORPORATION. *Platform Designer User Guide*. Intel FPGA Documentation, 2024.

[10] BERGERON, Janick. *Writing Testbenches: Functional Verification of HDL Models*. 2. ed. Springer, 2003.

[11] WOLF, Wayne. *Computers as Components: Principles of Embedded Computing System Design*. 3. ed. Morgan Kaufmann, 2012.