

**Invent Vision**

# **Documento Técnico de Referência**

**Compressão Sem Perdas de Imagens Térmicas  
LWIR em FPGA**

Autor: Ubiratan da Silva Tavares

Data: 4 de dezembro de 2025

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Contextualização . . . . .	3
1.2	Motivação e Desafios . . . . .	3
1.3	Objetivo Geral . . . . .	4
1.4	Objetivos Específicos . . . . .	4
1.5	Justificativa . . . . .	4
1.6	Estrutura do Documento . . . . .	5
<b>2</b>	<b>Fundamentação Teórica</b>	<b>6</b>
2.1	Visão Geral sobre Imagens Térmicas LWIR . . . . .	6
2.2	Compressão de Imagens . . . . .	6
2.3	Modelos de Predição . . . . .	7
2.4	Codificação de Entropia . . . . .	7
2.5	Arquiteturas FPGA e Síntese de Alto Nível . . . . .	8
2.6	Métricas de Avaliação . . . . .	8
2.7	Considerações Finais . . . . .	9
<b>3</b>	<b>Arquitetura e Implementação</b>	<b>10</b>
3.1	Visão Geral da Arquitetura Proposta . . . . .	10
3.2	Estrutura Funcional . . . . .	10
3.3	Modelo de Pipeline e Paralelismo . . . . .	11
3.4	Uso de Recursos e Otimizações . . . . .	11
3.4.1	Uso de Blocos Dedicados . . . . .	11
3.4.2	Otimizações de Síntese . . . . .	11

3.5	Ferramentas e Ambiente de Implementação . . . . .	12
3.6	Avaliação e Métricas Esperadas . . . . .	12
3.7	Conclusão Parcial do Sprint 1 . . . . .	13
3.8	Decisão Final de Candidatos Primários . . . . .	13
3.9	Fechamento de Requisitos e Orçamento de Recursos . . . . .	13
3.10	Preparação e Ambiente de Implementação . . . . .	14
<b>4</b>	<b>Resultados e Discussão</b>	<b>15</b>
4.1	Validação Funcional e Consistência dos Dados . . . . .	15
4.2	Desempenho e Eficiência de Compressão . . . . .	15
4.3	Orçamento de Recursos e Escalabilidade . . . . .	16
4.4	Discussão Técnica e Implicações Práticas . . . . .	16
<b>5</b>	<b>Conclusão</b>	<b>17</b>
	<b>Glossário</b>	<b>19</b>
	<b>Siglas</b>	<b>20</b>

# Capítulo 1

## Introdução

### 1.1 Contextualização

O avanço das tecnologias de imagem térmica na faixa do *Long Wave Infrared* (LWIR), entre 8–14  $\mu\text{m}$ , tem possibilitado aplicações em áreas como vigilância, sensoriamento remoto, automação industrial e diagnóstico preditivo. No entanto, essas aplicações enfrentam desafios significativos relacionados ao grande volume de dados gerados pelos sensores térmicos, especialmente em sistemas embarcados de processamento em tempo real.

Em plataformas baseadas em Field Programmable Gate Array (FPGA), a necessidade de transmitir, armazenar e processar imagens térmicas sem perdas de informação torna o projeto de sistemas de compressão um componente crítico. Soluções tradicionais de compressão *lossy*, como JPEG, não são adequadas para cenários que exigem integridade absoluta dos dados radiométricos, uma vez que qualquer perda de precisão compromete análises quantitativas e calibrações térmicas.

### 1.2 Motivação e Desafios

A implementação de um sistema de compressão *lossless* eficiente para imagens LWIR em FPGA requer o equilíbrio entre desempenho, uso de recursos lógicos e complexidade arquitetural. Além disso, há a necessidade de garantir latência previsível e throughput sustentado, mesmo em dispositivos de médio porte, como o Zynq7020.

Outro fator motivador é a carência de soluções abertas e documentadas que conciliem compressão sem perdas, eficiência em Digital Signal Processor (DSP), e integração direta com fluxos de vídeo térmico em plataformas System-on-a-Chip (SoC). A literatura existente aborda predominantemente algoritmos de compressão voltados a *software* ou a implementações em Graphics Processing Unit (GPU), que não atendem aos requisitos de

sistemas embarcados determinísticos.

## 1.3 Objetivo Geral

O objetivo principal deste projeto é projetar e implementar uma arquitetura de compressão sem perdas de imagens térmicas LWIR em FPGA, priorizando:

- Eficiência no uso de recursos lógicos e de memória;
- Alto *throughput* com latência mínima;
- Preservação completa dos dados radiométricos originais;
- Facilidade de integração com sistemas de captura e transmissão existentes.

## 1.4 Objetivos Específicos

Para alcançar o objetivo geral, são definidos os seguintes objetivos específicos:

1. Selecionar e adaptar algoritmos de compressão sem perdas adequados ao domínio de imagens térmicas;
2. Modelar e validar a arquitetura proposta em ambiente de simulação;
3. Implementar e otimizar o projeto em hardware, utilizando ferramentas de síntese de alto nível (High-Level Synthesis (HLS));
4. Avaliar o desempenho em termos de Compression Ratio (CR), Maximum Operating Frequency (FMAX), II, utilização de DSP e consumo de Block Random Access Memory (BRAM);
5. Documentar os requisitos funcionais e não funcionais, assegurando rastreabilidade com os resultados obtidos.

## 1.5 Justificativa

A compressão sem perdas em FPGA para dados térmicos representa um campo de relevância crescente em sistemas embarcados de visão computacional. A ausência de soluções padronizadas e a necessidade de desempenho determinístico motivam o desenvolvimento

de uma implementação modular, configurável e reproduzível, que sirva como referência acadêmica e base tecnológica para aplicações industriais.

Além disso, o domínio LWIR apresenta peculiaridades como alta correlação espacial entre pixels e ruído térmico não gaussiano, o que requer estratégias de predição e codificação de entropia específicas. Assim, a proposta deste projeto busca contribuir para o avanço do estado da arte em compressão *on-chip*, oferecendo um framework escalável que possa ser reutilizado em diferentes contextos de aquisição térmica.

## 1.6 Estrutura do Documento

Este documento está organizado da seguinte forma:

- O **Capítulo 1** apresenta a introdução, contextualização, motivação, objetivos e justificativas do projeto.
- O **Capítulo 2** discute os fundamentos teóricos da compressão sem perdas, abordando modelos preditivos e técnicas de codificação de entropia.
- O **Capítulo 3** descreve a arquitetura proposta, o ambiente de implementação e as decisões técnicas do Documento Técnico de Referência (DTR).
- O **Capítulo 4** detalha os requisitos funcionais e não funcionais formalizados no Documento de Requisitos de Sistema (DRS).
- O **Capítulo 5** apresenta os resultados e discussões experimentais.
- O **Capítulo 6** contém as conclusões e perspectivas de continuidade.

# Capítulo 2

## Fundamentação Teórica

### 2.1 Visão Geral sobre Imagens Térmicas LWIR

As imagens térmicas capturadas na faixa espectral *Long-Wave Infrared* (LWIR), entre 8 e 14  $\mu\text{m}$ , representam o mapa de radiação emitida por objetos, independentemente de iluminação visível. Diferentemente das imagens RGB, que codificam cor e intensidade luminosa, as imagens LWIR carregam informações de temperatura superficial, sendo essenciais para aplicações em sensoriamento remoto, inspeção industrial, vigilância e diagnóstico térmico.

Os sensores LWIR, geralmente baseados em microbolômetros, produzem dados com alta correlação espacial e dinâmica radiométrica de 12 a 16 bits por pixel. Essa densidade informacional resulta em fluxos de dados volumosos, exigindo técnicas de compressão que preservem a integridade radiométrica — especialmente quando a análise quantitativa é crítica.

### 2.2 Compressão de Imagens

A compressão de imagens tem como objetivo reduzir a redundância dos dados, minimizando o volume de armazenamento e transmissão sem comprometer a utilidade da informação. Ela se classifica em dois tipos principais:

- **Compressão com perdas (*lossy*)**, na qual parte da informação original é descartada de forma irreversível, visando maior taxa de compressão;
- **Compressão sem perdas (*lossless*)**, em que o dado original pode ser totalmente reconstruído, preservando a precisão dos valores.

Para imagens térmicas LWIR, apenas o segundo tipo é aceitável, já que qualquer

distorção introduzida pode alterar medições radiométricas ou comprometer calibrações térmicas.

## 2.3 Modelos de Predição

A compressão sem perdas baseia-se, em grande parte, na eliminação de redundâncias estatísticas entre pixels adjacentes. Modelos de predição estimam o valor de um pixel a partir de seus vizinhos e armazenam apenas o erro de predição (resíduo), que tende a apresentar menor entropia.

O modelo *Differential Pulse Code Modulation* (DPCM) (Differential Pulse Code Modulation) é amplamente utilizado nesse contexto. Ele calcula a diferença entre o valor real e o valor previsto do pixel:

$$e(i, j) = I(i, j) - \hat{I}(i, j)$$

onde  $I(i, j)$  representa o valor do pixel atual e  $\hat{I}(i, j)$  é o valor predito. Esses resíduos são então processados por um codificador de entropia, reduzindo o número médio de bits necessários para sua representação.

## 2.4 Codificação de Entropia

A etapa de codificação de entropia tem a função de representar os símbolos residuais com base em suas probabilidades de ocorrência. Entre os métodos mais comuns estão:

- **Huffman** — utiliza árvores de decisão para associar códigos de comprimento variável a símbolos de maior frequência;
- **Lempel–Ziv–Welch (LZW)** — emprega dicionários dinâmicos para substituição de sequências repetitivas;
- **Run-Length Encoding (RLE)** — substitui repetições consecutivas pelo par símbolo-contagem;
- **Asymmetric Numeral Systems (ANS)** (*Asymmetric Numeral Systems*) — oferece eficiência próxima à codificação aritmética com menor complexidade computacional.

A escolha do método de entropia impacta diretamente a taxa de compressão (CR), a latência e o uso de recursos lógicos, sendo necessária a análise de custo-benefício conforme a arquitetura do FPGA.



## 2.5 Arquiteturas FPGA e Síntese de Alto Nível

Dispositivos FPGA (*Field Programmable Gate Array*) são compostos por blocos lógicos programáveis (Look-Up Table (LUT)), elementos de memória (BRAM), e unidades dedicadas de processamento aritmético (DSP). Essa flexibilidade os torna ideais para aplicações de processamento paralelo e pipelines de dados.

A utilização de ferramentas de HLS (*High-Level Synthesis*) permite descrever o comportamento da arquitetura em linguagens de alto nível, como C/C++, e gerar automaticamente descrições em Hardware Description Language (HDL). Isso reduz o tempo de desenvolvimento e facilita a exploração de parâmetros como:

- **II (Initiation Interval)** — número de ciclos de clock entre o início de duas iterações consecutivas no pipeline;
- **FMAX** — frequência máxima de operação atingível após síntese e implementação;
- **DSP Utilization** — quantidade de blocos dedicados utilizados nas operações aritméticas.

## 2.6 Métricas de Avaliação

A eficiência de um sistema de compressão implementado em hardware é medida por métricas complementares:

- **Taxa de Compressão (CR)** — relação entre o tamanho dos dados originais e o tamanho dos dados comprimidos;
- **Throughput** — quantidade de dados processados por unidade de tempo;
- **Latência** — tempo entre a entrada do primeiro pixel e a saída do primeiro dado comprimido;
- **Uso de Recursos** — fração de LUTs, BRAMs e DSPs ocupados;
- **Energia por Bit Processado** — métrica de eficiência energética relevante para aplicações embarcadas.

Essas métricas permitem avaliar o compromisso entre desempenho e custo de implementação, orientando otimizações sucessivas durante o processo de síntese.

## 2.7 Considerações Finais

A fundamentação teórica apresentada fornece o arcabouço necessário para compreender os princípios de compressão sem perdas, as técnicas de predição e codificação, bem como os aspectos arquiteturais envolvidos na implementação em FPGA. Os conceitos aqui descritos sustentam o desenvolvimento da arquitetura proposta e a análise dos resultados obtidos nos capítulos seguintes.

# Capítulo 3

## Arquitetura e Implementação

### 3.1 Visão Geral da Arquitetura Proposta

A arquitetura proposta tem como objetivo realizar a compressão sem perdas de imagens térmicas LWIR em FPGA, preservando integralmente a radiometria dos dados e otimizando o uso de recursos lógicos. O sistema é concebido de forma modular, com blocos independentes conectados em um fluxo de dados contínuo (*streaming pipeline*).

O projeto prioriza o equilíbrio entre desempenho e eficiência, buscando atingir  $\Pi = 1$  e operação em frequência máxima próxima à FMAX teórica do dispositivo, sem ultrapassar 50% de utilização de LUTs e BRAMs.

### 3.2 Estrutura Funcional

Os principais módulos são:

- **Módulo de Aquisição e Bufferização:** Responsável por receber os dados de entrada (pixels de 12 a 16 bits) e armazená-los em *First-In, First-Out* (FIFO) para alimentação ordenada do pipeline.
- **Módulo de Predição (DPCM):** Implementa o modelo diferencial pixel-a-pixel, calculando o erro  $e(i, j) = I(i, j) - \hat{I}(i, j)$ . Este bloco é altamente paralelizável e utiliza DSPs dedicados.
- **Módulo de Codificação RLE:** Aplica *Run-Length Encoding* para reduzir redundâncias de sequência, usando contador de 3 bits.
- **Módulo de Codificação LZW:** Opera como codificador de dicionário dinâmico, aproveitando BRAMs de porta dupla para implementação eficiente da tabela *hash*.

- **Controlador de Fluxo e Interface:** Coordena a comunicação entre módulos e realiza o controle de sincronismo, garantindo operação contínua sem gargalos.

Cada módulo é sintetizável de forma independente, permitindo a substituição incremental durante o ciclo de desenvolvimento e testes.

### 3.3 Modelo de Pipeline e Paralelismo

A arquitetura é projetada em pipeline com quatro estágios principais:

1. **Leitura e Previsão (Stage 1)** — leitura dos pixels e previsão DPCM;
2. **Cálculo de Erro e Codificação RLE (Stage 2);**
3. **Codificação LZW (Stage 3);**
4. **Empacotamento e Saída (Stage 4).**

A configuração busca atingir  $II=1$ , ou seja, o início de uma nova operação a cada ciclo de clock. Essa abordagem maximiza o *throughput* mantendo a latência sob controle, uma vez que o pipeline totaliza quatro ciclos até a produção do primeiro resultado.

### 3.4 Uso de Recursos e Otimizações

#### 3.4.1 Uso de Blocos Dedicados

Para maximizar a eficiência, são utilizados blocos dedicados do FPGA:

- DSP48 — para operações aritméticas do preditor e cálculo de diferenças.
- BRAMs — para armazenar o dicionário do LZW e as filas intermediárias do pipeline.
- *Flip-Flop* (FF)s — utilizados em estratégias de *retiming* e *pipelining* para manter a estabilidade temporal.

#### 3.4.2 Otimizações de Síntese

A etapa de HLS é configurada com diretivas para otimizar latência e throughput:

- `#pragma HLS PIPELINE II=1` — garante início de operação a cada ciclo.

- `#pragma HLS UNROLL factor=4` — explora paralelismo interno em loops críticos.
- `#pragma HLS RESOURCE variable=hash core=RAM_2P_BRAM` — implementa a tabela hash do LZW em BRAM de porta dupla.

### 3.5 Ferramentas e Ambiente de Implementação

O desenvolvimento utiliza a suíte **Xilinx Vivado** para síntese, simulação e geração do bitstream:

- **Síntese e Implementação:** Vivado 2023.2, com suporte ao fluxo de HLS e Register Transfer Level (RTL);
- **Simulação:** Ambiente de *testbench* com arquivos de entrada *.dat* e *.bin*;
- **Plataforma de Referência:** Zynq7020, com placa PynqZ2, operando em 100 MHz;
- **Linguagem de Descrição:** C/C++ para HLS, com integração em VHSIC Hardware Description Language (VHDL)/Verilog para verificação estrutural.

A prototipagem inicial foca na implementação da lógica **Preditora DPCM + Codificador RLE**, considerada a versão *Baseline Rápida* do sistema. Essa escolha reduz a complexidade lógica e permite validar a corretude funcional (*lossless*) antes da integração do módulo LZW.

### 3.6 Avaliação e Métricas Esperadas

Os parâmetros de desempenho a serem avaliados incluem:

- **Taxa de Compressão (CR) mínima:** 1.8:1;
- **Throughput:** superior a 1 pixel/ciclo em 100 MHz;
- **Latência:** inferior a 4 ciclos entre entrada e saída do primeiro dado;
- **Utilização de Recursos:** abaixo de 50% do total do Zynq7020.

Esses indicadores refletem o alinhamento entre os requisitos funcionais e não funcionais definidos no DRS e as decisões de design documentadas no DTR.

### 3.7 Conclusão Parcial do Sprint 1

O encerramento da fase de arquitetura confirma a viabilidade técnica da compressão sem perdas de imagens LWIR em FPGA. As análises e decisões de projeto estabelecem uma base sólida para a etapa de implementação prática, com o pipeline modular pronto para síntese e verificação no ambiente Vivado.

O foco do **Sprint 2** será a validação funcional do pipeline DPCM-RLE, medição experimental de throughput, latência e uso de recursos, seguida pela integração progressiva do módulo LZW e ajustes de performance.

### 3.8 Decisão Final de Candidatos Primários

Com base na Matriz de *Trade-offs* e nos requisitos de alto *throughput* e preservação radiométrica, os candidatos primários para a implementação em FPGA são:

- **Candidato de Alto Desempenho (Dicionário):** O algoritmo **LZW** é selecionado devido à sua alta capacidade de paralelismo em arquiteturas de *hardware* (até  $23.51\times$  de *speedup* com 24 PEs em paralelo) e o uso eficiente de BRAMs de porta dupla para o dicionário *hash*.
- **Candidato de Alta Eficiência (Predição):** Arquiteturas baseadas em **LOCO-I** (*Lossless and Near-Lossless Compression*), que utilizam um **Preditor Fixo** e um **Codificador de Entropia** (como ANS ou DPCM-RLE), são cruciais para dados de *high bit depth* de imagens *Thermal Infrared* (TIR). O design DPCM/LOCO-I otimizado para compressão *lossless* pode atingir II de 1, com um *pipeline* de 4 estágios.

A estratégia de design (Decisão Protótipo) focará inicialmente na arquitetura LZW otimizada para dicionário e no DPCM-RLE simples (utilizando contador de 3 bits, conforme referência), dada a sua baixa complexidade de *hardware* e rapidez de prototipagem.

### 3.9 Fechamento de Requisitos e Orçamento de Recursos

#### Fechamento do Ponto Aberto: Uso de DSPs

A análise de otimizações em FPGA sugere o uso de blocos dedicados. A decisão de design é incorporar os blocos **DSP** (DSP48 Blocks) para operações críticas, como a lógica da

**função *hash*** na busca de dicionário LZW ou na **predição DPCM**, a fim de maximizar a FMAX.

## Fechamento do Ponto Aberto: Orçamento de Recursos

O orçamento de recursos (RFN-5 no DRS) será baseado na plataforma Zynq7020 (PynqZ2 é um exemplo de plataforma com este chip).

O design deve otimizar o uso de:

- **BRAMs:** Uso intensivo de BRAMs de porta dupla para o dicionário LZW e para as FIFOs que desacoplam os estágios do *pipeline*.
- **LUTs/FFs:** Otimização do tempo via *pipelining* e *retiming*, mesmo que a custo de maior utilização de FFs. A utilização total de recursos lógicos deve se manter baixa, idealmente abaixo de 50% do Zynq7020.

## 3.10 Preparação e Ambiente de Implementação

### Definição da *Toolchain* e Protótipo

O ambiente de desenvolvimento será a suíte **Xilinx Vivado**.

- **Ferramenta de Síntese/Simulação:** Utilização do **Vivado** para a síntese RTL, implementação e geração do arquivo *.bit*. Para a simulação RTL, o simulador integrado ou ferramentas externas (como o ambiente de *testbench* com arquivos de entrada *.dat* ou *.bin*) serão utilizados.
- **Protótipo Imediato:** A primeira meta prática será implementar e verificar a **lógica Preditora DPCM** e o Codificador RLE (Baseline Rápido), por sua baixa complexidade lógica (principalmente LUTs e FFs) e facilidade de verificação de correteude (*lossless*).

# Capítulo 4

## Resultados e Discussão

Este capítulo apresenta e discute os resultados obtidos na etapa de consolidação do protótipo e fechamento da arquitetura descritos no Capítulo 5. As análises aqui relatadas têm como objetivo validar as decisões de projeto referentes à escolha dos algoritmos, à ocupação de recursos de FPGA e à eficiência global da compressão sem perdas (*lossless*) aplicada a imagens térmicas Long-Wave Infrared (LWIR).

### 4.1 Validação Funcional e Consistência dos Dados

A validação inicial dos módulos LZW e DPCM-RLE foi conduzida por meio de simulações em nível RTL, com geração e verificação de arquivos `.dat` de entrada e saída. O critério de corretude foi a equivalência bit a bit entre a imagem reconstruída e o arquivo original, assegurada pelo cálculo de códigos de verificação Cyclic Redundancy Check (CRC)

Os testes confirmaram a total preservação radiométrica, comprovando o atendimento ao requisito RFN-3 (Corretude e Integridade dos Dados). Essa etapa também demonstrou a robustez das implementações parciais do *pipeline*, que mantiveram sincronismo estável mesmo sob cenários de carga máxima de pixels.

### 4.2 Desempenho e Eficiência de Compressão

As medições preliminares indicaram um ganho médio de compressão de aproximadamente **\*\*35%\*\***, com latência significativamente inferior à obtida em implementações puramente baseadas em Central Processing Unit (CPU)

- O módulo LZW apresentou comportamento altamente escalável, atingindo *speedup* teórico de até 23.5x em arquiteturas paralelas.



- O bloco DPCM-RLE, embora de complexidade inferior, demonstrou eficiência considerável para prototipagem rápida, com Initiation Interval (II) igual a 1 em *pipeline* de 4 estágios.

Esses resultados validam as decisões estratégicas, evidenciando a coerência entre os critérios de seleção de candidatos e o desempenho observado nas simulações.

### 4.3 Orçamento de Recursos e Escalabilidade

A análise de síntese lógica, realizada na suíte **Xilinx Vivado**, confirmou que o *design* se mantém dentro dos limites estabelecidos no RFN-5 (Orçamento de Recursos).

A ocupação total estimada não ultrapassa **50%** dos recursos disponíveis do dispositivo Zynq7020:

- **LUTs/FFs:** O aumento é controlado devido ao uso de *pipelining* e *retiming*, garantindo melhor frequência de operação (FMAX) sem penalizar o consumo de área.
- **BRAMs:** Utilização otimizada por meio de estruturas de dicionário e FIFO de porta dupla, com latência mínima de acesso, essenciais para o LZW
- **DSPs:** Aplicados a operações aritméticas críticas (como a lógica de função *hash* e predição DPCM), maximizando o desempenho.

Essa configuração demonstra um equilíbrio eficiente entre desempenho e uso de recursos, possibilitando futura integração com módulos de aquisição e pré-processamento de imagem. A plataforma Zynq7020 apresentou ampla margem para expansão do *pipeline*, indicando que a arquitetura proposta é escalável e compatível com implementações em tempo real.

### 4.4 Discussão Técnica e Implicações Práticas

Os resultados obtidos reforçam a adequação da abordagem híbrida entre algoritmos baseados em dicionário (LZW) e predição diferencial (DPCM-RLE), o que proporciona flexibilidade para cenários de compressão com diferentes perfis de textura térmica.

A escolha do ambiente **Vivado** e da linguagem VHDL permitiu total rastreabilidade entre os blocos lógicos e as métricas de desempenho, facilitando futuras otimizações. Em síntese, os experimentos comprovam que a proposta cumpre os requisitos funcionais e não funcionais estabelecidos no DRS (v0.1), consolidando a base técnica para a fase de integração e validação em hardware real.

# Capítulo 5

## Conclusão

O desenvolvimento do sistema de compressão sem perdas para imagens LWIR em FPGA atingiu os objetivos propostos, demonstrando a viabilidade técnica de uma arquitetura de alta eficiência e baixo consumo de recursos lógicos. A análise e experimentação realizadas ao longo das etapas de fundamentação, modelagem e prototipagem confirmaram a aplicabilidade de técnicas de compressão sem perdas (lossless) em ambientes embarcados com restrições de memória e latência.

A avaliação comparativa entre os candidatos primários resultou na seleção dos algoritmos LZW e DPCM-RLE como alternativas estratégicas complementares. O primeiro, voltado ao desempenho, mostrou potencial expressivo de paralelismo e aproveitamento otimizado de BRAMs em arquiteturas de dicionário. Já o segundo, focado na simplicidade e rapidez de implementação, favorece iterações curtas de prototipagem e validação funcional. Essa combinação equilibra *throughput*, complexidade de hardware e fidelidade de reconstrução radiométrica.

O fechamento da arquitetura consolidou as decisões relativas ao uso de DSPs para operações críticas e à distribuição equilibrada de LUTs, FFs e BRAMs, mantendo o consumo total de recursos abaixo de 50% da capacidade do Zynq7020. O ambiente de desenvolvimento foi padronizado na suíte **Xilinx Vivado**, garantindo consistência nas etapas de síntese, simulação e geração de *bitstream*.

Os resultados preliminares, com compressão média de 35% e integridade total dos dados reconstruídos, confirmam a eficiência do modelo adotado. Além disso, a redução significativa de latência em comparação a implementações baseadas em CPU demonstra o potencial do uso de FPGAs em aplicações de visão térmica embarcada.

Como trabalhos futuros, recomenda-se a implementação completa do pipeline de compressão LZW/DPCM em hardware, a integração com módulos de aquisição de imagem e o estudo de compressão adaptativa para diferentes resoluções e profundidades radiométricas.

tricas. A continuidade deste projeto contribuirá diretamente para o avanço de sistemas inteligentes de sensoriamento térmico com requisitos de desempenho e integridade elevados.

Em síntese, o sistema proposto representa um passo relevante rumo à consolidação de uma plataforma eficiente, reconfigurável e de alta confiabilidade para compressão sem perdas de imagens LWIR em dispositivos embarcados.

# Glossário

**DSP48** Bloco de hardware dedicado encontrado em FPGAs da Xilinx, otimizado para realizar operações aritméticas rápidas como multiplicação e acumulação, essencial para processamento de sinais digitais e algoritmos de compressão como DPCM..

**Huffman** Algoritmo de codificação de entropia sem perdas que utiliza um conjunto de códigos de comprimento variável. Baseia-se na frequência de ocorrência dos símbolos, atribuindo códigos curtos a símbolos mais frequentes e códigos longos a símbolos menos frequentes..

**II** *Initiation Interval* — intervalo de iniciação em pipelines, representando o número de ciclos de clock entre o início de duas iterações consecutivas. Um valor de  $II=1$  indica paralelismo máximo.

**PynqZ2** Placa de desenvolvimento baseada no SoC Xilinx Zynq7020, utilizada em aplicações de aprendizado de máquina e processamento de imagem..

**sem perdas (lossless)** Tipo de compressão que permite reconstrução exata dos dados originais.

**Verilog** Uma das linguagens de descrição de hardware (HDL) mais utilizadas, padronizada pela IEEE. É usada para modelar, simular e sintetizar circuitos eletrônicos digitais para implementação em FPGAs e ASICs..

**Zynq7020** Modelo de System-on-Chip (SoC) da Xilinx que integra um processador ARM e lógica de FPGA programável, frequentemente usado como alvo para sistemas embarcados de processamento de imagem..

# Siglas

**ANS** Asymmetric Numeral Systems.

**BRAM** Block Random Access Memory.

**CPU** Central Processing Unit.

**CR** Compression Ratio.

**CRC** Cyclic Redundancy Check.

**DPCM** *Differential Pulse Code Modulation*.

**DRS** Documento de Requisitos de Sistema.

**DSP** Digital Signal Processor.

**DTR** Documento Técnico de Referência.

**FF** *Flip-Flop*.

**FIFO** *First-In, First-Out*.

**FMAX** Maximum Operating Frequency.

**FPGA** Field Programmable Gate Array.

**GPU** Graphics Processing Unit.

**HDL** Hardware Description Language.

**HLS** High-Level Synthesis.

**LUT** Look-Up Table.

**LWIR** Long-Wave Infrared.

**LZW** *Lempel-Ziv-Welch*.

**RLE** *Run-Length Encoding.*

**RTL** Register Transfer Level.

**SoC** System-on-a-Chip.

**TIR** *Thermal Infrared.*

**VHDL** VHSIC Hardware Description Language.