

UNIVERSIDADE CATÓLICA DE PELOTAS
CENTRO DE CIÊNCIAS SOCIAIS E TECNOLÓGICAS
MESTRADO EM ENGENHARIA ELETRÔNICA E COMPUTAÇÃO

BIANCA SILVEIRA

**Exploração Arquitetural nas Métricas de
Similaridade para Codificadores de Vídeo
do Padrão HEVC**

Dissertação apresentada como requisito parcial para
a obtenção do grau de Mestre em Engenharia
Eletrônica e Computação

Orientador: Prof. Dr. Eduardo Antonio César da
Costa

Pelotas
2016

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Silveira, Bianca

Exploração Arquitetural nas Métricas de Similaridade para Codificadores de Vídeo do Padrão HEVC / Bianca Silveira. – Pelotas: 2016.

69 f.: il.

Dissertação (mestrado) – Universidade Católica de Pelotas. 2016. Orientador: Eduardo Antonio César da Costa.

1. Codificação de vídeo. 2. HEVC. 3. Arquitetura de hardware. 4. SATD. 5. SAD. 6. Somadores compressores. I. Costa, Eduardo Antonio César da. II. Título.

UNIVERSIDADE CATÓLICA DE PELOTAS

Reitor: Prof. José Carlos Pereira Bachettini Júnior

Pró-Reitor-Acadêmico: Profa. Patrícia Haertel Giusti

Coordenador de Pesquisa e Pós-Graduação Stricto Sensu: Prof. Ricardo Tavares Pinheiro

Diretor do Centro de Ciências Sociais e Tecnológicas: Profa. Ana Cláudia Lucas

Coordenador do Mestrado em Engenharia Eletrônica e Computação: Prof. Eduardo Antonio César da Costa

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. Eduardo A.C. da Costa, e coorientador, Prof. Dr. Cláudio Diniz, por toda dedicação e orientação, pelos seus conhecimentos a mim transmitidos e por todas as revisões e colaborações essenciais para a realização deste projeto.

Aos meus pais, Vania e Paulo, que sempre me apoiaram e motivaram para minha educação e por todo suporte e incentivo nos momentos mais difíceis.

Ao meu grande parceiro, Leonardo Langone, por algumas ajudas técnicas e pelo carinho e compreensão nas horas de desespero.

Aos meus colegas de mestrado, em especial a Anderson Cardozo e Rafael Ferreira, pela grande parceria formada nas horas de estudos.

Ao Prof. Matheus Fonseca e colega Guilherme Paim por suas colaborações e sugestões no desenvolvimento deste projeto.

A todas aquelas pessoas que de alguma forma contribuíram, direta ou indiretamente, para a concretização deste trabalho.

RESUMO

O padrão de compressão de vídeo HEVC (High Efficiency Video Coding) é um dos mais novos padrões desenvolvidos. Ele alcança taxas de compressão de vídeo maiores em relação aos outros padrões já existentes e dobra a taxa de compressão comparado ao padrão anterior, o H.264/AVC (Advanced Video Coding). No codificador de vídeo, os módulos de Estimação de Movimento e Predição Intra estão presentes e utilizam intensamente as seguintes métricas de similaridade para decisão de modo de codificação: a SAD (Sum of Absolute Differences) e a SATD (Sum of Absolute Transformed Differences). O uso intenso destas métricas representa a maior parte da complexidade computacional do codificador de vídeo HEVC. O objetivo deste trabalho é explorar arquiteturas de hardware para as métricas de similaridade SAD e SATD utilizando técnicas de redução do consumo de potência. Para a métrica SATD foram realizados dois estudos: o primeiro estudo foca na exploração arquitetural em diferentes níveis de paralelismo da Transformada Hadamard 8x8; o segundo estudo visa a implementação de arquiteturas baseadas em múltiplos tamanhos da Transformada Hadamard. Como técnicas de redução de potência, foram utilizados somadores compressores nas arquiteturas de SATD. Também foram utilizados somadores compressores em diferentes recombinações em arquitetura da métrica SAD. Todas arquiteturas apresentadas neste trabalho foram implementadas em linguagem de descrição de *hardware* VHDL e sintetizadas para ASIC na tecnologia 45nm da *Nangate* utilizando a ferramenta *Cadence RTL Compiler*. A estimativa de potência das arquiteturas foi obtida utilizando entradas reais extraídas do *software* de referência do padrão HEVC. Foram realizadas análises comparativas entre estas arquiteturas, assim como comparações com arquiteturas estado-da-arte. As arquiteturas utilizando somadores/subtratores compressores em comparação às arquiteturas utilizando somadores da ferramenta de síntese apresentam uma redução de potência de 16,3% para a sequencial, 21,1% para a semi-paralela e 26,6% para a paralela para SATD baseada na HT 8x8 e 10,07% para a SATD baseada em múltiplos tamanhos da HT. Para a arquitetura de SAD utilizando somadores compressores 8:2 a redução de potência foi de 61,8%.

Palavras-chave: Codificação de vídeo. HEVC. Arquitetura de hardware. SATD. SAD. Somadores compressores.

Architectural Exploration to the Similarity Metrics for Video Encoders of the HEVC Standard

ABSTRACT

The HEVC video compression standard is one of the newest standards. It achieves higher compression ratio compared to existing standards and doubles the compression ratio of the previous standard, the H.264/AVC (Advanced Video Coding). In video encoder, the Motion Estimation and Intra Prediction modules are present and uses intensively the follow similarity metrics for mode decision: the Sum of Absolute Differences and the Sum of Absolute Transformed Differences. The intense use of these metrics represents the major part of the computational complexity of the HEVC video encoder. The aim of this work is to explore hardware architectures for SAD and SATD similarity metrics using power consumption reduction techniques. For the SATD metric two studies were performed: the first study focuses on the architectural exploration at different levels of parallelism of the 8x8 Hadamard Transform; the second study aims at the implementation of architectures based on multiple sizes of the Hadamard Transform. As power reduction techniques, adders compressor were used in the SATD architectures. Adders compressor were also used in different recombinations of SAD metric. All architectures presented in this work were implemented in hardware description language VHDL and synthesized for ASIC in Nangate's 45nm technology using the Cadence RTL Compiler tool. The power estimation of the architectures was obtained using real inputs extracted from the reference software of the HEVC standard. Comparative analyzes were performed between these architectures, as well as comparisons with architectures state-of-the-art. The architectures using adders/subtractors compressors compared to architectures using adder from the synthesis tool have a power reduction of 16.3 % for the sequential, 21.1 % for the semi-parallel and 26.6 % for the parallel for the SATD based on HT 8x8 and 10.07 % for SATD based on multiple sizes of HT. For the SAD architecture using 8: 2 compressor adders the power reduction was 61.8 %.

Keywords: video coding, SATD, SAD, adder compressors, HEVC.

LISTA DE ABREVIATURAS E SIGLAS

1-D	One Dimension
2-D	Two Dimension
ASIC	Application Specific Integrated Circuits
AVC	Advanced Video Coding
CSA	Carry Save Adder
CTC	Common Test Conditions
CTU	Coding Tree Units
CU	Coding Units
DSP	Digital Signal Processing
FME	Fractional-pixel Motion Estimation
FPGA	Field Programmable Gate Array
HEVC	High Efficiency Video Coding
HSI	Hue Saturation Intensity
HT	Hadamard Transform
HM	HEVC Test Model
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
IME	Integer Motion Estimation
ITU	International Telecommunication Union
MPEG	Moving Picture Experts Group
RGB	Red Green Blue
RTL	Register Transfer Language
PU	Prediction Unit
SAD	Sum of Absolute Differences

SATD Sum of Absolute Transformed Differences

SAV Sum of Absolute Values

SSD Sum of Square Differences

VCD Value Change Dump

VCEG Video Coding Experts Group

VHDL VHSIC Hardware Description Language

VHSIC Very High Speed Integrated Circuits

YCbCr Luma Chroma blue Chroma red

LISTA DE FIGURAS

Figura 2.1 Representação das amostras de luminância (círculos cinza), croma (círculos azuis) e crominância vermelha (círculos vermelhos), nos diferentes formatos de sub-amostragem. (a) formato 4:4:4; (b) formato 4:2:2; (c) formato 4:2:0.	18
Figura 3.1 Diagrama em blocos do codificador HEVC (CORRÊA, 2014)	22
Figura 3.2 Processo de Estimação de Movimento	23
Figura 3.3 Modos de Predição Intra: (a) H.264.AVC(SEIDEL et al., 2014)	24
Figura 4.1 Compressor 3:2. (a) Estrutura do bloco, (b) Estrutura interna em portas lógicas. .	29
Figura 4.2 Circuito somador de 8 bits de entrada utilizando compressor 3:2.....	29
Figura 4.3 (a) Somador compressor 4:2; (b) Somador compressor 4:2 composto por somadores completos.....	30
Figura 4.4 Somador compressor 4:2 utilizando multiplexadores (OKLOBDZIJA; VILLEGER; LIU, 1996).....	31
Figura 4.5 Circuito somador de 8 bits de entrada utilizando compressor 4:2.....	32
Figura 4.6 Subtrator compressor 4:2 modificado.....	32
Figura 4.7 Circuito somador/subtrator compressor 4:2 proposto.	33
Figura 4.8 Estrutura do subtrator/somador proposto para uma soma de quatro valores de 9 bits.....	34
Figura 4.9 Estrutura do somador compressor 8:2 usando: (a) a combinação dos compressores 3:2 e 4:2 (OKLOBDZIJA; VILLEGER; LIU, 1996); (b) compressores 4:2 (OKLOBDZIJA; VILLEGER; LIU, 1996).....	34
Figura 4.10 Circuito somador de 8 bits de entrada utilizando compressor 8:2.....	36
Figura 4.11 (a) Somador <i>Full Adder</i> - FA; (b) Circuito RCA com FA.....	37
Figura 4.12 Somador <i>full-adder</i> baseado em MUX.....	37
Figura 4.13 Operador <i>c</i> para o cálculo do prefixo.....	38
Figura 4.14 Somador Kogge-Stone (KOGGE; STONE, 1973).	39
Figura 5.1 Modelo da arquitetura de <i>hardware</i> da SATD.....	42
Figura 5.2 (a) Circuito interno da Hadamard 1-D; (b) Circuito interno da Hadamard 1-D com somador e subtrator compressor modificado; (c) Circuito interno da Hadamard 1-D com somador/subtrator compressor proposto.....	43
Figura 5.3 (a) Circuito interno do módulo SAV original; (b) Circuito interno do módulo SAV com compressor 8:2.....	44
Figura 5.4 Diagrama de blocos da arquitetura sequencial.	45
Figura 5.5 Diagrama de blocos da arquitetura semi-paralela.....	46
Figura 5.6 Diagrama de blocos da arquitetura totalmente paralela.....	47
Figura 5.7 Ilustração da divisão das matrizes.	48
Figura 5.8 SATD baseada em múltiplos tamanhos da HT.	49
Figura 5.9 SATD baseada em múltiplos tamanhos da HT usando compressores.	50
Figura 5.10 Diagrama em blocos da arquitetura da SAD: (a) usando o somador da ferramenta; (b) usando compressor 8:2.	51
Figura 6.1 Diagrama da metodologia de síntese.....	54
Figura 6.2 Potência de <i>leakage</i> por vídeo para todas versões de arquitetura de SATD.....	56
Figura 6.3 Potência dinâmica por vídeo para todas versões de arquitetura de SATD.	56
Figura 6.4 Potência total por vídeo para todas versões de arquitetura de SATD.....	57
Figura 6.5 Redução média de Potência total para cada arquitetura.	57
Figura 6.6 Comparação de Energia/ <i>Throughput</i> com arquitetura estado-da-arte (referência).	59

Figura 6.7 Potência Total por vídeo para todas as versões de arquitetura de SAD.....	62
Figura 6.8 Redução Média de dissipação de potência.	62

LISTA DE TABELAS

Tabela 4.1	Tabela Verdade do Somador Compressor 4:2.....	31
Tabela 4.2	Tabela Verdade do Somador Compressor 8:2.....	35
Tabela 5.1	Cálculo da Hadamard Horizontal	44
Tabela 5.2	Cálculo da Hadamard Vertical	45
Tabela 6.1	Resultados de Síntese	55
Tabela 6.2	Energia por operação de SATD	57
Tabela 6.3	Resultados de Síntese	59
Tabela 6.4	Resultados de Potência	60
Tabela 6.5	Resultados de Síntese	61
Tabela 6.6	Energia por operação de SAD	63

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivos	14
1.2 Principais Contribuições	15
1.3 Organização do Trabalho	15
2 COMPRESSÃO DE VÍDEO	16
2.1 Vídeo Digital	16
2.2 Espaço de Cores	16
2.2.1 Sub-Amostragem de cores	17
2.3 Redundância de Dados	19
2.4 Resumo	20
3 O PADRÃO DE COMPRESSÃO DE VÍDEO HEVC	21
3.1 Módulos do Padrão HEVC	21
3.1.1 Predição Inter	23
3.1.2 Predição Intra	24
3.1.3 Otimização Taxa-Distorção	24
3.2 Métricas de Similaridade SAD e SATD	25
3.2.1 Métrica SAD	26
3.2.2 Métrica SATD	26
3.3 Resumo	27
4 SOMADORES COMPRESSORES	28
4.1 Compressor 3:2	28
4.2 Compressor 4:2	30
4.2.1 Somador Compressor 4:2 Modificado	32
4.2.2 Somador/Subtrator Compressor 4:2 Proposto	33
4.3 Compressor 8:2	34
4.3.1 Aplicação de Somadores Eficientes na Linha de Recombinação	36
4.4 Resumo	39
5 ARQUITETURAS PROPOSTAS PARA AS MÉTRICAS SAD E SATD COM O USO DE SOMADORES COMPRESSORES	40
5.1 Exploração Arquitetural na Métrica SATD	40
5.1.1 Arquiteturas da SATD baseadas na HT 8x8	41
5.1.1.1 Arquitetura Sequencial	44
5.1.1.2 Arquitetura Semi-Paralela	46
5.1.1.3 Arquitetura Totalmente Paralela	47
5.1.2 Arquitetura da SATD com Múltiplos Tamanhos de Transformada Hadamard	48
5.2 Exploração Arquitetural na Métrica SAD	50
5.3 Resumo	52
6 RESULTADOS DE SÍNTESE E TRABALHOS RELACIONADOS	54
6.1 Metodologia de Síntese	54
6.2 Resultados para SATD - Exploração Arquitetural baseada na HT8x8	55
6.2.1 Trabalhos Relacionados com a Exploração Arquitetural da SATD	58
6.3 Resultados para SATD - Implementação Baseada em Múltiplos Tamanhos da HT	59
6.3.1 Trabalhos Relacionados com SATD Usando HTs de Múltiplos Tamanhos	60
6.4 Resultados para SAD	61
6.4.1 Trabalhos Relacionados com a Métrica SAD	63
6.5 Resumo	64
7 CONCLUSÕES E TRABALHOS FUTUROS	65
7.1 Trabalhos Futuros	66

REFERÊNCIAS..... 67

1 INTRODUÇÃO

A utilização de vídeos digitais se torna cada vez maior, sendo estes utilizados em diversas aplicações, tais como videoconferências, transmissões em broadcasting, vídeo pela Internet, etc. Um vídeo digital possui elevado volume de dados: uma sequência de 10 minutos de vídeo no formato Sony 4K (4096x2160 pixels) com 30 quadros por segundo e 24 bits por pixel requer 477 GB para ser armazenado (DINIZ, 2015). Desta forma, para que os vídeos digitais possam ser transmitidos com eficiência é necessário aplicar o processo de compressão de vídeos, que tem como objetivo a redução do número de bits necessários para representar um vídeo digital sem que haja perda de informações significativas para a representação do vídeo.

Os processos de compressão possibilitam a redução da quantidade de dados explorando as redundâncias na representação do vídeo e também as características do sistema visual humano para inserção de perdas na informação. A compressão de vídeo é de grande importância especialmente nos meios em que o volume de informação é muito elevado, como a TV digital, por exemplo. Como forma de solucionar este problema, foram desenvolvidos padrões de compressão de vídeo ao longo dos anos, dentre os principais: MPEG-1 (ISO/IEC, 1993), MPEG-2 (ITU-T, 1994), MPEG-4 (ISO/IEC, 1999) e padrões mais novos como o H.264/AVC (Advanced Video Coding) (ITU-T; 1, 2011) e o HEVC (High Efficiency Video Coding) (ITU-T; ISO/IEC, 2013).

O HEVC é um dos mais novos padrões de compressão de vídeo e foi desenvolvido com o objetivo de dobrar a taxa de compressão em relação aos demais padrões existentes. O seu desenvolvimento foi inspirado pelo aumento de aplicações utilizando vídeo, pela popularidade da TV digital de alta definição bem como o aparecimento de novos meios de transmissão. Estas aplicações demandam um padrão com eficiência de codificação bem maior do que os existentes. Alcança a melhor e mais eficiente compressão para uma grande variedade de aplicativos, como broadcast, DVD, videoconferência, vídeo sob demanda, transmissão e mensagens multimídia.

Para se obter todas essas vantagens seria difícil a implementação do projeto deste padrão sem que existisse um aumento na complexidade das operações. Este padrão de compressão agrega então, uma complexidade computacional muito elevada pois são empregados tamanhos de blocos maiores, de até 64x64 *pixels*, e abrange mais modos de predição comparado ao padrão anterior H.264/AVC. Este aumento de complexidade dificulta, pelo menos na tecnologia atual, a utilização de codecs (codificadores/decodificadores) HEVC, quando as resoluções são elevadas (por exemplo, 4K) e quando se deseja tempo real, com no mínimo 30 quadros por segundo.

Essa grande dificuldade do aumento da complexidade encontrada nesse padrão de codi-

ficação de vídeo é a grande motivação para grupos de pesquisas e desenvolvimento, do mundo todo, focalizar o estudo em técnicas de otimizações algorítmicas e de implementações em hardware, visando atender às necessidades das devidas aplicações.

Juntamente com a crescente utilização dos vídeos digitais, também houve um aumento significativo do uso de dispositivos móveis operados por baterias, tais como palmtops, câmeras e filmadoras digitais portáteis, telefones celulares entre outros. Aplicações com vídeos digitais para estes dispositivos serão cada vez mais utilizadas. Para este tipo de aplicação, um fator de extrema importância, além de um codec eficaz, é a dissipação de potência, pois dispositivos móveis não podem apresentar aquecimento elevado em seus circuitos, para não comprometer o seu funcionamento e a sua utilização pelos usuários. Além disso, o aumento do tempo de vida útil da bateria é outro fator motivador para a redução da dissipação de potência em dispositivos móveis operados por bateria que possam utilizar vídeos digitais. Neste contexto, torna-se fundamental a implementação em hardware de blocos do padrão que corresponda às exigências de desempenho, utilização de recursos e principalmente de consumo de energia.

Diante do exposto, surge a motivação deste trabalho em estudar e implementar técnicas de redução de potência existentes na literatura, bem como propor novas técnicas, para blocos do codec do padrão HEVC, e assim obter ganhos em desempenho e consumo de energia nos circuitos implementados, viabilizando a sua utilização, como por exemplo, em dispositivos móveis.

1.1 Objetivos

Com o estudo realizado neste trabalho foi possível localizar no padrão HEVC alguns blocos onde podem ser dadas contribuições para o aumento de desempenho e redução do consumo de potência e energia. Este é o caso das métricas SATD (*Sum of Absolute Transformed Differences*) e SAD (*Sum of Absolute Differences*) que são métricas de similaridade utilizadas nos módulos de Estimação de Movimento e na Predição Intra.

Este trabalho apresenta explorações arquiteturais nos blocos de hardware das métricas de SAD e SATD visando à combinação ótima entre velocidade de operação e consumo de energia. Além disto, devido ao grande número de operadores aritméticos de soma envolvidos nestes blocos de hardware, este trabalho foca na aplicação de somadores compressores eficientes que realizam as somas simultâneas de mais de dois operandos.

1.2 Principais Contribuições

As principais contribuições deste trabalho estão voltadas às explorações arquiteturais, em diferentes níveis de paralelismo, da SATD explorando múltiplos tamanhos de transformada Hadamard (Hadamard Transform - HT), para o projeto de hardware do SATD e ao projeto de arquiteturas utilizando somadores compressores em módulos de hardware para cálculo do SAD e SATD, como técnica para redução do consumo de potência.

As principais contribuições deste trabalho são:

- Exploração arquitetural em três níveis de paralelismo (sequencial, semi-paralela e paralela) para a arquitetura em hardware da SATD baseada na Transformada de Hadamard 8x8 e aplicação de somadores compressores nas três arquiteturas;
- Exploração de diferentes combinações para os compressores 8:2, baseados em compressores 4:2 e 3:2, assim como a aplicação destes na arquitetura da métrica SAD;
- Arquitetura de hardware para a métrica SATD baseada em múltiplos tamanhos da Transformada de Hadamard e empregando somadores compressores;
- Implementação de uma nova estrutura para somadores compressores 4:2;

1.3 Organização do Trabalho

No capítulo 2 é apresentada uma breve introdução aos conceitos básicos de compressão de vídeo. O capítulo 3 apresenta um histórico do padrão de compressão de vídeo HEVC, os módulos que o compõe, e uma visão geral das métricas SATD e SAD. No capítulo 4 são apresentados os conceitos de somadores compressores, assim como uma estrutura modificada do somador compressor 4:2 e uma nova estrutura desenvolvida neste trabalho para o compressor 4:2. No capítulo 5 é realizada uma breve introdução às HTs 2x2, 4x4 e 8x8, assim como são apresentadas as arquiteturas desenvolvidas para as métricas de SATD e SAD utilizando compressores. No capítulo 6 são apresentados os resultados de síntese para estas arquiteturas e a comparação realizada com trabalhos encontrados na literatura. No capítulo 7 são apresentadas as principais conclusões do trabalho e sugestões de trabalhos futuros.

2 COMPRESSÃO DE VÍDEO

A compressão de um vídeo digital permite reduzir a quantidade de dados necessária para representá-lo, diminuindo sua taxa de transmissão ou o espaço para seu armazenamento. Este capítulo apresenta inicialmente alguns conceitos relacionados aos vídeos digitais e a seguir uma breve introdução aos aspectos gerais da compressão de vídeo.

2.1 Vídeo Digital

O processo de captação e digitalização de vídeo resulta em um sinal de vídeo digital, composto por uma sequência de quadros (*frames*) amostrados temporalmente, que, quando reproduzidos, geram a sensação de movimento ao observador. Em um sinal de vídeo digital, existe também uma amostragem espacial, resultando em uma imagem formada por pontos, denominados *pixels*. Os *pixels*, organizados nas direções horizontal e vertical, representam um elemento luminoso da imagem que contém uma cor específica (RICHARDSON, 2004).

Hoje em dia os vídeos digitais possuem resoluções cada vez maiores, por exemplo, resoluções 4k Ultra HD (3840 x 2160 *pixels*), se comparado às maiores resoluções disponíveis no mercado há poucos anos atrás, como a resolução Full HD (1920 x 1080). Assim, as sequências de vídeo necessitam de um grande volume de dados para serem representadas. Porém, estes dados apresentam um grau de redundância muito grande, por exemplo, muita semelhança entre os quadros do vídeo, sendo desnecessária a representação de uma boa parte desta quantidade de dados, como será discutido a seguir.

2.2 Espaço de Cores

A luz visível é parte do espectro eletromagnético que, por ser contínuo, abrange um número infinito de comprimentos de onda. Cada comprimento de onda é percebido como uma cor diferente pelo sistema visual humano. A interpretação das cores é dada pelos cones, que são células sensíveis à cor, localizadas na retina, as quais respondem a comprimentos de onda das cores vermelha, verde e azul. Estas três cores são consideradas as cores primárias em fontes de luz, e a partir da combinação de intensidades distintas dessas cores é que o sistema visual humano é capaz de identificar uma ampla gama de cores. Na retina também estão localizados os bastonetes, que são as células sensíveis aos níveis de intensidade luminosa. O sistema visual

humano não consegue distinguir mais do que duas dúzias de tons de cinza, que na verdade, indicam a intensidade luminosa da imagem (GONZALEZ; WOODS, 2000).

Os vídeos digitais são normalmente representados por somente três sinais de cores. O sistema para a representação das cores na forma digital é chamado de espaço de cores, cuja utilização é de suma importância para a eficiência da codificação de vídeo, visto que elas é que formarão a imagem. São inúmeros os espaços de cores utilizados para representar imagens digitais, se destacando os seguintes: RGB, HSI e YCbCr (SHI; SUN, 1999). O espaço de cores mais comum é o RGB, cuja sigla vem do inglês: Red (vermelho), Green (verde) e Blue (azul). Ele representa as cores primárias em três matrizes diferentes, como o próprio nome já define. Este é o sistema utilizado em TVs e monitores coloridos. Porém não é muito indicado na codificação de vídeos, pois apesar de ser um espaço de cor adequado para os dispositivos de visualização, essa representação não é tão propícia para a codificação de vídeo, já que existe uma alta correlação entre os três canais de cor, resultando em uma certa redundância quando usada tal representação (MANOEL et al., 2007).

No espaço de cores YCbCr, as componentes utilizadas são: luminância (Y), croma-
nância azul (Cb) e croma-
nância vermelha (Cr), onde a primeira é a responsável por representar a intensidade de luz e as outras duas responsáveis pela colorização. Este espaço de cor é mais indicado para a codificação de vídeos digitais, pois apresenta algumas vantagens sobre o RGB. Dentre essas vantagens, destacam-se: i) não apresentar uma alta redundância como no sistema RGB; ii) as informações de cor e brilho estão completamente separadas, podendo assim, serem tratadas de forma diferenciada pelos compressores de imagens estáticas e vídeos.

2.2.1 Sub-Amostragem de cores

O sistema visual humano é muito mais sensível à intensidade de luz do que às cores, podendo assim utilizar uma menor resolução na amostragem dos sinais que representam as cores, quase sem prejuízos à qualidade subjetiva das imagens.

Desta forma, os padrões de compressão de imagens estáticas e vídeos exploraram esta característica do sistema visual humano para aumentar a eficiência de codificação através da redução da taxa de amostragem dos componentes de croma-
nância em relação aos componentes de luminância (RICHARDSON, 2004).

Esta redução da taxa de amostragem entre os componentes de croma-
nância e luminância é chamada de sub-amostragem de cores, aplicada sobre o espaço de cores YCbCr dos compressores de vídeos atuais. Existe uma nomenclatura padrão para as diferentes formas de sub-

amostragem comumente utilizadas em codificação de vídeo que são os formatos 4:4:4, 4:2:2 e 4:2:0, apresentados na figura 2.1.

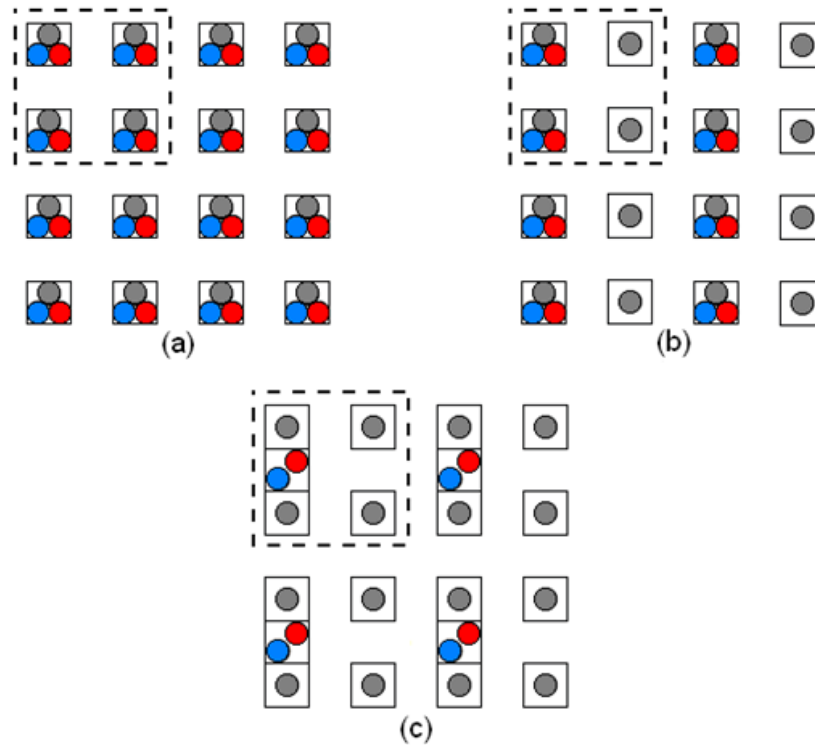


Figura 2.1 – Representação das amostras de luminância (círculos cinza), crominância azul (círculos azuis) e crominância vermelha (círculos vermelhos), nos diferentes formatos de sub-amostragem. (a) formato 4:4:4; (b) formato 4:2:2; (c) formato 4:2:0.

- No formato 4:4:4, para cada quatro amostras de luminância (Y), existem quatro amostras de crominância azul (Cb) e quatro amostras de crominância vermelha (Cr) correspondentes, ou seja, não há aplicação da sub-amostragem neste caso;
- No formato 4:2:2, a resolução dos sinais de cores são reduzidos pela metade na direção horizontal, ou seja, para cada quatro amostras de luminância, existem duas amostras para crominância azul (Cb) e duas amostras para crominância vermelha (Cr). Porém, verticalmente, a resolução das amostras de luminância e crominância é a mesma;
- No formato 4:2:0, a resolução dos sinais de crominância são reduzidos pela metade, tanto horizontalmente como verticalmente, isto é, para cada quatro amostras de luminância (Y), existe uma amostra para crominância azul (Cb) e uma amostra para crominância vermelha (Cr). A nomenclatura 4:2:0 é usada por motivos históricos, pois os números não representam a relação lógica entre os componentes de cor, a qual deveria ser 4:1:1 (RICHARDSON, 2004).

Para uma eficiente compressão dos vídeos, a sub-amostragem de cor se apresenta de maneira muito significativa, tendo em vista que uma parte da informação da imagem é simplesmente descartada, sem causar um impacto visual aparente. Levando em conta o formato de sub-amostragem 4:2:0, um vídeo YCbCr utilizará somente a metade das amostras de crominância necessárias para um vídeo RGB ou para um vídeo YCbCr no formato 4:4:4. Isso implica em uma taxa de compressão de 50%, considerando a sub-amostragem.

2.3 Redundância de Dados

Considera-se redundante aquele dado que não tem nenhuma contribuição com novas informações relevantes para a representação do vídeo. Com isso, a compressão de vídeos procura uma redução de uma quantidade de dados que são considerados redundantes na representação computacional das informações do vídeo. Os padrões de compressão de vídeo exploram a redundância dos vídeos digitais para redução do volume de dados para representar o vídeo. Há três tipos de redundância na compressão de vídeo: Redundância espacial; Redundância temporal; Redundância entrópica.

- **Redundância Espacial:** chamada também de redundância intra-quadro ou *interpixel*, se origina da correlação entre os *pixels* espacialmente distribuídos em um *frame* do vídeo. Esta correlação existe também no domínio das frequências. A correlação no domínio espacial pode ser reduzida através da predição intra-quadro, que será detalhada a seguir. A correlação no domínio das frequências é reduzida através da quantização, que também é detalhada a seguir. A quantização é um processo irreversível que gera perdas de informação de vídeo. O objetivo é que as perdas geradas pela etapa da quantização não gerem um impacto na qualidade da imagem (AGOSTINI, 2007).
- **Redundância Temporal:** também conhecida como redundância inter-quadros, se deve à correlação existente entre *frames* temporalmente próximos em um vídeo. Muitas vezes os blocos, formados pelos pixels, não mudam de um quadro para outro, por exemplo, em um vídeo que possui um fundo estático. Por outro lado, outros blocos apresentam uma pequena variação de valores, originados de uma variação da iluminação, por exemplo. Por fim, é possível que um bloco simplesmente não altere seus valores, mas sim, desloque-se de um quadro para outro, como exemplo, a cena de um objeto em movimento. A redundância temporal é explorada pelos padrões de compressão de vídeo atuais tendo em vista que nela ocorre a maior parte de informação redundante.

- **Redundância Entrópica:** este tipo de redundância tem relação com a probabilidades de ocorrência dos símbolos do vídeo codificado. A entropia é uma medida da quantidade média de informação transmitida por símbolo do vídeo (SHI; SUN, 1999). Quanto maior a probabilidade de ocorrência de um símbolo, menor a quantidade de informação nova transmitida por símbolo. Deste modo, os codificadores de vídeo atuais exploram a redundância entrópica para transmitir o máximo de informação possível por símbolo codificado. O objetivo final é transmitir a mesma informação com menor número de bits. São utilizados diferentes técnicas de compressão sem perdas para atingir este objetivo.

2.4 Resumo

Neste capítulo foram apresentados alguns conceitos básicos sobre compressão de vídeo. O próximo capítulo apresentará um breve estudo sobre o padrão de compressão de vídeo HEVC, abordando uma visão geral sobre seu histórico, seus principais conceitos e um breve detalhamento dos blocos onde estão inseridas as métricas de similaridade abordadas neste trabalho.

3 O PADRÃO DE COMPRESSÃO DE VÍDEO HEVC

A padronização de compressores de vídeo digital é fundamental para permitir que equipamentos que manipulam vídeos digitais, produzidos por diferentes fabricantes, possam se comunicar entre si.

O padrão de compressão de vídeo HEVC (*High Efficiency Video Coding*) é o padrão de compressão estado-da-arte (ITU-T; ISO/IEC, 2013), o qual praticamente dobra a eficiência de compressão comparado com o padrão anterior, o H.264/AVC (*Advanced Video Coding*) (ITU-T; 1, 2011)(SULLIVAN et al., 2012). Atualmente, os principais órgãos responsáveis por tal padronização são o ITU-T Video Coding Experts Group (VCEG) e o ISO/IEC Moving Picture Experts Group (MPEG). A primeira edição do HEVC foi finalizada em janeiro de 2013. O HEVC foi criado para melhorar substancialmente a eficiência de codificação em comparação com H.264/AVC, isto é, reduzir os requisitos de taxa de bits pela metade, mantendo a mesma qualidade de imagem, à custa de uma maior complexidade computacional (SULLIVAN et al., 2012). O suporte para vídeos de resolução maior (3840x2160 pixels, ou maiores), a flexibilidade no particionamento de blocos e a proposta de novas ferramentas de codificação são as principais melhorias do HEVC em relação ao padrão anterior, o H.264/AVC.

3.1 Módulos do Padrão HEVC

No padrão HEVC, foram introduzidas novas técnicas para alcançar maior compressão de vídeo. Uma delas diz respeito ao uso de tamanhos de blocos maiores do que o padrão anterior, de até 64x64 *pixels*. Uma nova estrutura flexível de particionamento de blocos também é proposta, a qual permite dividir estes blocos hierarquicamente até blocos elementares de 8x8 *pixels*. Primeiramente, o quadro de entrada é dividido em CTUs (*Coding Tree Units*), que tem tamanho menor ou igual a 64x64 *pixels*, no qual este tamanho é definido para cada vídeo pelo codificador de vídeo. Cada CTU pode ser subdividida em CUs (*Coding Units*), até um tamanho mínimo de 8x8 *pixels*, através de uma estrutura *quadtree* recursiva. Essas CUs ainda podem ser divididas em pequenas unidades de predição (PUs – *Prediction Units*). Cada unidade de codificação pode conter uma ou mais PUs, e cada uma das PUs pode ser do tamanho da CU ou particionadas em unidades menores de no mínimo 4x4 *pixels*. Dessa forma, pode-se notar que o tamanho dos blocos de predição no HEVC podem variar de 4x4 até 64x64 amostras. A Figura 3.1 apresenta o diagrama em blocos do codificador HEVC.

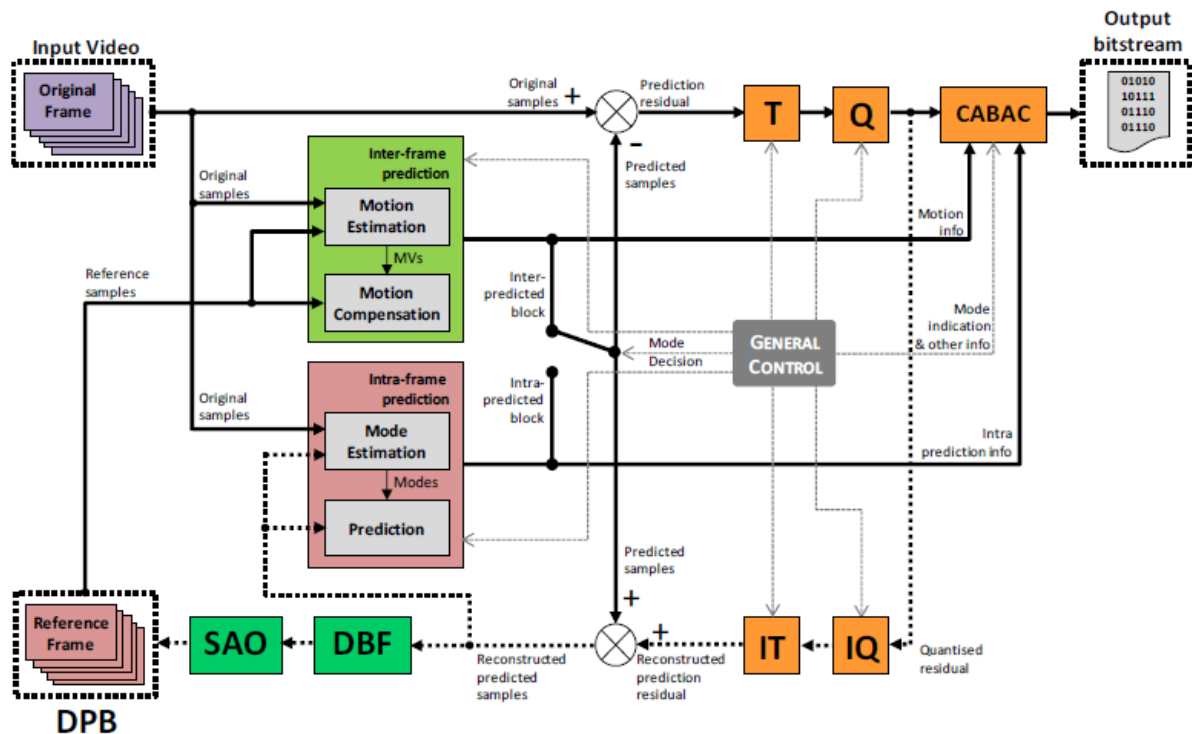


Figura 3.1 – Diagrama em blocos do codificador HEVC (CORRÊA, 2014)

O codificador HEVC é basicamente composto pelos módulos de Predição Intra e Inter, Transformadas, Quantização, Transformada inversa, Quantização inversa, Codificação de Entropia e Filtro de Deblockagem. Na predição Inter, o codificador faz uma predição de cada *frame* a partir dos *frames* previamente codificados utilizando os módulos de Estimção de Movimento (ME – *Motion Estimation*) e Compensação de Movimento (MC – *Motion Compensation*). No módulo de Predição Intra, para fazer a predição de cada bloco original são utilizadas amostras dos blocos vizinhos já reconstruídos a partir do *frame* atual, de acordo com o modo escolhido na Decisão de Modo (*Mode Decision*), como mostrado na Figura 3.1. Após o processo de predição, é calculado o resíduo (diferença entre blocos original e predito), o qual passa pelos módulos das transformadas, quantização e codificação de entropia (CABAC na Figura 3.1). Outras informações, como modos de predição e vetores de movimento são codificados pelo módulo CABAC, resultando no fluxo de bits do vídeo codificado, denominado *bitstream*. Nas próximas seções são detalhados os módulos de Predição Inter e Intra, visto que são nestes módulos que as métricas de similaridade são aplicadas (foco deste trabalho).

3.1.1 Predição Inter

O módulo de Predição Inter é composto pelos blocos de Estimação de Movimento (ME) e Compensação de Movimento (MC). Neste módulo, o objetivo é gerar uma predição a partir das imagens anteriores, chamadas imagens de referência. A Estimação de Movimento busca o melhor casamento entre dois blocos, isto é, qual o bloco na imagem de referência é mais parecido com o bloco na imagem atual. A Figura 3.2 mostra este processo.

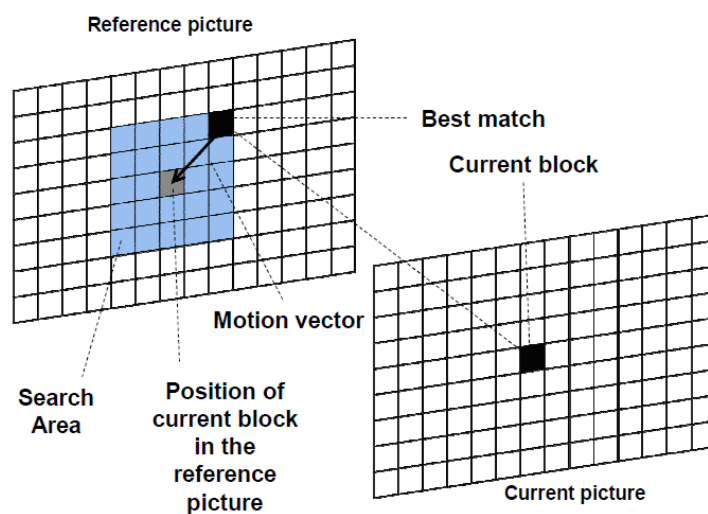


Figura 3.2 – Processo de Estimação de Movimento
(DINIZ, 2015)

Um Vetor de Movimento (MV – *Motion Vector*) é gerado quando o melhor casamento entre os blocos é encontrado. O vetor de movimento indica o deslocamento entre a posição do bloco atual e o bloco escolhido na imagem de referência e é codificado pela codificação de entropia e após enviado ao *bitstream* juntamente com o resíduo. O módulo de Compensação de Movimento é presente também no decodificador de vídeo e realiza o processo inverso da estimação de movimento, gerando um bloco reconstruído a partir de um vetor de movimento.

A busca da ME é conduzida por um algoritmo de *block matching*, o qual normalmente é otimizado de acordo com o objetivo específico estabelecido pelo projetista (desempenho, qualidade, potência e/ou área). Por ser uma ferramenta não padronizada, um grande número de implementações de algoritmos da ME são encontrados na literatura. Independente destes algoritmos de busca, algumas métricas de qualidade para determinar o melhor casamento de entre blocos são necessárias, como será detalhado nas próximas seções.

3.1.2 Predição Intra

O módulo da Predição Intra é responsável por gerar a predição de blocos candidatos, os quais são baseados em informações de blocos vizinhos da imagem atual (ou seja, a imagem capturada pela câmera) e que já foram processados e reconstruídos. Busca-se então, entre os blocos gerados, um que tenha a maior similaridade com o bloco atual. O padrão HEVC abrange mais modos de predição do que o H.264/AVC, chegando a ter 35 diferentes modos de predição (SULLIVAN et al., 2012) como pode ser visto na Figura 3.3.

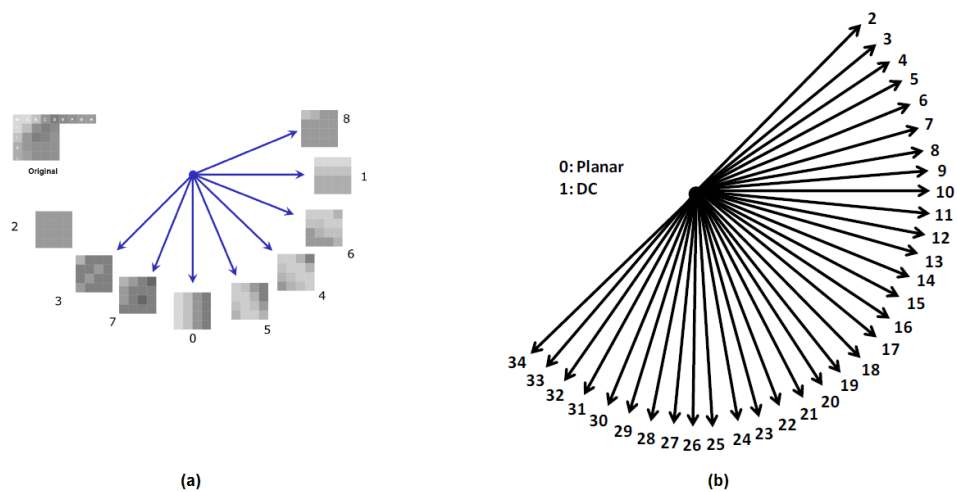


Figura 3.3 – Modos de Predição Intra: (a) H.264.AVC(SEIDEL et al., 2014)
; (b) HEVC(CORRÊA, 2014)

Para determinar quais blocos são mais ou menos similares, dentro de universo de busca pré-determinado, é necessária a utilização de métricas de similaridade (ou distorção). Algumas métricas eficientes devem ser aplicadas na etapa de decisão do modo na predição, como por exemplo a SATD e SAD que serão discutidas nas próximas seções.

3.1.3 Otimização Taxa-Distorção

Com a existência de um grande número de modos de codificação, é necessária a utilização de um método de otimização da taxa-distorção (RDO - *Rate-Distortion Optimization*) para selecionar o melhor modo de codificação para cada bloco a ser codificado. Em (WIEGAND et al., 2003) é proposto um método que utiliza a minimização de Lagrange, o qual é o mais utilizado para melhorar a taxa de compressão (taxa entre vídeo **raw** e vídeo codificado) com uma perda controlada na qualidade de vídeo de saída. A equação pode ser vista em (3.1), onde D é a

distorção (perda de informação no processo de codificação), R é a taxa do vídeo codificado (em bits/segundo), e J é o custo que deve ser minimizado na escolha do melhor modo.

$$J = D + \lambda \cdot R \quad (3.1)$$

Existem alguns métodos alternativos para a realização da otimização de custos da R-D, por exemplo, poder minimizar a distorção de nível de *frame* ou minimizar uma distorção média de *frame*. Estes métodos não são computacionalmente práticos, ficando sujeitos a uma quantidade significativa de complexidade e atraso (SULLIVAN et al., 2012). Algumas métricas para estimar esta distorção são utilizadas, tais como a soma das diferenças ao quadrado (SSD - *Sum of Squared Differences*), soma das diferenças absolutas (SAD - *Sum of Absolute Differences*) e soma das diferenças absolutas transformadas (SATD - *Sum of Absolute Transformed Differences*).

3.2 Métricas de Similaridade SAD e SATD

No módulo de Estimação de Movimento e no módulo de Predição Intra algumas métricas são utilizadas para estimar a distorção entre dois blocos, no estágio de decisão de modo dos codificadores de vídeo. A soma das diferenças absolutas (SAD) é a soma dos valores absolutos aplicados no bloco residual (diferença entre o bloco de vídeo original e o bloco de vídeo de referência). A soma das diferenças ao quadrado (SSD) é a soma dos valores ao quadrado do bloco residual. A soma das diferenças transformadas absolutas (SATD) é a soma dos valores dos coeficientes obtidos após a Transformada de Hadamard ser aplicada ao bloco residual.

O padrão HEVC possui um maior número de modos de predição e tamanhos de unidades de predição (PU) comparado ao H.264/AVC. A decisão de modo mais precisa seria feita utilizando a estimativa de distorção baseada na SSD (SZE VIVIENNE; BUDAGAVI,). Porém, esta métrica, mesmo sendo mais eficiente para estimar a distorção, é pouco utilizada devido ao alto nível de complexidade, pois exige operações de multiplicação que são mais custosas do que as operações de soma. As métricas SAD e SATD necessitam somente de operações de soma e subtração, portanto, são mais usadas na prática. Como este trabalho tem foco na implementação em hardware destas duas métricas (SAD e SATD), elas são melhor detalhadas nas duas seções seguintes.

3.2.1 Métrica SAD

A SAD é a métrica mais simples, sendo normalmente aplicada nas etapas executadas mais frequentemente executadas nos codificadores de vídeo. Normalmente é utilizada dentro da estimação de movimento inteira (*Integer Motion Estimation* - IME), para determinar o melhor casamento entre dois blocos de vídeo, mas também pode ser utilizada na estimação de movimento fracionária (*Fractional-pixel Motion Estimation* - FME) (DINIZ, 2015) ou na Predição Intra para determinar o melhor modo intra. Esta métrica representa até 12% do tempo de execução de todo o codificador HEVC em software quando usada somente para a IME (VANNE et al., 2012).

A equação (3.2) apresenta a fórmula genérica para o cálculo da SAD, onde O é o bloco original, R é o bloco de referência e m e n são as dimensões dos blocos em amostras.

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |O_{i,j} - R_{i,j}| \quad (3.2)$$

Por apresentar uma baixa complexidade computacional, sua implementação em hardware é simples. Sua desvantagem é uma baixa eficiência para estimação da distorção quando comparadas a outras métricas mais eficientes como a SSD e SATD.

3.2.2 Métrica SATD

A métrica SATD é uma boa relação entre complexidade e estimação de distorção, visto que essa métrica é mais simples do que a SSD e mais eficiente do que a SAD. Por esta razão, a métrica de SATD é normalmente aplicada nos codificadores de vídeo modernos nas etapas de decisão do modo na predição intra e na estimação de movimento fracionária. Por exemplo, habilitando a SATD na estimação de movimento do codificador HEVC, esta métrica reduz a taxa de bit em 2.2% e aumenta a qualidade de vídeo em 0,16 dB de PSNR (CORREA et al., 2012).

As implementações de hardware para o padrão HEVC frequentemente usam o módulo SATD baseado na Transformada de Hadamard 4x4. Poucos trabalhos exploram arquiteturas de hardware para o cálculo da SATD baseado na Transformada de Hadamard 8x8. O software de referência do padrão HEVC (HM) (FRAUNHOFER, 2015) utiliza a SATD baseada na Transformada de Hadamard 8x8 nos blocos de vídeo 8x8 ou maiores nas etapas de estimação de movimento e na predição intra. A SATD é um dos módulos com maior tempo de execução em

um codificador HEVC em software (DINIZ, 2015). Portanto, a exploração de arquiteturas eficientes de hardware para SATD, visando à eficiência de energia, é muito importante para uma eficiência energética do padrão HEVC como um todo.

Na equação (3.3) é apresentada a equação para o cálculo da SATD, onde $HT_{(i,j)}$ são os coeficientes do bloco obtido após a Transformada de Hadamard de duas dimensões (2-D) ser aplicada no bloco residual.

$$SATD = \sum_{i,j} |HT_{(i,j)}| \quad (3.3)$$

Por terem um foco muito importante neste trabalho, as HTs utilizadas nesta métrica SATD, serão discutidas no capítulo das arquiteturas implementadas.

3.3 Resumo

Neste capítulo foram apresentados alguns conceitos do padrão HEVC. Foram mostrados os módulos que compõem o padrão, bem como algumas características e conceitos associados a estes módulos. Dentre os módulos apresentados, destacam-se a Predição Intra e Estimação de Movimento, onde estão presentes as métricas de similaridade SAD e SATD (foco deste trabalho). O próximo capítulo será dedicado ao estudo de somadores compressores e a apresentação de novos somadores compressores propostos no escopo deste trabalho.

4 SOMADORES COMPRESSORES

O circuito somador é um dos operadores aritméticos mais básicos, com ampla utilização em diversos circuitos digitais. Entre os circuitos de aplicação para os somadores, podem-se citar os circuitos multiplicadores, que são amplamente utilizados em circuitos da área de Processamento Digital de Sinais (DSP – *Digital Signal Processing*). Para essa aplicação, uma das principais etapas é a adição de produtos parciais, pois é essa etapa que determina a velocidade de operação do multiplicador. Desta forma, torna-se importante a utilização de circuitos somadores eficientes que possam realizar essa etapa de forma eficiente.

Outra aplicação que requer a utilização de circuitos somadores eficientes, e alvo deste trabalho, é no *hardware* dedicado para compressão de vídeo do padrão HEVC. Em particular, as métricas de SAD e SATD apresentam um elevado número de operadores de soma e subtração. Neste trabalho, os operadores eficientes de soma e subtração, são aplicados na árvore de somas que compõe a SAD e a SATD e também nas HTs que compõem a SATD.

Os circuitos somadores eficientes utilizados neste trabalho, levam em consideração a possibilidade da soma de vários operandos simultaneamente. Neste contexto, o somador *Carry-Save* (KIM; JAO; TJIANG, 1998) é o circuito mais utilizado, pois realiza a soma de 3 números simultaneamente. Neste trabalho, utilizam-se circuitos somadores/compressores, como forma de aumentar o número de somas simultâneas em operações que envolvam um elevado número destes operadores aritméticos. Embora existam vários tamanhos de circuitos compressores encontrados na literatura, neste trabalho foram utilizados apenas compressores 3:2, 4:2 e 8:2, pois são os mais indicados às métricas SATD e SAD. A seguir são apresentados os diferentes tipos de somadores compressores utilizados neste trabalho.

4.1 Compressor 3:2

O circuito compressor 3:2, mostrado na Figura 4.1-(a), apresenta três entradas e duas saídas, onde as entradas A, B, C e a saída Sum apresentam o mesmo peso (j) e a saída *Carry* apresenta peso de um bit maior ($j+1$). A estrutura interna do compressor 3:2 é apresentada na Figura 4.1-(b). O atraso deste circuito é dado por duas portas XOR. As equações 4.1 e 4.2 expressam as saídas *Sum* e *Carry*.

$$Sum = A \oplus B \oplus C_{in} \quad (4.1)$$

$$Carry = [(A \oplus B) \cdot C_{in}] + [(\overline{A \oplus B}) \cdot C_{in}] \quad (4.2)$$

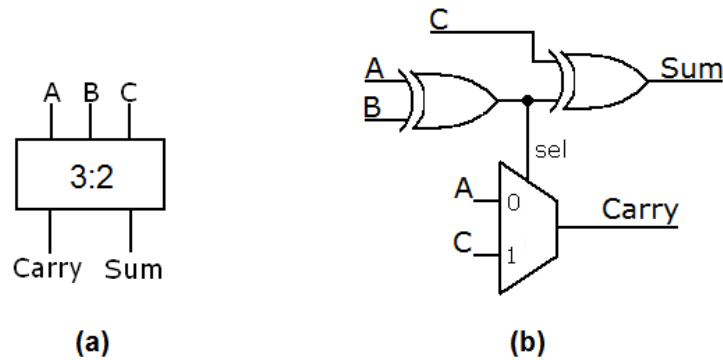


Figura 4.1 – Compressor 3:2. (a) Estrutura do bloco, (b) Estrutura interna em portas lógicas.

O resultado final de soma, *Sum*, do compressor 3:2 (Soma= A+B+C) é dada na equação 4.3. A estrutura de portas lógicas deste compressor também pode ser usada como um somador completo (*full-adder* baseado em multiplexador), quando a entrada C é utilizada como entrada de *carry*.

$$S = Sum + 2Carry \quad (4.3)$$

Um exemplo da utilização dos compressores 3:2 em uma soma de três números de 8 bits pode ser visto na Figura 4.2. Deve-se observar que são necessários circuitos somadores para a recombinação dos valores parciais das somas anteriores, visto que há a propagação de *carry* de um módulo de soma do compressor para o próximo bloco.

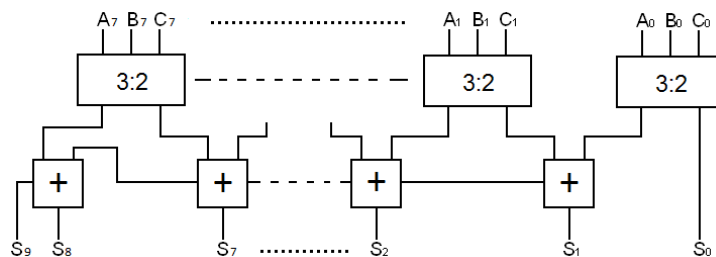


Figura 4.2 – Circuito somador de 8 bits de entrada utilizando compressor 3:2.

4.2 Compressor 4:2

O circuito compressor 4:2 é formado pela conexão em série de dois somadores completos (*full-adders*), como mostra a Figura 4.3 (WEINBERGER, 1981). Observa-se que esta estrutura apresenta cinco entradas e três saídas, onde, A, B, C, D, C_{in} e Sum apresentam o mesmo peso (j). Entretanto, as saídas *Carry* e C_{out} apresentam peso maior (j+1).

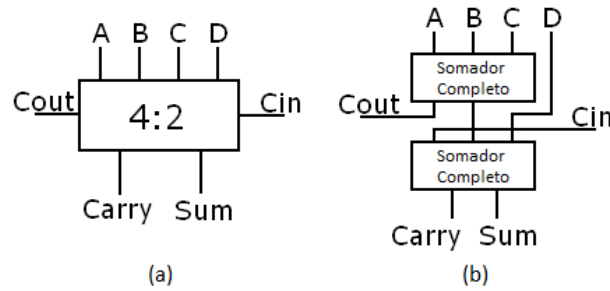


Figura 4.3 – (a) Somador compressor 4:2; (b) Somador compressor 4:2 composto por somadores completos.

Nota-se que a saída C_{out} não é função de C_{in} , e, portanto, não ocorre a propagação do *carry*, o que é benéfico para o aumento de desempenho desta estrutura. As equações (4.4), (4.5) e (4.6) expressam as saídas *Sum*, C_{out} e *Carry* e a Tabela 4.1 mostra a tabela verdade do somador compressor 4:2.

$$Sum = \{[(A \oplus B) \oplus C] \oplus D\} C_{in} \quad (4.4)$$

$$C_{out} = A.B + A.C + B.C \quad (4.5)$$

$$Carry = [(A \oplus B) \oplus C] \cdot (D + C_{in}) + D.C_{in} \quad (4.6)$$

A tabela verdade do somador compressor 4:2 expressa resultados parciais de soma. Para se obter a soma correta dos valores das entradas, torna-se necessário o uso da equação (4.7).

$$Soma = Sum + 2(C_{out} + Carry) \quad (4.7)$$

Por exemplo, se todas as entradas estiverem em nível lógico 1, inclusive a entrada C_{in} , a resposta parcial da tabela verdade será 111, que representa o valor decimal 7. Aplicando-se a equação 4.7, logo a soma correta será obtida, ou seja, $1 + 2(1 + 1) = 5$.

Tabela 4.1 – Tabela Verdade do Somador Compressor 4:2

				Cin=0			Cin=1		
A	B	C	D	Cout	Carry	Sum	Cout	Carry	Sum
0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0	1	0
0	0	1	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0	1	1
0	1	0	0	0	0	1	0	1	0
0	1	0	1	0	1	0	0	1	1
0	1	1	0	1	0	0	1	0	1
0	1	1	1	1	0	1	1	1	0
1	0	0	0	0	0	1	0	1	0
1	0	0	1	0	1	0	0	1	1
1	0	1	0	1	0	0	1	0	1
1	0	1	1	1	0	1	1	1	0
1	1	0	0	1	0	0	1	0	1
1	1	0	1	1	0	1	1	1	0
1	1	1	0	1	0	1	1	1	0
1	1	1	1	1	1	0	1	1	1

Na equação (4.4), pode-se observar que o caminho crítico do compressor 4:2 é dado pelo atraso de 4 portas lógicas XOR. Porém, foi realizado em (OKLOBDZIJA, 1996) um aperfeiçoamento dessa estrutura. Foram utilizados circuitos multiplexadores (MUX), reduzindo assim, o caminho crítico para 3 portas lógicas XOR. Além disso, uma outra vantagem do uso de multiplexadores, é que eles podem ser otimizados em nível de transistores. A nova estrutura para o compressor 4:2 pode ser observada na Figura 4.4.

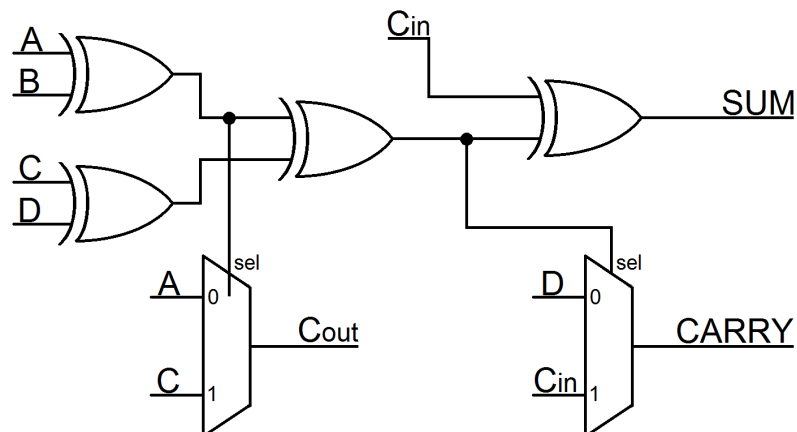


Figura 4.4 – Somador compressor 4:2 utilizando multiplexadores (OKLOBDZIJA; VILLEGGER; LIU, 1996).

A Figura 4.5 apresenta um exemplo de utilização de compressores 4:2 em uma soma de 4 números de 8 bits. Deve-se observar que são necessários circuitos somadores para a recombinação dos valores parciais das somas anteriores, visto que há a propagação de *carry* de um módulo de soma do compressor para o próximo bloco. Este exemplo foi utilizado, devido ao fato das arquiteturas testadas neste trabalho utilizarem amostras de 9 bits.

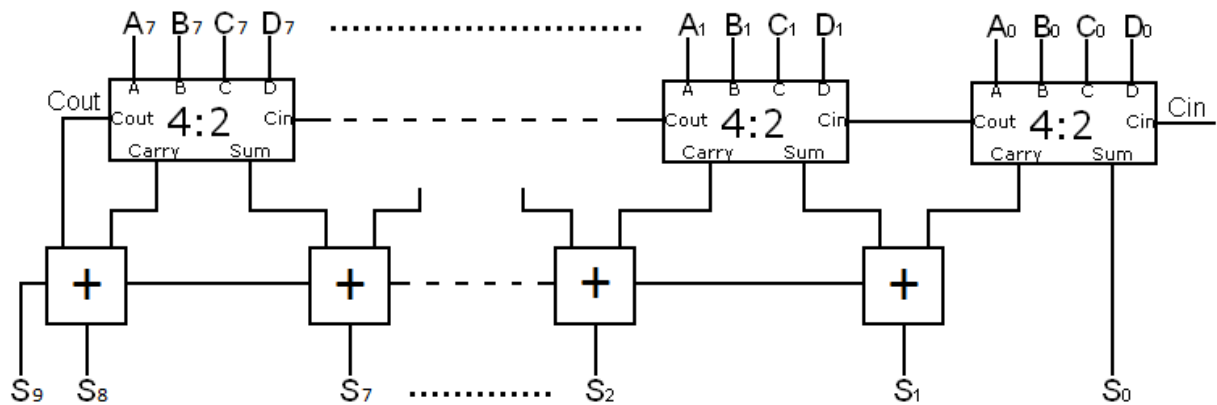


Figura 4.5 – Circuito somador de 8 bits de entrada utilizando compressor 4:2.

4.2.1 Somador Compressor 4:2 Modificado

A estrutura interna do somador compressor 4:2 foi modificada, no âmbito desse trabalho, para funcionar como um subtrator compressor 4:2, realizando a operação $(A+B-C-D)$, para ser utilizado nas arquiteturas de SATD. Nesta estrutura as entradas C e D foram invertidas e à entrada Cin foi atribuído o valor igual a 1. Ao bit menos significativo da saída Sum foi somado o valor igual a 1 para se obter a operação correta de subtração, como mostrado na Figura 4.6.

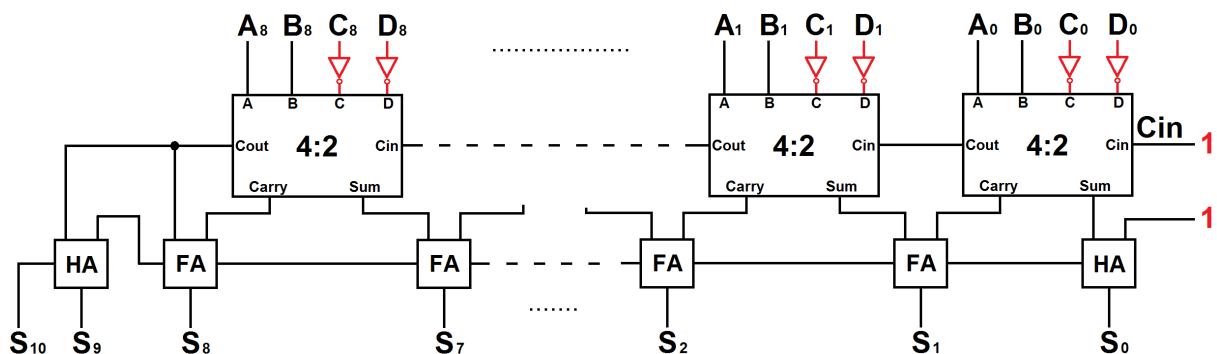


Figura 4.6 – Subtrator compressor 4:2 modificado.

4.2.2 Somador/Subtrator Compressor 4:2 Proposto

Devido ao fato dos circuitos internos do somador compressor 4:2 e do subtrator compressor 4:2 serem muito parecidos, foi realizado um estudo de forma a otimizar estes dois circuitos em apenas um, que realize ambas as operações de soma e subtração. Nesta nova arquitetura proposta, a estrutura do somador compressor 4:2 foi modificada para trabalhar como um somador e subtrator, realizando as operações $(A+B+C+D)$ e $(A+B-C-D)$ ao mesmo tempo. A necessidade desta nova proposta, deve-se ao fato de que a Transformada de Hadamard realiza inúmeras vezes em seus cálculos esses tipos de operações. Nessa nova estrutura são utilizados dois multiplexadores (MUX) e uma porta XOR adicionais em relação ao compressor convencional. As entradas C_{in} , C e D foram invertidas para serem utilizadas como entradas dos novos multiplexadores e da nova porta XOR, como mostra a Figura 4.7.

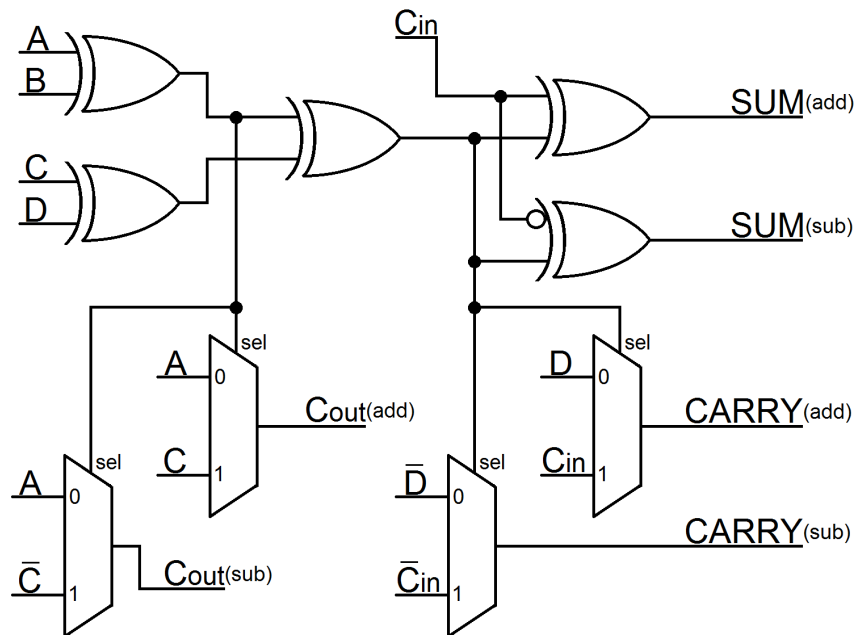


Figura 4.7 – Circuito somador/subtrator compressor 4:2 proposto.

Esta estrutura do somador/subtrator proposta apresenta seis saídas, onde $SUM_{(add)}$, $Carry_{(add)}$ e $Cout_{(add)}$ são resultados da adição $(A+B+C+D)$, enquanto $SUM_{(sub)}$, $Carry_{(sub)}$ e $Cout_{(sub)}$ são resultados da operação $(A+B-C-D)$, como pode ser observado na Figura 4.7. A Figura 4.8 apresenta um exemplo de uma soma e subtração de quatro valores de 9 bits.

Como mostrado anteriormente (ver seção 4.2), no somador compressor 4:2 convencional, é necessária uma linha de recombinação de meio somadores (HA - *Half Adder*) e somadores completos (FA - *Full-Adders*) para recombinar as somas parciais de valores prévios, pois há a

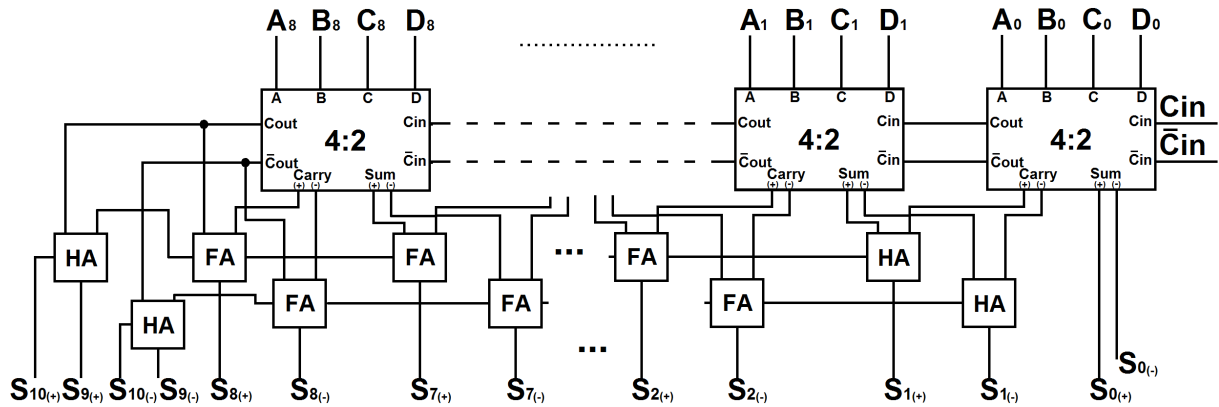


Figura 4.8 – Estrutura do subtrator/somador proposto para uma soma de quatro valores de 9 bits.

propagação do *carry* de um bloco para o próximo bloco. Já no caso do somador/subtrator compressor proposto, são necessárias duas linhas de recombinação, sendo uma para a realização da adição e outra para a operação da subtração, como mostra a Figura 4.7.

4.3 Compressor 8:2

A partir dos circuitos internos dos compressores 3:2 e 4:2, pode-se estender estas estruturas para a realização de um maior número de somas simultâneas. O compressor 8:2 pode ser construído hierarquicamente usando três compressores 4:2, como mostra a Figura 4.9-(a) ou utilizando quatro somadores compressores 3:2 e um compressor 4:2, como mostrado na Figura 4.9-(b).

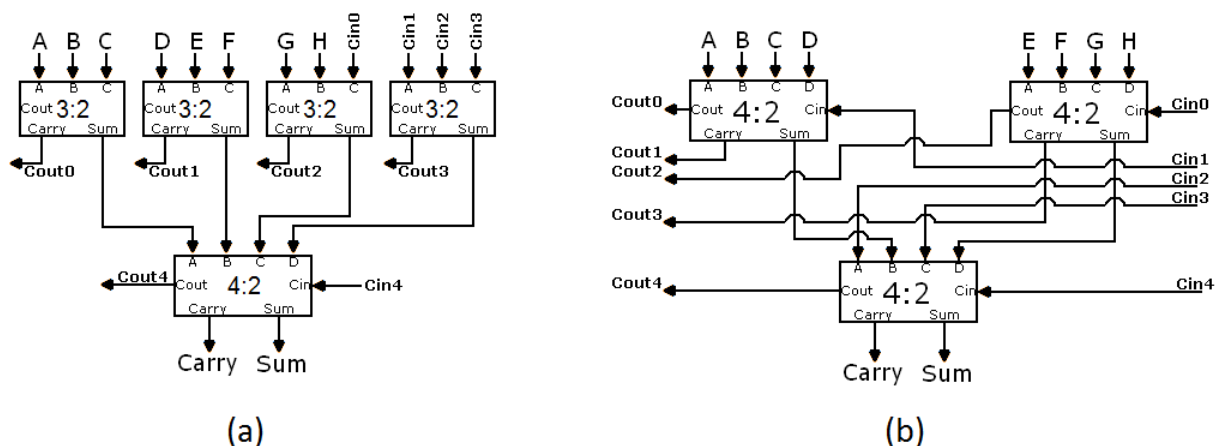


Figura 4.9 – Estrutura do somador compressor 8:2 usando: (a) a combinação dos compressores 3:2 e 4:2 (OKLOBDZIJA; VILLEGGER; LIU, 1996); (b) compressores 4:2 (OKLOBDZIJA; VILLEGGER; LIU, 1996).

Tabela 4.2 – Tabela Verdade do Somador Compressor 8:2

ENTRADA (j)	SAÍDA (j+1)	SUM (j)
0	0	0
1	0	1
2	1	0
3	1	1
4	2	0
5	2	1
6	3	0
7	3	1
8	4	0
9	4	1
10	5	0
11	5	1
12	6	0
13	6	1

O somador compressor 8:2 é composto por treze entradas, sendo oito delas primárias (A, B, C, D, E, F, G e H) e cinco entradas de *carry* de entrada (C_{in0} , C_{in1} , C_{in2} , C_{in3} e C_{in4}). A estrutura do compressor também apresenta sete saídas, sendo duas primárias (*Sum* e *Carry*) e cinco para C_{out} (C_{out0} , C_{out1} , C_{out2} , C_{out3} e C_{out4}). Assim como no compressor 4:2, o compressor 8:2 tem suas entradas e a saída *Sum* com peso (j) e as saídas restantes com peso (j+1). A Tabela 4.2 apresenta a tabela verdade para o compressor 8:2, considerando os pesos j e j+1 das saídas. O resultado final da soma para o somador compressor 8:2 é dado pela equação 4.8.

$$S = Sum + 2(C_{out0} + C_{out1} + C_{out2} + C_{out3} + C_{out4} + Carry) \quad (4.8)$$

Por exemplo, se todas as entradas estiverem em nível lógico 1, inclusive todas as entradas de C_{in} , a soma deverá ser 13, ou seja, $Soma = 1 + 2(1+1+1+1+1+1) = 13$. Observa-se na tabela 4.2 que para a entrada (j) igual a 13, a saída com peso (j+1) é igual a 6 e a saída *Sum* com peso j é igual a 1. A interpretação deste resultado é estabelecida no fato de que o valor 6 multiplicado por 2 (peso j+1 =2) é somado ao valor de *Sum* de valor igual a 1 (peso j = 1). O mesmo raciocínio é utilizado para os demais valores da tabela 4.2.

A Figura 4.10 mostra um exemplo de soma de 8 números de 8 bits a partir da utilização de blocos somadores compressores 8:2. Da mesma forma, como mostrado anteriormente para os compressores 3:2 e 4:2, são necessários circuitos somadores para recombina-
 ção das somas.

parciais dos valores anteriores (linha de recombinação). Neste trabalho são utilizados diferentes topologias de circuitos somadores para a realização das somas dos resultados parciais dos módulos compressores.

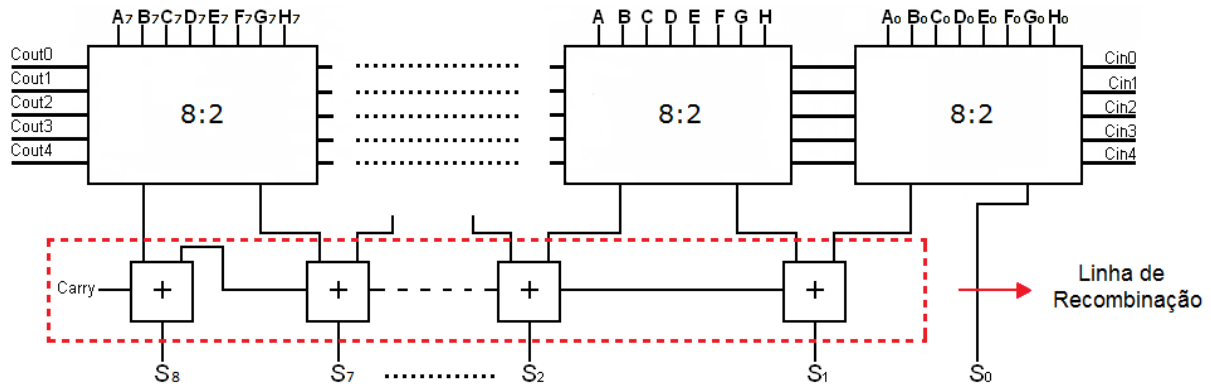


Figura 4.10 – Circuito somador de 8 bits de entrada utilizando compressor 8:2.

4.3.1 Aplicação de Somadores Eficientes na Linha de Recombinação

Como mostrado na Figura 4.10, são necessários circuitos somadores para a recombinação das somas parciais. A decisão de avaliar diferentes somadores na linha de recombinação é dado pelo fato de que esta linha representa o caminho crítico do circuito, necessitando assim de somadores eficientes para a sua implementação.

Nas duas estruturas do circuito somador compressor 8:2 (ver Figura 4.10) foram explorados três diferentes tipos de somadores na linha de recombinação: somador RCA (*Ripple Carry Adder*) composto por *full-adder*, somador RCA composto por *full-adder* baseado em MUX e somador Kogge-Stone.

- Somador RCA com *Full-adder*: O circuito somador *full-adder* é o somador mais comum encontrado na literatura (PARHAMI, 1999) e pode ser visto na Figura 4.11-(a). Estes somadores (FA) podem ser interligados em cascata para fazer a soma de N-bits. O circuito RCA é um circuito lógico composto por somadores, onde o *carry-out* de um bloco somador é conectado ao *carry-in* do próximo bloco. É chamado assim pela característica de que cada bit de *carry* é propagado para o próximo estágio. A Figura 4.11-(b) mostra um exemplo de ligação para soma de dois valores de 4 bits.

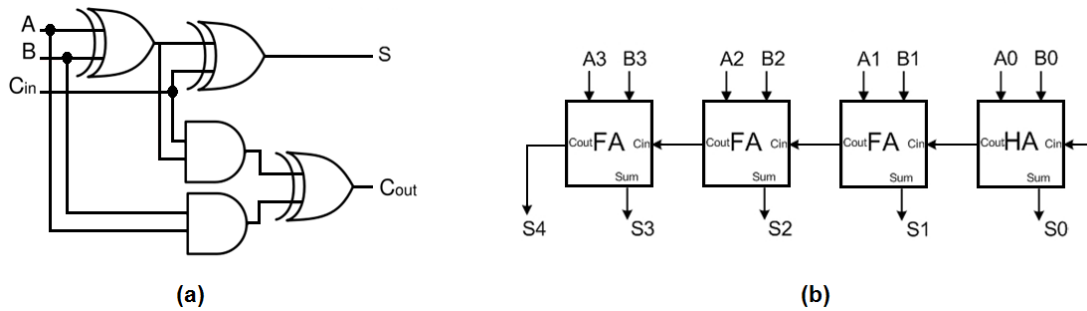


Figura 4.11 – (a) Somador *Full Adder* - FA; (b) Circuito RCA com FA.

Como observado na Figuar 4.11-(a), o somador *full-adder* apresenta três portas lógicas no caminho crítico, sendo assim um circuito RCA de N-bits apresenta N x 3 portas lógicas no caminho crítico.

- Somador RCA com *full-adder* baseado em MUX: O circuito somador *full-adder* baseado em MUX é composto por duas portas lógicas XOR e um multiplexador, como mostra a Figura 4.12. Este circuito somador é simplificado em relação ao *full-adder* comum, com o objetivo de reduzir o caminho crítico de propagação do *carry*. O circuito lógico RCA com somadores *full-adder* baseado em MUX tem a mesma configuração do circuito RCA com *full-adder* apresentado anteriormente na Figura 4.11-(b), mudando apenas os blocos *full-adder* por *full-adders* baseados em MUX.

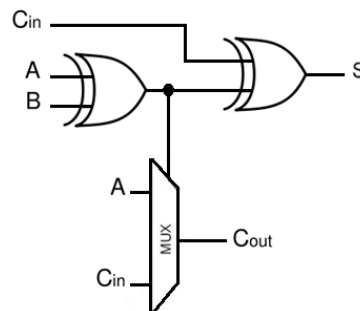


Figura 4.12 – Somador *full-adder* baseado em MUX.

Pode ser visto na Figura 4.12, que o caminho crítico para o *carry-out* é dado apenas pelo multiplexador, tornando-se interessante a sua utilização comparado ao *full-adder* comum, para o aumento de desempenho.

- Somador Kogge-Stone: O somador Kogge-Stone é conhecido como um somador de prefixo paralelo (PPA - *Parallel Prefix Adder*), que também é conhecido como somador de atraso logarítmico, onde a soma pode ser funcionalmente dividida em três etapas: pré-processamento, computação de prefixo e pós-processamento. No pré-processamento os vetores de entrada são utilizados para calcular os sinais necessários às etapas posteriores.

Para cada posição de entrada do somador são calculados os valores de geração (G) e propagação (P) de *carry*. As equações para esses respectivos cálculos são definidas e acordo com as equações (4.9) e (4.10).

$$G_i = A_i \cdot B_i \quad (4.9)$$

$$P_i = A_i \oplus B_i \quad (4.10)$$

Como as equações (4.9) e (4.10) podem ser efetuadas em paralelo, para todas as posições de entrada, não acrescenta grande atraso ao processamento das operações, e o custo em área é proporcional ao número de entradas. No cálculo do prefixo todos os sinais obtidos no pré-processamento são processados por uma estrutura de prefixo, que irá calcular todos os sinais necessários para efetuar a soma na última etapa. A maneira de calcular estes sinais, que constituem o fator c é mostrada na Figura 4.13.

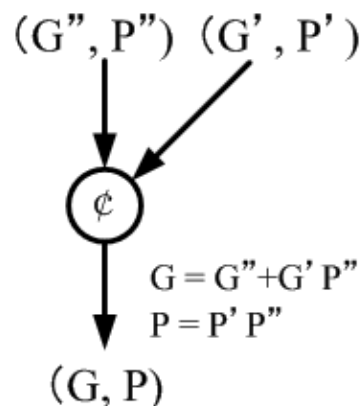


Figura 4.13 – Operador c para o cálculo do prefixo.

Os cálculos mostrados na Figura 4.13 são sempre realizados em árvores aproveitando os sinais que já foram calculados. Por fim, no pós-processamento é realizado o cálculo da soma utilizando os resultados das etapas anteriores.

Para a composição de somadores de prefixo paralelo (atraso logarítmico) são propostas na literatura várias estruturas de cálculo de prefixo (*carries*), onde as duas principais são as de Brent-Kung e Kogge-Stone. Entretanto, o somador Kogge-Stone apresenta uma estrutura mais regular e com um menor caminho crítico. O somador Kogge-Stone foi proposto por (KOGGE; STONE, 1973) em meados dos anos 70. A partir da pesquisa realizada na literatura, observa-se que os somadores Kogge-Stone são muito utilizados para os casos que necessitam de um alto grau de desempenho. Entretanto, para a obtenção

desse desempenho, são necessários inúmeros cálculos para alcançar os sinais de *carry* desejados, levando assim em um aumento de área. A Figura 4.14 apresenta o somador proposto por Kogge-Stone.

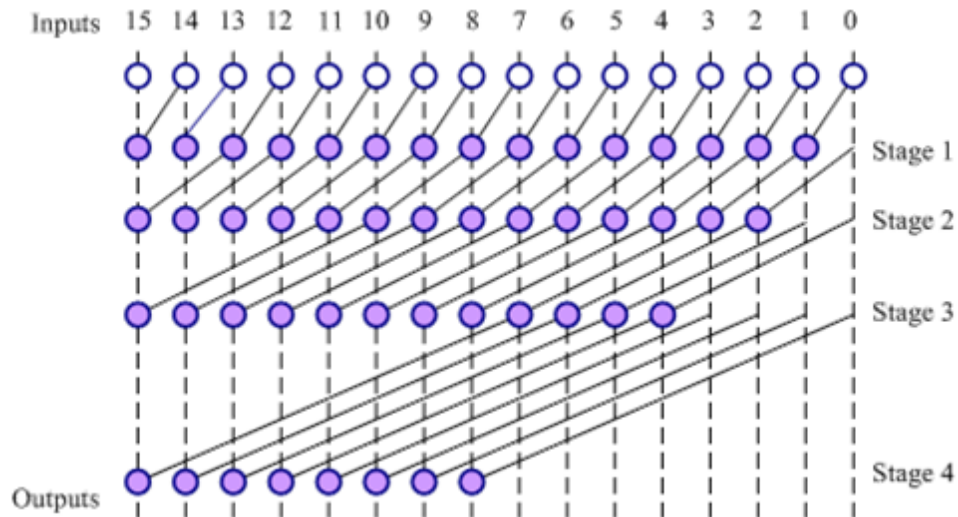


Figura 4.14 – Somador Kogge-Stone (KOGGE; STONE, 1973).

4.4 Resumo

Foi apresentado neste capítulo um estudo sobre circuitos somadores compressores que serão aplicados nas arquiteturas de SATD e SAD. Esse estudo mostrou a possibilidade da soma simultânea de vários operandos a partir do uso de circuitos somadores/subtratores compressores. Além do somador compressor tradicional, foram mostrados dois outros tipos de somadores compressores, sendo um modificado para a realização de subtração e um outro proposto no âmbito deste trabalho para a realização de somas e subtrações. O próximo capítulo dedica-se às arquiteturas desenvolvidas neste trabalho com o uso dos compressores, assim como uma relação de trabalhos relacionados encontrados na literatura.

5 ARQUITETURAS PROPOSTAS PARA AS MÉTRICAS SAD E SATD COM O USO DE SOMADORES COMPRESSORES

Neste capítulo são apresentadas as arquiteturas de *hardware* dedicadas desenvolvidas para a métricas de similaridade SAD e SATD. Também são apresentados trabalhos similares encontrados na literatura.

O foco deste trabalho é a exploração de arquiteturas de *hardware* da métrica SAD e da métrica SATD baseada nas HTs encontradas no *software* de referência do padrão HEVC. As técnicas são utilizadas no sentido do aumento do desempenho e redução da dissipação de potência das arquiteturas da SAD e SATD, que são partes integrantes dos módulos de Estimção de Movimento e Predição Intra do padrão HEVC. As arquiteturas foram descritas em linguagem de descrição de *hardware* VHDL (*VHSIC Hardware Description Language*), onde VHSIC quer dizer *Very High Speed Integrated Circuit*. Como principal técnica de redução de potência aplicada às arquiteturas desenvolvidas, são utilizados circuitos somadores compressores que realizam a soma simultânea de mais de dois operandos. São explorados vários tipos de somadores compressores presentes na literatura, que são aplicados às arquiteturas de SAD e SATD. Também são utilizados os somadores/subtratores compressores propostos neste trabalho.

5.1 Exploração Arquitetural na Métrica SATD

No software de referência (HM) (FRAUNHOFER, 2015) do padrão HEVC, os blocos são avaliados pela SATD, que é baseada em três diferentes tamanhos da HT. A HT 8x8 é aplicada para blocos de amostras 8x8 ou maiores. Os menores blocos são avaliados com as HTs 4x4 e 2x2. O cálculo dessas três transformadas podem ser vistos nas equações (5.1), (5.2) e (5.3), respectivamente.

$$HT_{2 \times 2} = H \cdot W \cdot H^T \quad (5.1)$$

$$HT_{4 \times 4} = (H \cdot W \cdot H^T) / 2 \quad (5.2)$$

$$HT_{8 \times 8} = (H \cdot W \cdot H^T) / 4 \quad (5.3)$$

Nas equações apresentadas acima, o termo W é o bloco residual e H é a matriz da HT. As matrizes das HTs 8x8, 4x4 e 2x2 podem ser vistas nas equações (5.4), (5.5) e (5.6),

respectivamente.

$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (5.4)$$

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} / 2 \quad (5.5)$$

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} / 4 \quad (5.6)$$

Como pode-se observar, os elementos das matrizes só apresentam valores 1 e -1. Portanto, só se faz necessário o uso de operações de soma e subtração para a realização dos cálculos das transformadas. A divisão por dois e por quatro são realizadas por meio de 1 e 2 deslocamentos para a direita, respectivamente. Desta forma, torna-se necessário o uso de técnicas eficientes que possam realizar estas operações de soma e subtração com a menor dissipação de potência.

As próximas subseções apresentam duas contribuições deste trabalho relacionadas à métrica SATD. A primeira subseção apresenta uma exploração arquitetural, com diferentes níveis de paralelismo, para a métrica SATD, baseada na HT 8x8. A segunda subseção apresenta uma arquitetura para SATD suportando múltiplos tamanhos da HT. Nas duas subseções, utilizam-se somadores compressores para redução da dissipação de potência.

5.1.1 Arquiteturas da SATD baseadas na HT 8x8

Nesta etapa, explora-se a possibilidade da soma de vários operandos simultaneamente utilizando os circuitos compressores, com o menor caminho crítico possível, como forma de acelerar os cálculos da SATD. As arquiteturas são desenvolvidas explorando três diferentes

níveis de paralelismo para a HT: i) arquitetura sequencial; ii) arquitetura semi-paralela; iii) arquitetura totalmente paralela.

Nas três arquiteturas desenvolvidas, a HT 8x8 2-D (duas dimensões) foi dividida em dois estágios de 1-D, chamados de Hadamard horizontal e Hadamard vertical. As três arquiteturas seguem o mesmo modelo mostrado na Figura 5.1.

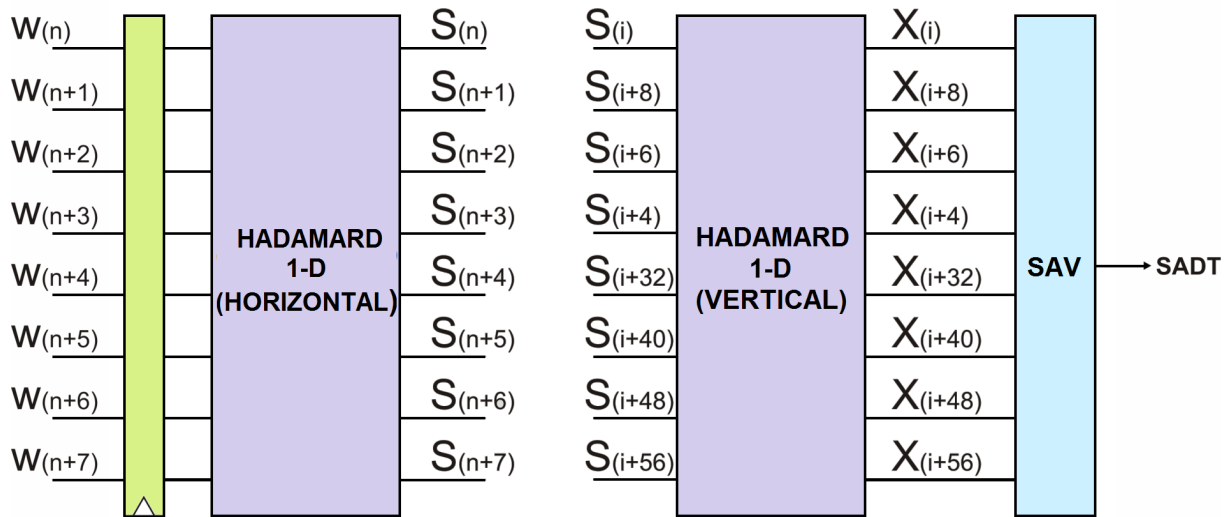


Figura 5.1 – Modelo da arquitetura de *hardware* da SATD

As transformadas horizontal e vertical recebem 8 amostras de entrada e entregam à saída em um ciclo de relógio. Os registradores de entrada (em verde na Figura 5.1) armazenam as amostras de entrada para mantê-las estáveis. As amostras de saída da HT vertical são acumuladas pelo módulo da soma dos valores absolutos (SAV – *Sum of Absolute Values*). O resultado final da SATD acumulado é armazenado em um registrador de saída. A Figura 5.2-a mostra o circuito interno da Hadamard 8x8 1-D. A Figura 5.2-b apresenta a arquitetura implementada com somador e subtrator compressor modificado (ver subseção 4.2.1), enquanto a Figura 5.2-c apresenta a arquitetura implementada com somador/subtrator compressor proposto (ver subseção 4.2.2).

Para o módulo de SAV também foram aplicados os somadores compressores. A Figura 5.3-a apresenta o circuito interno do módulo SAV original, enquanto que a Figura 5.3 apresenta este módulo com o somador compressor 8:2.

O circuito interno dos blocos das transformadas Hadamard horizontal e vertical são idênticos mudando somente os valores das entradas e saídas de cada um. O módulo SAV é composto por um conjunto de 8 registradores que armazenam as amostras entregues pela Hadamard vertical, 8 operadores absolutos e uma árvore de somadores que acumula os valores. A Tabela 5.1

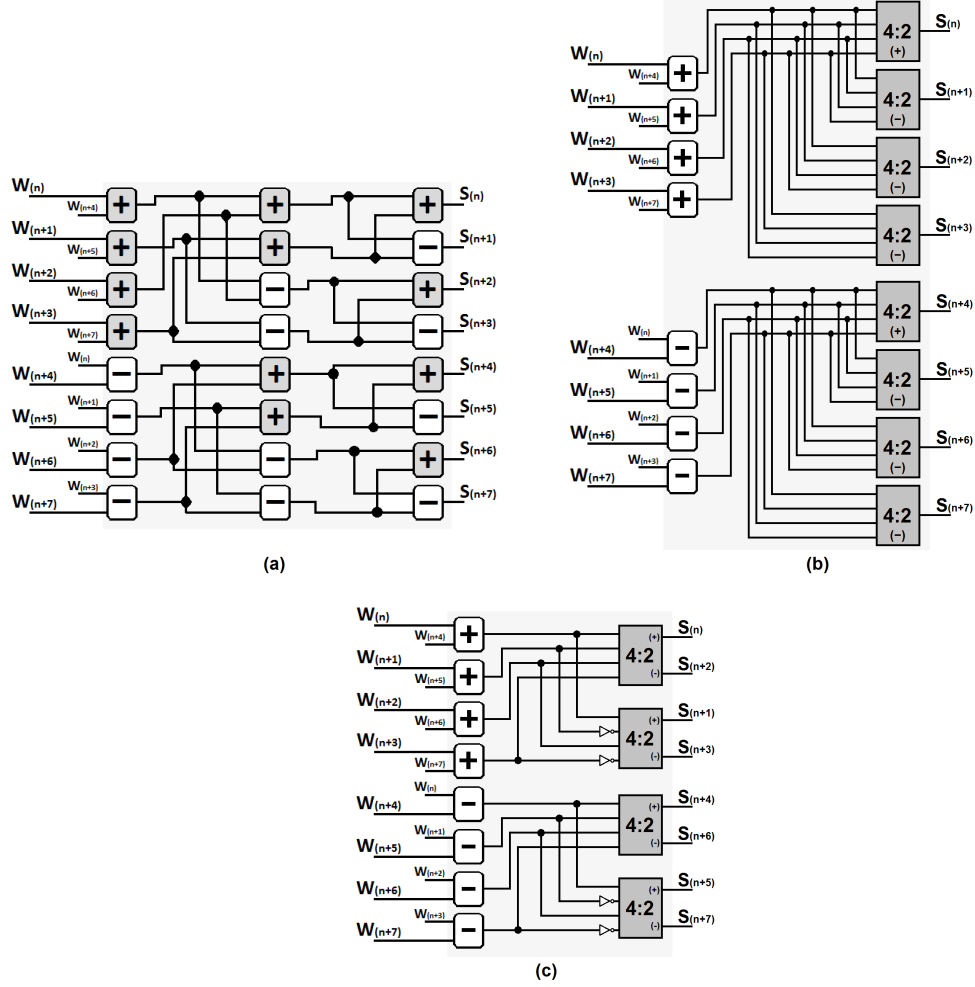


Figura 5.2 – (a) Circuito interno da Hadamard 1-D; (b) Circuito interno da Hadamard 1-D com somador e subtrator compressor modificado; (c) Circuito interno da Hadamard 1-D com somador/subtrator compressor proposto.

apresenta a operação matemática da transformada horizontal, enquanto a Tabela 5.2 mostra a operação para a transformada vertical.

Na Tabela 5.1, $W_{(n)}$ até $W_{(n+7)}$ são as amostras de entrada e $S_{(n)}$ até $S_{(n+7)}$ são as saídas parciais da transformada horizontal, onde n é a variável que define a posição da matriz no algoritmo, a qual começa com 0 e é incrementada de 8 em 8. Na Tabela 5.2, somente a posição da matriz é que varia, onde n é trocado por i , o qual é inicializado com 0 e incrementado de 1 em 1.

Na Tabela 5.2, $S_{(i)}$ até $S_{(i+56)}$ são as amostras de entrada e $X_{(i)}$ até $X_{(i+56)}$ são as saídas finais da transformada vertical. Nas Tabelas 5.1 e 5.2, $A_{(n)}$ até $A_{(n+7)}$, $B_{(n)}$ até $B_{(n+7)}$, $C_{(i)}$ até $C_{(i+56)}$ e $D_{(i)}$ até $D_{(i+56)}$ são apenas variáveis internas do algoritmo das transformadas horizontal e vertical. No final dos cálculos da segunda Hadamard 1-D, os resultados são entregues ao módulo SAV. As três arquiteturas são detalhadas nas próximas subseções.

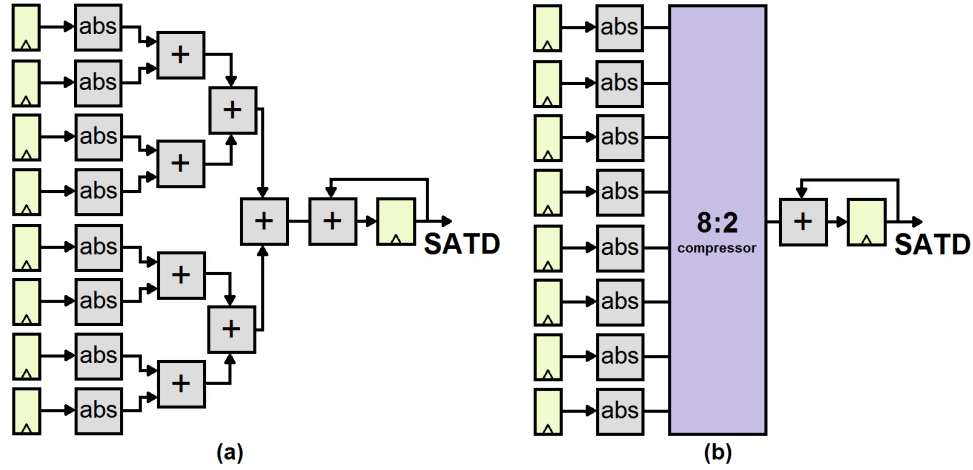


Figura 5.3 – (a) Circuito interno do módulo SAV original; (b) Circuito interno do módulo SAV com compressor 8:2.

Tabela 5.1 – Cálculo da Hadamard Horizontal

$A_{(n)} = W_{(n)} + W_{(n+4)}$	$B_{(n)} = A_{(n)} + A_{(n+2)}$	$S_{(n)} = B_{(n)} + B_{(n+1)}$
$A_{(n+1)} = W_{(n+1)} + W_{(n+5)}$	$B_{(n+1)} = A_{(n+1)} + A_{(n+3)}$	$S_{(n+1)} = B_{(n)} - B_{(n+1)}$
$A_{(n+2)} = W_{(n+2)} + W_{(n+6)}$	$B_{(n+2)} = A_{(n)} - A_{(n+2)}$	$S_{(n+2)} = B_{(n+2)} + B_{(n+3)}$
$A_{(n+3)} = W_{(n+3)} + W_{(n+7)}$	$B_{(n+3)} = A_{(n+1)} - A_{(n+3)}$	$S_{(n+3)} = B_{(n+2)} - B_{(n+3)}$
$A_{(n+4)} = W_{(n)} - W_{(n+4)}$	$B_{(n+4)} = A_{(n+4)} + A_{(n+6)}$	$S_{(n+4)} = B_{(n+4)} + B_{(n+5)}$
$A_{(n+5)} = W_{(n+1)} - W_{(n+5)}$	$B_{(n+5)} = A_{(n+5)} + A_{(n+7)}$	$S_{(n+5)} = B_{(n+4)} - B_{(n+5)}$
$A_{(n+6)} = W_{(n+2)} - W_{(n+6)}$	$B_{(n+6)} = A_{(n+4)} - A_{(n+6)}$	$S_{(n+6)} = B_{(n+6)} + B_{(n+7)}$
$A_{(n+7)} = W_{(n+3)} - W_{(n+7)}$	$B_{(n+7)} = A_{(n+5)} - A_{(n+7)}$	$S_{(n+7)} = B_{(n+6)} - B_{(n+7)}$

5.1.1.1 Arquitetura Sequencial

Na arquitetura sequencial a Hadamard horizontal recebe 8 amostras de entrada, $W_{(n)}$ até $W_{(n+7)}$. A cada ciclo de relógio uma linha de amostras do bloco 8x8 (64 amostras) são entregues como entrada para a primeira transformada 1-D (horizontal). O bloco inteiro 8x8 é processado em 8 ciclos de relógio pela Hadamard horizontal. A Figura 5.4 apresenta a configuração para esta arquitetura sequencial.

Os dois estágios 1-D da Hadamard são conectados por um conjunto de oito registradores sequencial/paralelo e oito multiplexadores. Eles são responsáveis por lerem as oito saídas da Hadamard horizontal (uma linha do bloco de amostra 8x8) e entregarem estes resultados coluna por coluna para a Hadamard vertical. Isto é feito porque a Hadamard vertical é aplicada na matriz transposta (ver equação (5.3)). Os registradores sequenciais (em cinza na Figura 5.4) armazenam as oito saídas da Hadamard horizontal a cada ciclo de relógio. Após 8 ciclos de relógio, todas as 64 amostras nos registradores sequenciais são então armazenadas nos registradores paralelos (em verde na Figura 5.4) e entregues para os multiplexadores. Estes multiplexadores são os responsáveis por selecionar os resultados armazenados nos registradores paralelos e entregar

Tabela 5.2 – Cálculo da Hadamard Vertical

$C_{(i)} = S_{(i)} + S_{(i+32)}$	$D_{(i)} = C_{(i)} + C_{(i+16)}$	$X_{(i)} = D_{(i)} + D_{(i+8)}$
$C_{(i+8)} = S_{(i+8)} + S_{(i+40)}$	$D_{(i+8)} = C_{(i+8)} + C_{(i+24)}$	$X_{(i+8)} = D_{(i)} - D_{(i+8)}$
$C_{(i+16)} = S_{(i+16)} + S_{(i+48)}$	$D_{(i+16)} = C_{(i)} - C_{(i+16)}$	$X_{(i+16)} = D_{(i+16)} + D_{(i+24)}$
$C_{(i+24)} = S_{(i+24)} + S_{(i+56)}$	$D_{(i+24)} = C_{(i+8)} - C_{(i+24)}$	$X_{(i+24)} = D_{(i+16)} - D_{(i+24)}$
$C_{(i+32)} = S_{(i)} - S_{(i+32)}$	$D_{(i+32)} = C_{(i+32)} + C_{(i+48)}$	$X_{(i+32)} = D_{(i+32)} + D_{(i+40)}$
$C_{(i+40)} = S_{(i+8)} - S_{(i+40)}$	$D_{(i+40)} = C_{(i+40)} + C_{(i+56)}$	$X_{(i+40)} = D_{(i+32)} - D_{(i+40)}$
$C_{(i+48)} = S_{(i+16)} - S_{(i+48)}$	$D_{(i+48)} = C_{(i+32)} - C_{(i+48)}$	$X_{(i+48)} = D_{(i+48)} + D_{(i+56)}$
$C_{(i+56)} = S_{(i+24)} - S_{(i+56)}$	$D_{(i+56)} = C_{(i+40)} - C_{(i+56)}$	$X_{(i+56)} = D_{(i+48)} - D_{(i+56)}$

coluna por coluna para a Hadamard vertical.

As saídas da Hadamard vertical são entregues para o módulo SAV, o qual acumula as 64 amostras e gera o valor final da SATD. O valor parcial da SATD é armazenado em um registrador a cada ciclo de relógio. Sendo assim, após 8 ciclos de relógio, o valor final da SATD é disponível no registrador final. A latência inicial desta arquitetura é de 19 ciclos de relógio, e após ter sido preenchido o *pipeline*, a arquitetura entrega um novo resultado SATD a cada 8 ciclos de relógio.

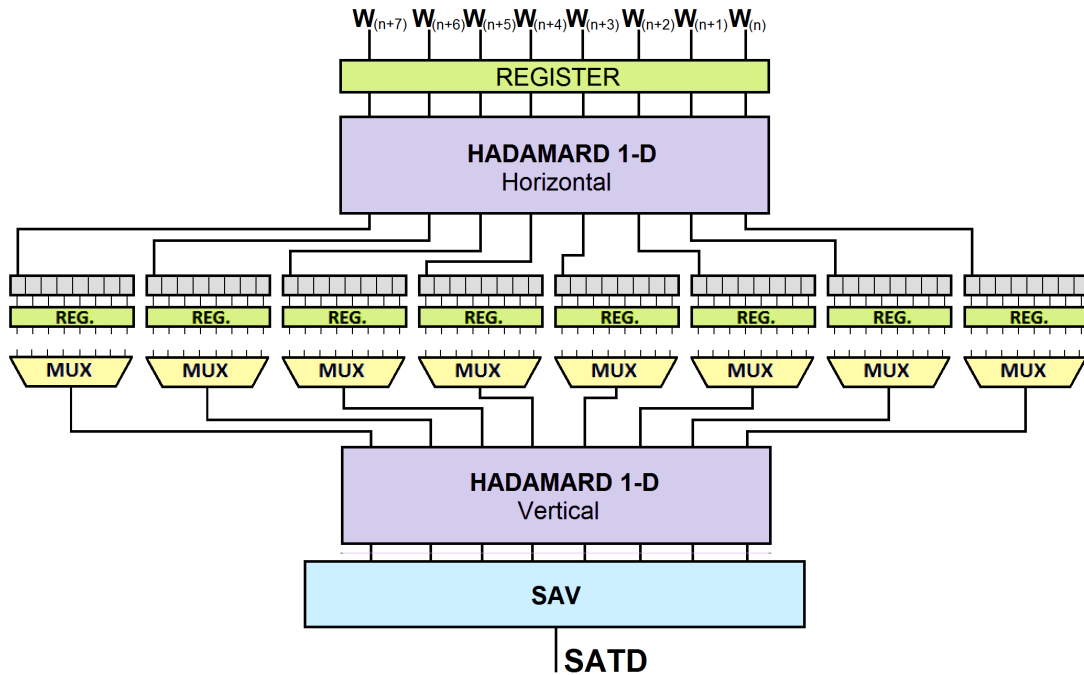


Figura 5.4 – Diagrama de blocos da arquitetura sequencial.

5.1.1.2 Arquitetura Semi-Paralela

A arquitetura semi-paralela é composta por 4 HTs em paralelo. Esta arquitetura processa 32 amostras de entrada ($W_{(n)}$ to $W_{(n+32)}$) ao mesmo tempo em um ciclo de relógio. Portanto, a cada ciclo de relógio, quatro linhas do bloco 8x8 de amostras são entregues para as entradas da primeira transformada 1-D, onde cada linha é entregue para cada Hadamard horizontal ao mesmo tempo. Todo bloco 8x8 é processado em 2 ciclos de relógio pela Hadamard horizontal. A Figura 5.5 mostra o diagrama de blocos desta arquitetura.

As transformadas horizontal e vertical são conectadas por um conjunto de 32 registradores sequencias e 32 registradores paralelos. Esses registradores são responsáveis por lerem as 32 saídas da Hadamard horizontal (4 linhas de um bloco de amostra 8x8) e por entregá-las para os 4 módulos da Hadamard vertical.

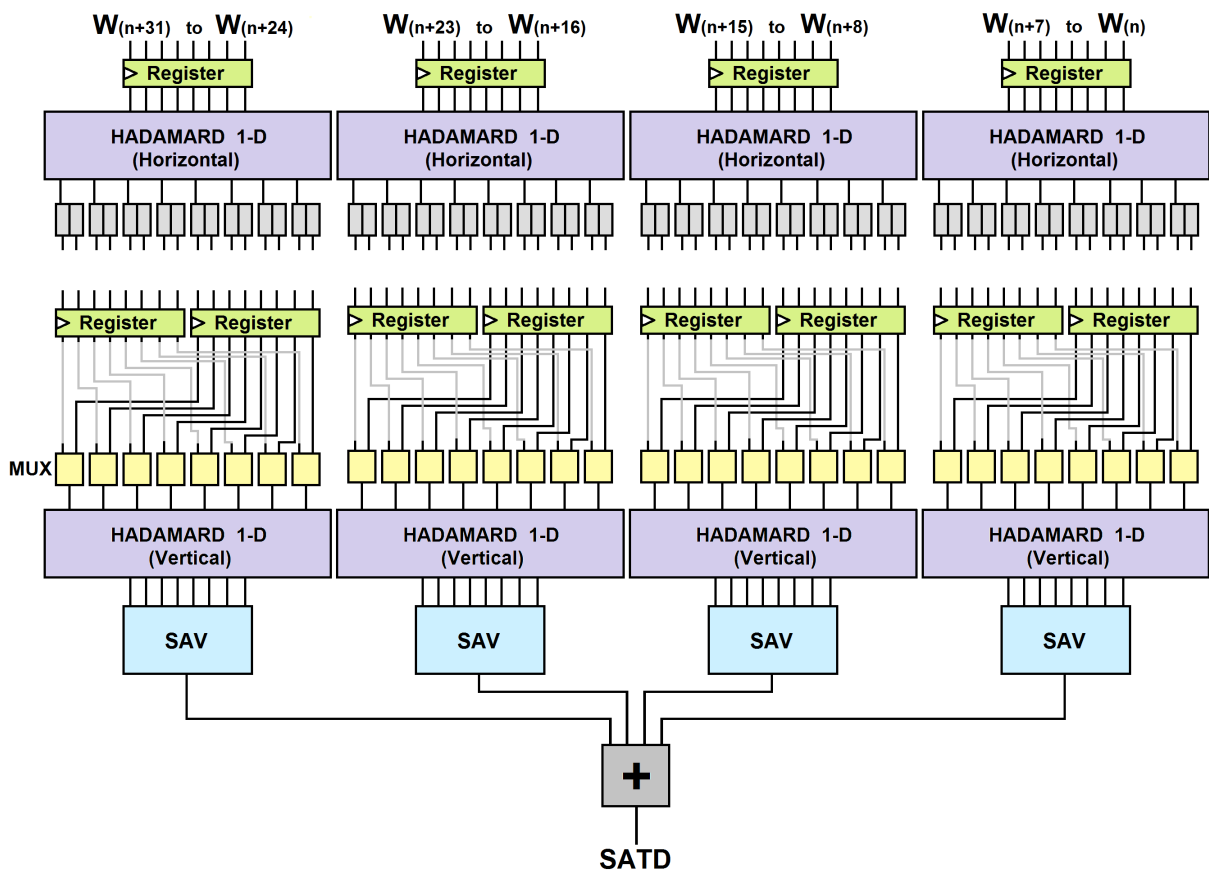


Figura 5.5 – Diagrama de blocos da arquitetura semi-paralela.

Nessa arquitetura, a cada ciclo de relógio, os registradores sequenciais (em cinza na Figura 5.5) armazenam as 32 amostras de saída da Hadamard horizontal. Após 2 ciclos de relógio, todas as 64 amostras nos registradores sequenciais são armazenadas nos registradores

paralelos (em verde na Figura 5.5) e então, entregues aos multiplexadores. Cada multiplexador entrega os resultados coluna por coluna para cada Hadamard vertical.

Quatro módulos de SAV são utilizados para receber as saídas das transformadas verticais. Os resultados são entregues para uma árvore de somadores, a qual gera o valor final acumulado da SATD. A latência inicial desta arquitetura é de 9 ciclos de relógio e, após o *pipeline* cheio, a arquitetura entrega um novo valor para a SATD a cada 2 ciclos de relógio.

5.1.1.3 Arquitetura Totalmente Paralela

Na arquitetura totalmente paralela, oito transformadas horizontal e vertical são utilizadas neste sistema como mostra a Figura 5.6. A transformada horizontal recebe as 64 amostras como entrada ($W_{(0)}$ até $W_{(63)}$), onde isto significa que todo o bloco de amostra 8x8 é processado em um ciclo de relógio pelo estágio da Hadamard horizontal.

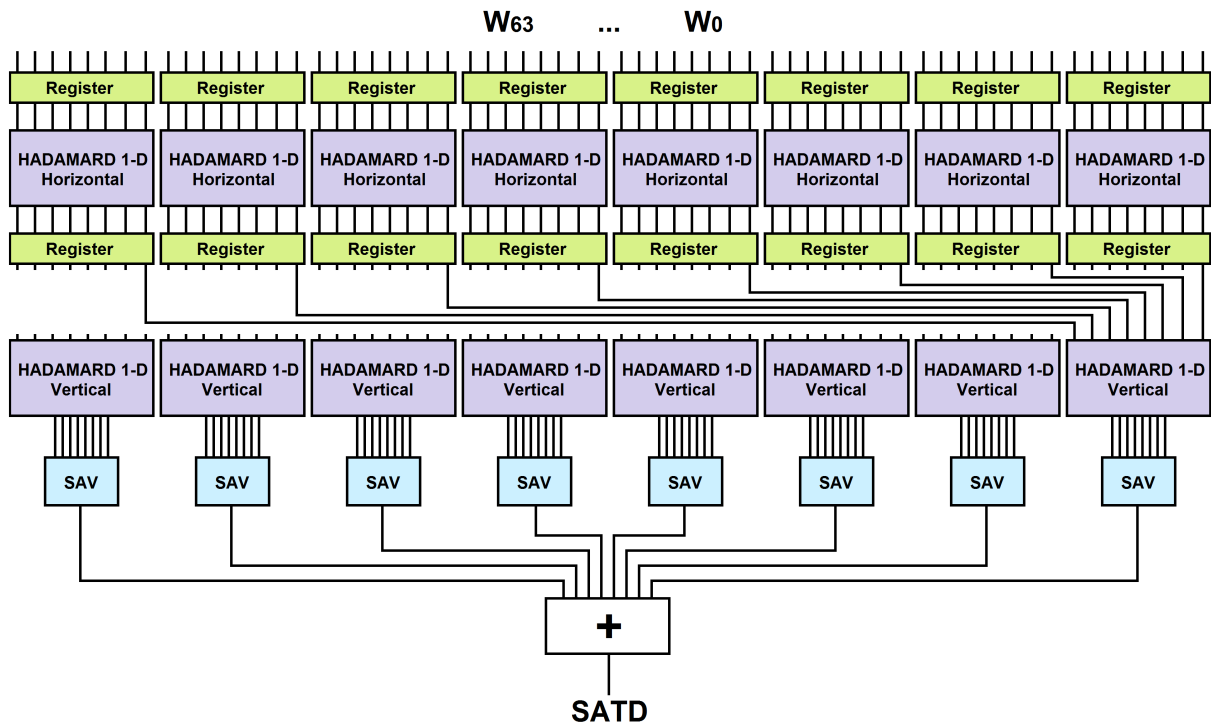


Figura 5.6 – Diagrama de blocos da arquitetura totalmente paralela.

Nesta arquitetura, não se faz necessária a utilização de registradores sequenciais, visto que as 64 amostras de saída da Hadamard horizontal são entregues no mesmo ciclo de relógio. A latência inicial desta arquitetura é de 4 ciclos de relógio e, após o *pipeline* cheio, um novo resultado da SATD é entregue a cada ciclo de relógio. Um conjunto de registradores é utilizado apenas para criar um estágio de *pipeline* (em verde na Figura 5.6). Para gerar o resultado final acumulado da SATD, é utilizado um conjunto de 8 módulos de SAV e uma árvore de somadores.

5.1.2 Arquitetura da SATD com Múltiplos Tamanhos de Transformada Hadamard

Nesta segunda etapa da exploração arquitetural da métrica SATD foi proposta uma arquitetura de hardware baseada em múltiplos tamanhos da HT (2x2, 4x4 e 8x8). Observa-se que a matriz da HT 8x8 é composta por quatro matrizes de tamanho 4x4 e esta, por sua vez, é composta por quatro matrizes 2x2. Com base nesta informação é possível um reaproveitamento de *hardware* das matrizes menores para os cálculos das matrizes maiores. Essa divisão das matrizes pode ser vista na Figura 5.7.

	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$
2x2 Matrices	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$
	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$
4x4 Matrices	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$
	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$
	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix}$
8x8 Matrix	$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix}$

Figura 5.7 – Ilustração da divisão das matrizes.

Com base no reaproveitamento das matrizes da HT foi possível desenvolver uma única arquitetura de *hardware* para as operações da SATD 8x8, 4x4 e 2x2. Essa arquitetura para tamanhos múltiplos da SATD é apresentada na Figura 5.8.

A SATD 2x2 é composta por quatro entradas (W_0 a W_3) e uma saída (SATD2x2). A operação dessa transformada é apresentada na caixa em vermelho da Figura 5.8. A SATD 4x4 é composta por dezesseis entradas, onde as quatro primeiras recebem as saídas (X_0 a X_3) da Hadamard 2x2 como pode ser visto na caixa em azul da Figura 5.8. Já a SATD 8x8 possui sessenta e quatro amostras de entrada. As dezesseis primeiras entradas recebem as saídas (S_0 a S_{15}) da transformada 4x4, e pode ser visto na caixa em verde da Figura 5.8. Observa-se que para calcular a transformada de tamanho 4x4 são utilizadas quatro vezes a transformada de tamanho 2x2, enquanto que para calcular a transformada de tamanho 8x8, a transformada de tamanho 4x4 é utilizada quatro vezes.

A arquitetura de *hardware* da SATD é composta por: um bloco de controle; três bancos de registradores para armazenar as amostras de entrada; o módulo da HT de múltiplos tamanhos; três bancos de registradores como *pipeline*; e o módulo de SAV. O bloco de controle

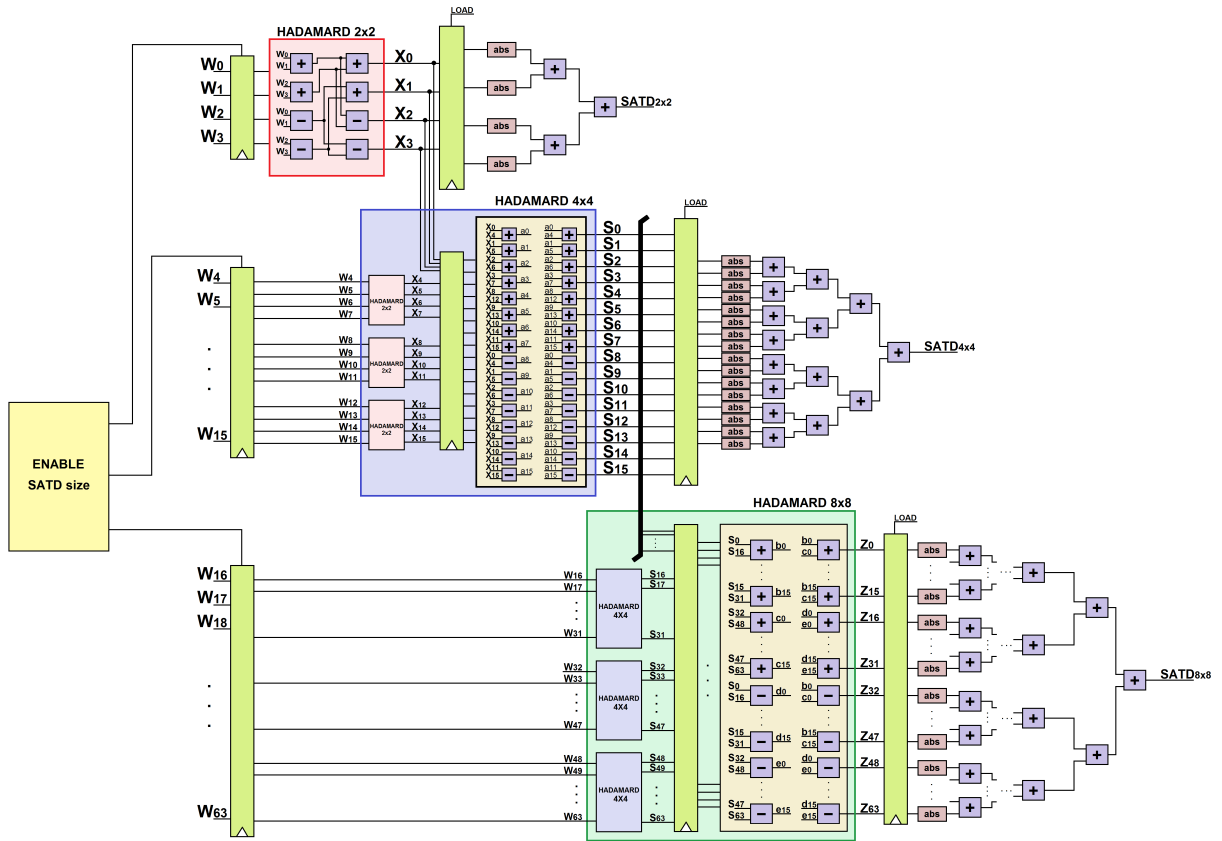


Figura 5.8 – SATD baseada em múltiplos tamanhos da HT.

(representado por Enable SATD size na Figura 5.8), tem como função selecionar qual registrador será carregado, assim como o número de amostras de entrada. Quando a saída do bloco de controle é ‘00’, a arquitetura de SATD recebe apenas as quatro primeiras amostras como entrada. Nessa situação, o primeiro banco de registradores (W_0 a W_3) e o primeiro estágio de *pipeline* (X_0 a X_3) são habilitados e, consequentemente, a arquitetura somente realiza a operação da SATD2x2. Por outro lado, se a saída do bloco de controle for igual a ‘01’, então a arquitetura de SATD recebe apenas as dezesseis primeiras amostras como entrada. O primeiro e o segundo bancos de registradores (W_0 a W_3 e W_4 a W_{15}), assim como o segundo estágio de *pipeline* (S_0 a S_{15}), são habilitados e a arquitetura somente realiza a operação de SATD4x4. Finalmente, se a saída do bloco de controle for igual a ‘10’, a arquitetura recebe todas as sessenta e quatro amostras de entrada. Neste caso, os três bancos de registradores (W_0 a W_{63}) e o terceiro estágio de *pipeline* (Z_0 a Z_{63}) são habilitados e, consequentemente, a arquitetura realiza apenas a operação de SATD8x8. Os estágios de *pipeline* são habilitados pelo sinal *LOAD*, que pode ser visto na Figura 5.8. As saídas das HTs de múltiplos tamanhos são acumuladas pelo módulo de SAV, o qual é composto por operandos absolutos e uma árvore de somadores (semelhante ao do primeiro estudo).

Na arquitetura de SATD para múltiplos tamanhos da HT foram introduzidos os somado-

res/subtratores compressores proposto como mostra a Figura 5.9. No módulos de SAV foram também utilizados os somadores compressores 4:2 convencionais (ver Figura 4.4) e somadores compressores 8:2 (ver Figura 4.9-b). Enquanto que a árvore de somadores do módulo de SAV 2x2 é substituída por um somador compressor 4:2, a árvore de somadores do módulo SAV 4x4 é substituída por dois somadores compressores 8:2. E por último, a árvore de somadores do módulo SAV 8x8 é substituída por nove somadores compressores 8:2.

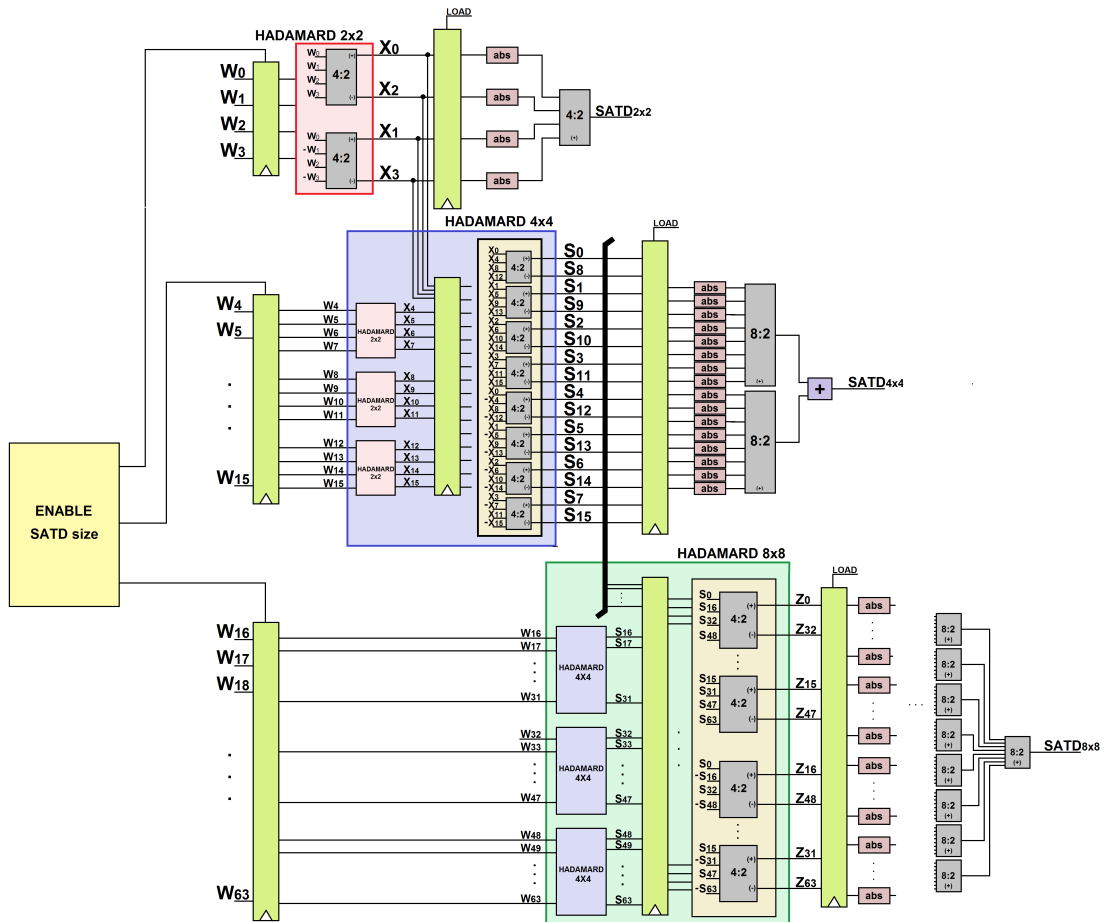


Figura 5.9 – SATD baseada em múltiplos tamanhos da HT usando compressores.

5.2 Exploração Arquitetural na Métrica SAD

O diagrama em blocos da arquitetura utilizada neste trabalho pode ser visto na Figura 5.10-(a). Esta arquitetura é composta por 8 subtratores, os quais realizam a operação de subtração entre o bloco original e o bloco de referência (bloco de resíduo), um conjunto de oito registradores para armazenar as amostras residuais, oito operadores absolutos e uma árvore de

somadores para acumular os valores parciais. Nessa arquitetura são utilizados somadores *macrofunction* provenientes da ferramenta de síntese. Os valores parciais são armazenados em um registrador a cada ciclo de relógio, tendo-se o valor final de SAD em 8 ciclos de relógio para um bloco 8x8. Essa arquitetura representa a primeira versão (SAD-v1) implementada neste trabalho.

Observando a arquitetura SAD-v1, nota-se que os valores intermediários da árvore de somadores não são utilizados para os cálculos. Sendo assim, essa árvore de somadores pode ser perfeitamente substituída por somadores compressores eficientes. A utilização de compressores 8:2 nessa árvore, torna-se interessante visto que este permite a soma simultânea de oito operandos ao mesmo tempo, apresentando um grande potencial para redução de potência na arquitetura de SAD.

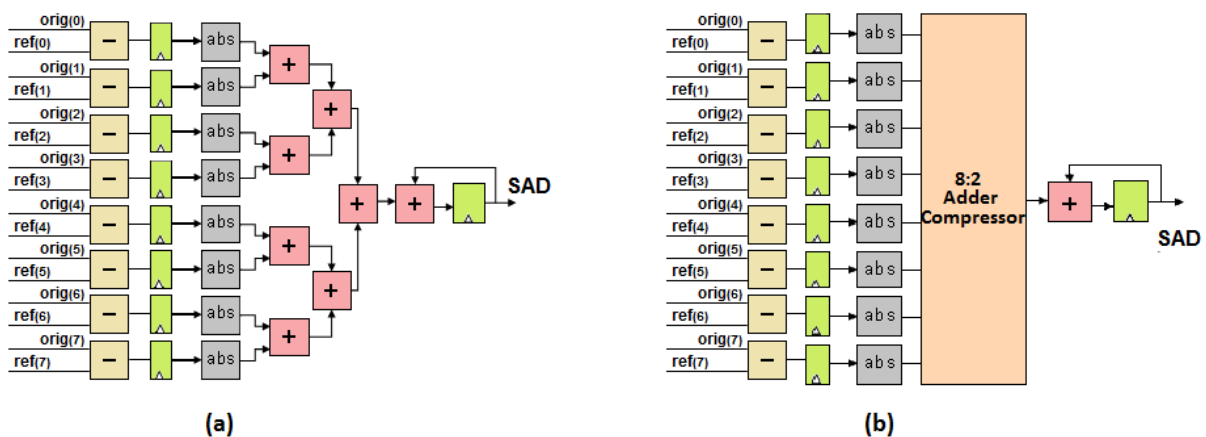


Figura 5.10 – Diagrama em blocos da arquitetura da SAD: (a) usando o somador da ferramenta; (b) usando compressor 8:2.

A Figura 5.10-(b) apresenta uma arquitetura genérica de SAD usando compressor 8:2. Nesta etapa do trabalho o foco principal é a exploração arquitetural para esta arquitetura de SAD utilizando diferentes estruturas do compressor 8:2. Também são explorados diferentes somadores na linha de recombinação (ver Figura 4.10) do compressor 8:2. Deste modo, as onze versões das arquiteturas propostas diferem entre elas em como o compressor 8:2 foi implementado e quais tipos de somadores são utilizados na linha de recombinação das somas parciais. A seguir são apresentadas todas as versões implementadas.

- SAD-v1 - arquitetura implementada com somadores da ferramenta de síntese;
- SAD-v2 - arquitetura implementada com compressor 8:2 (composto por três compressores 4:2 - ver Figura 4.9-(b)) e somadores *full-adder* na linha de recombinação;

- SAD-v3 - arquitetura implementada com compressor 8:2 (composto por três compressores 4:2 - ver Figura4.9-(b)) e somadores *full-adder* baseado em MUX na linha de recombinação;
- SAD-v4 - arquitetura implementada com compressor 8:2 (composto por quatro compressores 3:2 e um compressor 4:2 - ver Figura4.9-(a)) e somadores *full-adder* na linha de recombinação;
- SAD-v5 - arquitetura implementada com compressor 8:2 (composto por quatro compressores 3:2 e um compressor 4:2 - ver Figura4.9-(a)) e somadores *full-adder* baseado em MUX na linha de recombinação;
- SAD-v6 - arquitetura implementada somente com somadores *full-adder* como RCA;
- SAD-v7- arquitetura implementada somente com somadores *full-adder*, baseado em MUX, como RCA;
- SAD-v8- arquitetura implementada com compressor 8:2 (composto por três compressores 4:2 - ver Figura4.9-(b)) e somadores da ferramenta de síntese na linha de recombinação;
- SAD-v9- arquitetura implementada com compressor 8:2 (composto por quatro compressores 3:2 e um compressor 4:2 - ver Figura4.9-(a)) e somadores da ferramenta de síntese na linha de recombinação;
- SAD-v10- arquitetura implementada com compressor 8:2 (composto por três compressores 4:2 - ver Figura4.9-(b)) e somadores Kogge-Stone linha de recombinação;
- SAD-v11- arquitetura implementada com compressor 8:2 (composto por quatro compressores 3:2 e um compressor 4:2 - ver Figura4.9-(a)) e somadores Kogge-Stone linha de recombinação.

5.3 Resumo

Este capítulo apresentou uma visão geral da HT em três diferentes tamanhos: 2x2, 4x4 e 8x8. Estas Transformadas são aplicadas na métrica de similaridade SATD que é parte integrante do módulo de Predição Intra e na Estimção de Movimento Fracionário do padrão de compressão HEVC. Foram apresentadas arquiteturas para a métrica SATD em três diferentes níveis de paralelismo (Sequencial, Semi-Paralela e Paralela), bem como arquiteturas implementadas para SATD baseada em múltiplos tamanhos da HT. Também foi apresentado uma exploração dos

circuitos compressores 8:2 com diferentes estruturas e linhas de recombinação, aplicados na métrica SAD. No próximo capítulo são apresentados os resultados de síntese para as arquiteturas implementadas.

6 RESULTADOS DE SÍNTESE E TRABALHOS RELACIONADOS

Nesta seção são apresentados os resultados obtidos para as arquiteturas da SAD e SATD e realizada uma comparação com trabalhos encontrados na literatura. Todas as arquiteturas foram descritas em linguagem de descrição de *hardware* VHDL. As arquiteturas também foram sintetizadas para ASIC (*Application Specific Integrated Circuit*) para CMOS 45nm *Nangate* (NANGATE,) *Open Cell Library* utilizando a ferramenta *Cadence RTL Compiler* (ENCOUNTER, 2015).

6.1 Metodologia de Síntese

Todas arquiteturas apresentadas neste trabalho foram validadas no Simulador *Cadence Incisive* (ENCOUNTER, 2015) com dados de entradas reais extraídos do *software* de referência (HM) (FRAUNHOFER, 2015) do HEVC. O número de portas é o equivalente a portas NAND de 2-entradas. A estimativa da potência dissipada foi realizada com alimentação de 1.1V e dados de vídeo como entradas. A Figura 6.1 apresenta a metodologia para a estimativa de potência dissipada.

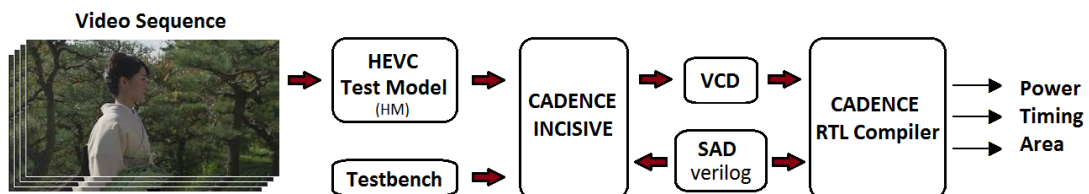


Figura 6.1 – Diagrama da metodologia de síntese.

Os valores de entrada das funções da SATD e SAD foram primeiramente extraídas do HM e armazenadas em arquivos texto. Após, os valores são entregues para a ferramenta de simulação, a qual roda o *testbench* com arquivos do projeto das arquiteturas e gera um arquivo de atividade de chaveamento no formato VCD (*Value Change Dump*). Os arquivos de projeto e VCD são entregues para a ferramenta de síntese, a qual gera relatórios de área, potência e atraso. De acordo com (COSTA, 2002) o número de vetores de amostras necessários para a estimativa de potência em circuitos CMOS é de 10.000 mil vetores. Assim, os resultados de potência deste trabalho foram gerados a partir de uma simulação de 10.000 vetores de entrada para as arquiteturas de SAD e SATD baseada na HT 8x8. Para a arquitetura de SATD, baseada em múltiplos tamanhos da HT, foram utilizados 30.000 vetores de entrada, dos quais 10.000

são usados para gerar os resultados da SATD2x2, 10.000 para gerar a SATD4x4 e 10.000 para gerar a SATD8x8.

6.2 Resultados para SATD - Exploração Arquitetural baseada na HT8x8

Para esta primeira parte foram implementadas três versões de cada arquitetura (sequencial, semi-paralela e totalmente paralela): 1) utilizando somadores da própria ferramenta de sínteses; 2) utilizando somadores e subtratores compressores modificado; 3) utilizando somadores/subtratores compressores proposto. As seguintes sequências de vídeo do CTCs (*Common Test Conditions*) foram utilizadas para a estimação de potência: *Basketballdrive*, *Kimono*, *NebutaFestival*, *ParkScene* e *Cactus* (1920x1080 pixels) e uma frequência alvo de 333 MHz.

A Tabela 6.1 mostra os resultados em ASIC das arquiteturas em termos de área das células, frequência máxima e números de ciclos de relógio por cada operação da SATD.

Tabela 6.1 – Resultados de Síntese

Arquitetura		Área (μm^2)	Freq. Máx. (MHz)	Ciclos de relógio	Throughput (um bilhão de amostras/s)
Sequencial	Ferramenta	14833	338,29	8	2,7
	Comp. Mod.	16729	333,33		
	Comp. Prop.	16250	333,33		
Semi-Paralela	Ferramenta	30809	339,44	2	10,7
	Comp. Mod.	37485	333,44		
	Comp. Prop.	35542	333,44		
Paralela	Ferramenta	45076	411,20	1	21,3
	Comp. Mod.	58722	355,20		
	Comp. Prop.	54853	355,20		

Os gráficos da Figura 6.2, Figura 6.3 e Figura 6.4 apresentam os resultados de potência de *leakage*, potência dinâmica e potência total das arquiteturas SATD sequencial, semi-paralela e paralela, respectivamente. Os resultados são mostrados para cada sequência de vídeo testada. Já o gráfico da Figura 6.5 apresenta a redução média de potência para cada arquitetura.

A função *macrofunction* realiza uma adição com células internas otimizadas, o que causa uma aceleração na propagação do *carry*. Porém, o uso de compressores é favorável para a redução de potência nas arquiteturas de SATD, como pode ser visto na gráfico da Figura 6.5. Embora as arquiteturas com compressores apresentem uma área maior, fato que contribui para

uma maior potência de *leakage*, o aspecto dos compressores de somar vários operandos simultaneamente, com um caminho crítico reduzido, contribui para uma menor potência dinâmica nas arquiteturas de SATD. Este aspecto, redução da potência dinâmica, tem um impacto para uma maior redução de potência total para todas sequências de vídeo testadas. Enquanto que se apresenta uma redução de potência média de 14,7% para a sequencial, 19% para a semi-paralela e 23,6% para a paralela, utilizando somadores e subtratores compressores, tem-se redução de potência média de 16,3% para a sequencial, 21,1% para a semi-paralela e 26,6% para a paralela utilizando somadores/subtratores compressores. A redução de potência obtida com o compressor proposto, comparado ao ganho do compressor modificado, se dá pelo fato dos compressores propostos ocuparem menor área.

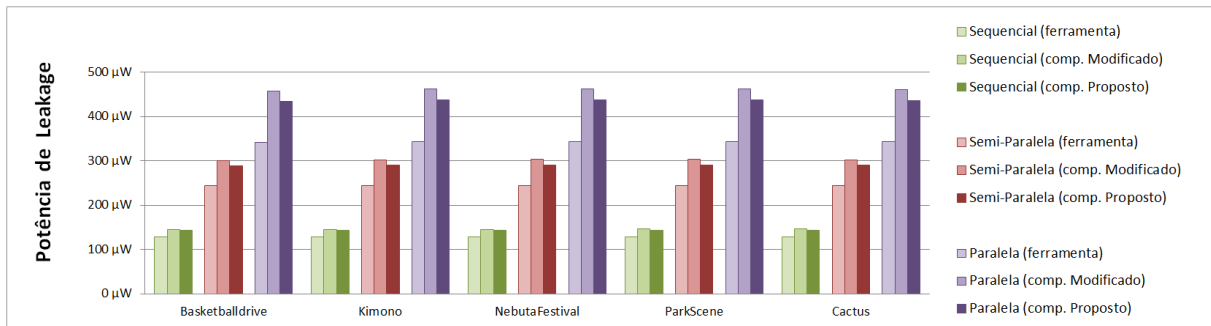


Figura 6.2 – Potência de *leakage* por vídeo para todas versões de arquitetura de SATD.

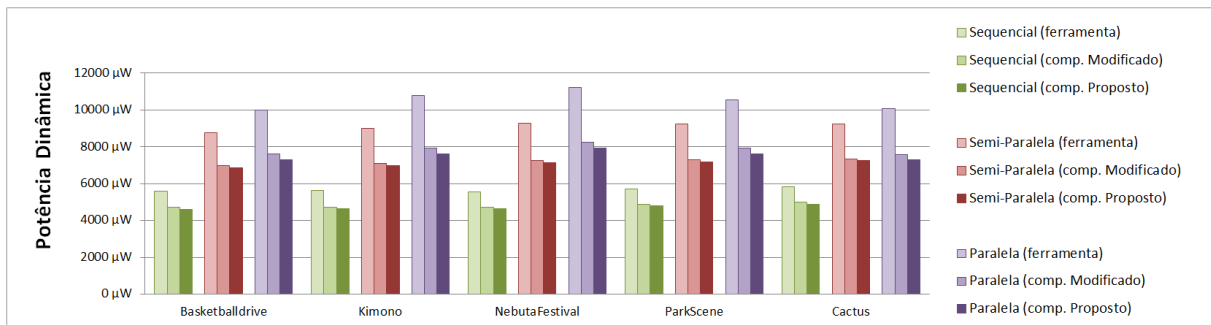


Figura 6.3 – Potência dinâmica por vídeo para todas versões de arquitetura de SATD.

Para as arquiteturas, também foi calculada a média de consumo de energia por cada operação de SATD, como mostra a Tabela 6.2. O cálculo da energia é dado pela equação (6.1).

$$E_{SATD} = C \cdot T \cdot P_{TOTAL} \quad (6.1)$$

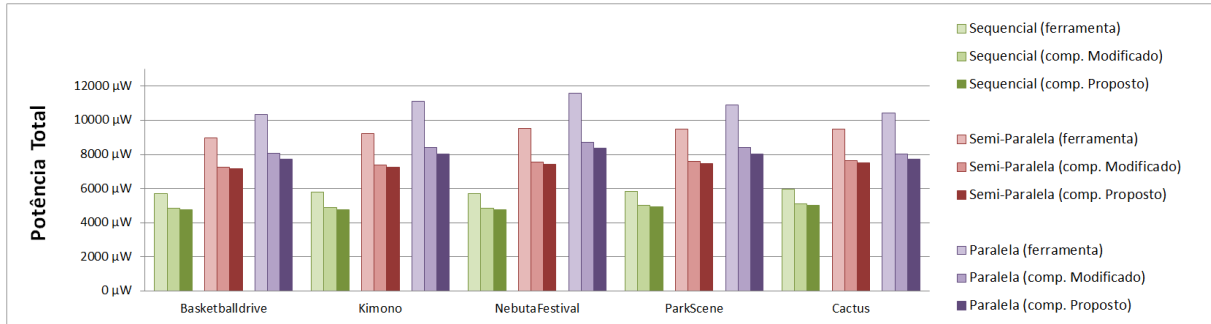


Figura 6.4 – Potência total por vídeo para todas versões de arquitetura de SATD.

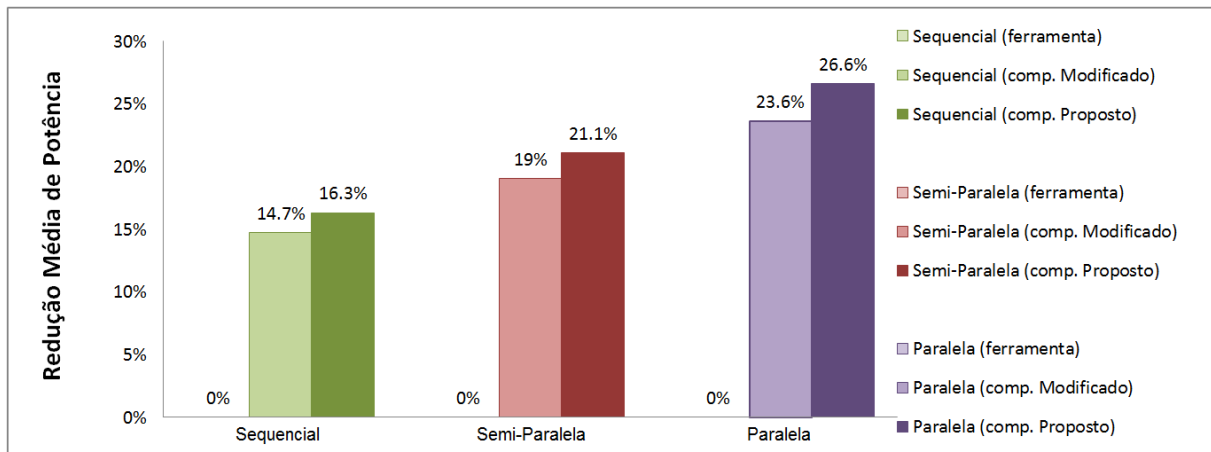


Figura 6.5 – Redução média de Potência total para cada arquitetura.

Na equação (6.1), C é o número de ciclos de relógio da arquitetura após o *pipeline* ser preenchido, T é o período do ciclo de relógio e P_{TOTAL} é a potência total da arquitetura. Os resultados levam em consideração apenas uma operação de SATD para um bloco de amostra 8×8 .

As arquiteturas semi-paralela com somador da ferramenta, com compressores modificado e com compressores proposto, alcançam reduções de energia de 59%, 62% e 62%, respectivamente, comparadas às arquiteturas sequenciais. Este resultado é possível porque o *throughput* aumenta em 4x (reduzindo o tempo computacional total), enquanto que a potência total aumenta em 1,6x, 1,5x e 1,5x. As arquiteturas paralelas com somador da ferramenta, com compressores modificado e com compressores proposto, alcançam reduções de energia de 77%, 79% e 79%, respectivamente, comparadas às arquiteturas sequenciais. Neste caso, o *throughput*

Tabela 6.2 – Energia por operação de SATD

Arquitetura	Ferramenta	Comp. Modificado	Comp. Proposto
Sequencial	139,22 pJ	118,83 pJ	116,47 pJ
Semi-Paralela	56,09 pJ	44,90 pJ	44,23 pJ
Paralela	32,61 pJ	24,90 pJ	23,93 pJ

aumenta em 7,9x enquanto que as potências totais aumentam em 1,9x, 1,7x e 1,6x, respectivamente. Os resultados de *throughput* são mostrados na Tabela 6.1 em amostras/segundo.

6.2.1 Trabalhos Relacionados com a Exploração Arquitetural da SATD

As arquiteturas de *hardware* para SATD são frequentemente incluídas como parte das arquiteturas completas para estimação de movimento (CHEN; HUANG; CHEN, 2004) (KAO; WU; LIN, 2010) ou predição intra (HUANG et al., 2005) (SHAFIQUE; BAUER; HENKEL, 2009). Todas as arquiteturas de *hardware* para SATD encontradas nos trabalhos em (CHEN; HUANG; CHEN, 2004) (DOMINGUES et al., 2011) são baseadas na HT 4x4. Em (DOMINGUES et al., 2011) é proposta uma arquitetura para a HT 8x8, porém visando somente a prototipação para FPGA (Field Programmable Gate Array), e não fornecendo resultados de potência e energia. O trabalho apresentado em (CANCELLIER et al., 2014) aborda o projeto de arquiteturas para SATD eficientes em energia baseadas na HT 4x4.

Os trabalhos da literatura sobre arquiteturas de SATD não têm o foco na eficiência de energia e, além disso, a maioria deles são baseados apenas na HT 4x4, enquanto que essa dissertação de mestrado tem como foco a exploração da HT 8x8. Além disso, os trabalhos da literatura não fazem uma exploração do uso de circuitos somadores eficientes em energia, como nesse trabalho de dissertação.

O gráfico da Figura 6.6 apresenta a comparação das arquiteturas desenvolvidas neste trabalho (somente as arquiteturas com compressores propostos), com duas arquiteturas apresentadas em (CANCELLIER et al., 2014). A arquitetura “*Tree-4stages*” (CANCELLIER et al., 2014) é similar à arquitetura totalmente paralela desenvolvida neste trabalho, enquanto que a arquitetura “*Transpose-2T*” também em (CANCELLIER et al., 2014) é similar à arquitetura sequencial deste trabalho de dissertação. As arquiteturas foram comparadas em termos da métrica energia/*throughput*, desde que todas as arquiteturas apresentadas em (CANCELLIER et al., 2014) têm menor *throughput* do que as desenvolvidas neste trabalho (apresentam 1 milhão de macroblocos/segundo). Devido aos trabalhos relacionados apresentarem resultados de *throughput* em macroblocos/segundo, neste trabalho o *throughput* foi normalizado para macroblocos/segundo, onde um macrobloco é um bloco de 16x16 amostras. Assim, a arquitetura sequencial apresenta um *throughput* de 10,42 milhões de macroblocos/segundo, enquanto que a semi-paralela apresenta um *throughput* de 41,7 milhões de macroblocos/segundo e, finalmente, a arquitetura paralela tem um *throughput* de 83,3 milhões de macroblocos/segundo.

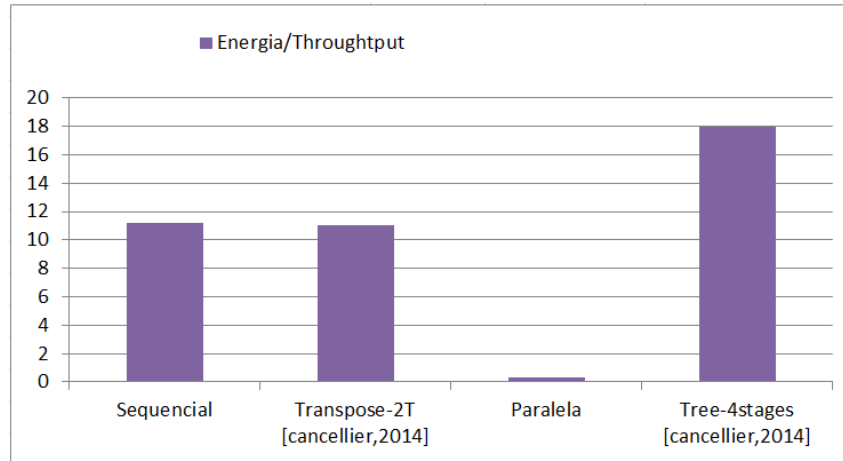


Figura 6.6 – Comparação de Energia/Throughput com arquitetura estado-da-arte (referência).

Tabela 6.3 – Resultados de Síntese

	SATD Ferramenta	SATD Comp. Modificado	SATD Comp. Proposto
Freq. Máx (MHz)	312.8	209.1	209.1
Área (μm^2)	57074	76190	70321

Com a métrica de energia/throughput, pode-se normalizar o consumo de energia de *throughput* de cada arquitetura. Observa-se que a arquitetura totalmente paralela desenvolvida neste trabalho, é 62x e 37x mais eficiente em termos de energia/throughput do que as arquiteturas “Tree-4stages” e “Transpose-2T” da literatura, respectivamente.

6.3 Resultados para SATD - Implementação Baseada em Múltiplos Tamanhos da HT

Para esta segunda etapa foi desenvolvida uma arquitetura de *hardware* para a SATD baseada em múltiplos tamanhos da HT. O *hardware* dos cálculos das matrizes menores são re-aproveitados para realizar o cálculo das matrizes maiores. A arquitetura foi implementada com somadores da própria ferramenta de síntese, com somadores e subtratores compressores e com somadores/subtratores proposto. Para a estimativa de potência foram utilizadas três sequências de vídeo: *ParkScene* e *Kimono* (1920x1080 *pixels*) e *NebutaFestival* (2560x1600 *pixels*). As arquiteturas foram sintetizadas para uma frequência alvo de 125MHz. A Tabela 6.3 apresenta os resultados de síntese em termos de área de células e frequência máxima para as arquiteturas de múltiplos tamanhos da SATD usando somadores e subtratores compressores modificado e somadores/subtratores proposto, ambos comparados à arquitetura implementada com somador da ferramenta de síntese (*macrofunction*). Já a Tabela 6.4 apresenta os resultados de potência obtidos para cada arquitetura por sequência de vídeo.

Tabela 6.4 – Resultados de Potência

Vídeo	Arquitetura versão	Potência de <i>Leakage</i> (μW)	Potência Dinâmica (μW)	Potência Total (μW)	Redução de Potência (%)
Kimono	Ferramenta	461,9	7655,9	8117,9	-
	Comp. Mod.	630,7	6972,5	7603,2	6.34
	Comp. Prop.	595,9	6601,0	7197,0	11.34
NebutaFestival	Ferramenta	459,7	8017,6	8477,3	-
	Comp. Mod.	630,9	7513,8	8144,8	3.92
	Comp. Prop.	597,5	7079,8	7677,3	9.44
ParkScene	Ferramenta	459,5	7672,7	8132,2	-
	Comp. Mod.	631,7	7227,4	7859,1	3.36
	Comp. Prop.	597,4	6767,6	7364,9	9.43

Como dito anteriormente, a função *macrofunction* realiza uma adição com células otimizadas, o que é benéfico para a aceleração na propagação do *carry*. Entretanto, o uso de compressores é favorável para a redução de potência na arquiteturas de SATD, como pode ser visto na Tabela 6.4. Como as arquiteturas SATD com somadores compressores apresentam menor dissipação de potência total, como mostrado anteriormente, logo, esse aspecto tem um impacto favorável para as transformadas de múltiplos tamanhos, que proporcionam numa melhor redução de potência total, para todas as sequências de vídeo testadas, com uma redução de potência média de 4,54% para as arquiteturas com somadores e subtratores compressores modificados e 10,07% para com somadores/subtratores compressores propostos.

6.3.1 Trabalhos Relacionados com SATD Usando HTs de Múltiplos Tamanhos

O trabalho em (CANCELLIER et al., 2014) visa ao projeto de uma arquitetura de SATD eficiente em energia, baseada na HT 4x4. Arquiteturas baseadas na HT 4x4 também são descritas em (KAO; WU; LIN, 2010) (LIN et al., 2009). Porém, poucos trabalhos exploram arquiteturas de *hardware* para o cálculo da SATD baseada na HT 8x8. O *software* de referência (HM) (FRAUNHOFER, 2015) do padrão HEVC utiliza a SATD na FME e na predição intra com diferentes tamanhos para a HT (2x2, 4x4 e 8x8), dependendo do tamanho do bloco de predição. O trabalho em (SILVEIRA et al., 2015) apresenta uma arquitetura de *hardware* para a SATD baseada na HT 8x8, incluindo dois módulos de uma dimensão (1-D) da Hadamard 8x8 (horizontal e vertical) com *buffer* sequencial-paralelo para realizar a matriz transposta. Entretanto, nenhum dos trabalhos mencionados exploram o uso de múltiplos tamanhos da HT na

SATD. Arquiteturas de multitransformada para SATD do padrão de compressão H.264/AVC foram propostas em (AGOSTINI et al., 2006), (CHEN et al., 2008), mas apenas os tamanhos de HT 2x2 e 4x4 foram explorados. Por outro lado, arquiteturas de multitransformada para SATD baseadas em tamanhos de HT 4x4 e 8x8 foram propostas em (HWANGBO; KYUNG, 2010), (LIU et al., 2011), ambos para o padrão H.264/AVC. Uma arquitetura de multitransformada para SATD para o novo padrão HEVC foi proposta em (HE et al., 2015), onde uma estratégia de reuso de dados é explorada para reduzir o custo de *hardware* da transformada. Entretanto, nenhum destes trabalhos relacionados exploram a combinação do reuso da transformada com o uso de somadores compressores eficientes para uma implementação eficiente com redução de potência da arquitetura de SATD.

6.4 Resultados para SAD

Para os resultados de SAD, as seguintes sequências de vídeos CTCs (*Common Test Conditions*)(BOSSSEN, 2013) foram utilizadas para a estimação de potência: sequência de vídeos de 1920x1080 *pixels*, *Basketballdrive*, *BQTerrace*, *Cactus* e *Kimono*, e as sequências de vídeos de 2560x1600 *pixels*, *PeopleOnStreet* e *Traffic*. A Tabela 6.5 mostra os resultados da síntese das onze versões de arquiteturas de SAD em termos de área, portas lógicas (*gates* equivalente a portas NAND) e frequência máxima.

Tabela 6.5 – Resultados de Síntese

Versões de Arquiteturas	Freq. Max. (MHz)	Gates	Cell Area (μm^2)
v1	128.67	3504	2803
v2	100.00	3141	2513
v3	92.53	3163	2530
v4	90.07	3102	2482
v5	89.81	3114	2491
v6	101.56	3258	2606
v7	111.94	3445	2756
v8	101.56	3258	2606
v9	111.94	3445	2756
v10	101.56	3258	2606
v11	111.94	3445	2756

A potência total de cada arquitetura para cada sequência de vídeo é mostrada no gráfico da Figura 6.7.

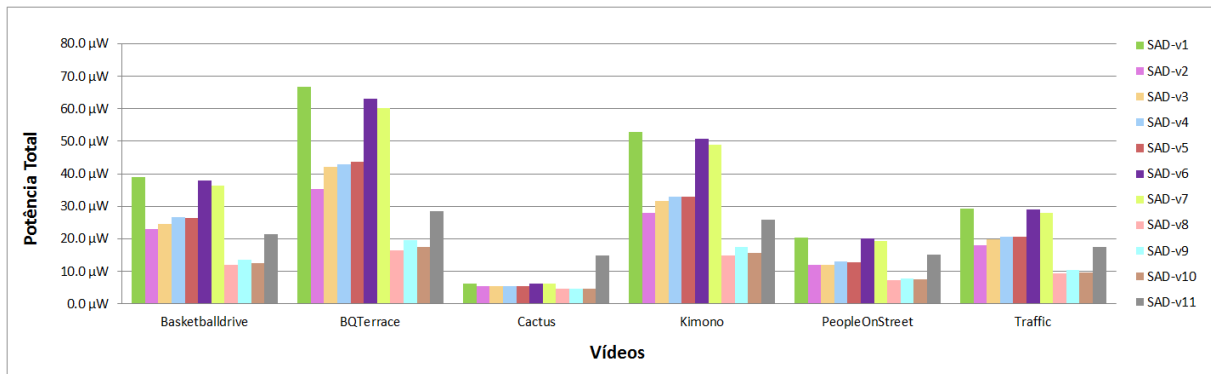


Figura 6.7 – Potência Total por vídeo para todas as versões de arquitetura de SAD.

O gráfico da Figura 6.7 mostra que as arquiteturas de SAD que não utilizam compressores (versões 1, 6 e 7) têm uma maior dissipação de potência comparadas às arquiteturas com compressores. Isto ocorre porque os compressores 8:2 realizam oito somas simultâneas com um caminho crítico reduzido. As versões 8 (linha de recombinação com somador da ferramenta) e 10 (linha de recombinação com somador Kogge-Stone) apresentam as menores potências dissipadas para todas as sequências de vídeos. O somador Kogge-Stone apresenta um baixo *fan-out* a cada estágio, o qual aumenta o desempenho comparado às outras versões usando somadores *full-adder* e *full-adder* baseado em MUX. Por outro lado, o somador da ferramenta utiliza células otimizadas. Essas células otimizadas usam o menor número de transistores possível e são solicitadas no processo de síntese, de acordo com a métrica a ser buscada (maior desempenho ou menor potência). A Figura 6.8 mostra a redução de potência média (média entre as sequências de vídeo) de cada versão de arquitetura de SAD comparada com a versão 1.

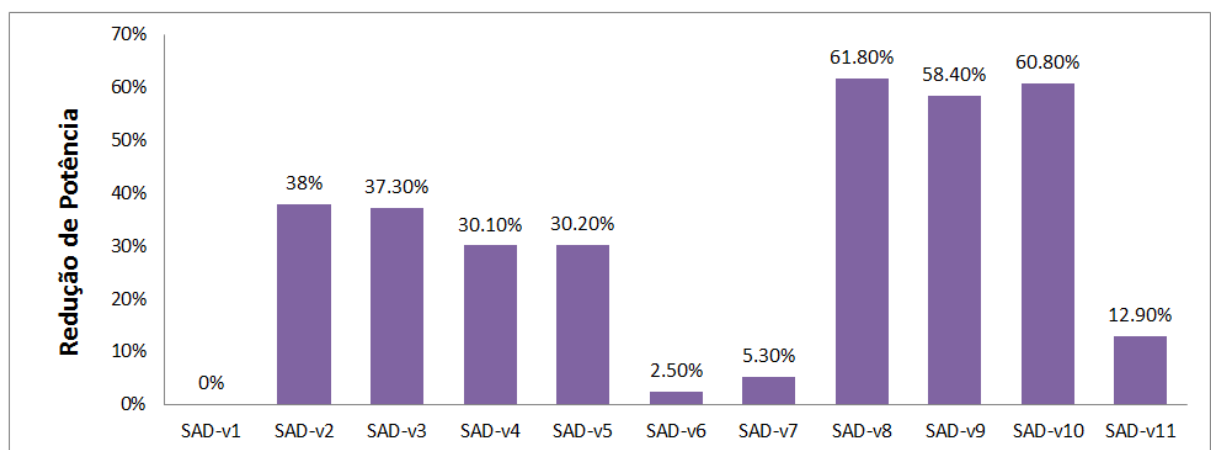


Figura 6.8 – Redução Média de dissipação de potência.

Foi realizada uma comparação entre as versões das arquiteturas em termos de energia por operação de SAD. O cálculo da energia por operação de SAD usa a mesma equação mostrada anteriormente para a SATD (ver em (6.1)). Todas as arquiteturas realizam a operação de SAD de oito amostras a cada ciclo de relógio e operam em um mesmo período de 20ns (*target timing*). A tabela 6.6 apresenta o resultado de energia por operação de SAD de cada versão de arquitetura.

Tabela 6.6 – Energia por operação de SAD

Sequência de Vídeo	Energia por SAD (pJ)										
	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11
<i>Basketballdrive</i>	0.78	0.46	0.49	0.54	0.53	0.76	0.73	0.24	0.27	0.25	0.43
<i>BQTerrace</i>	1.33	0.71	0.84	0.86	0.87	1.26	1.20	0.33	0.40	0.35	0.57
<i>Cactus</i>	0.12	0.11	0.11	0.11	0.11	0.12	0.12	0.09	0.10	0.10	0.30
<i>Kimono</i>	1.06	0.56	0.66	0.66	0.66	1.02	0.98	0.30	0.35	0.31	0.52
<i>PeopleOnStreet</i>	0.41	0.24	0.24	0.26	0.26	0.40	0.39	0.15	0.16	0.15	0.30
<i>Traffic</i>	0.59	0.36	0.40	0.41	0.41	0.58	0.56	0.19	0.21	0.19	0.35

6.4.1 Trabalhos Relacionados com a Métrica SAD

As arquiteturas de SAD normalmente são utilizadas como parte das arquiteturas completas da ME (PORTO et al., 2010) ou Predição Intra (DINIZ et al., 2010) na codificação de vídeo. Alguns trabalhos focam na otimização da unidade de SAD, que é um dos pontos centrais deste projeto de dissertação. O trabalho em (VANNE et al., 2006) propõe uma arquitetura de *hardware* para a SAD de 3 estágios contendo unidades para cálculo da diferença absoluta, módulo de soma usando compressores e módulo para determinação do menor valor de SAD. Árvores de somadores do tipo CSA (*Carry Save Adder*) são utilizadas no módulo de soma. Em (YUFEI; XIUBO; QIN, 2007) é proposto um algoritmo de SAD e uma arquitetura de SAD de 3 estágios empregando o uso de compressores 4:2 em árvore para realizar a acumulação. Duas arquiteturas para ME usando compressores 4:2 e 8:2, para a realização das somas na árvore de SAD, foram propostas por (PORTO et al., 2010). O trabalho apresentado em (DINIZ et al., 2010) introduz uma arquitetura de *hardware* para a predição intra utilizando compressores 4:2 na unidade de SAD.

Os trabalhos em (PORTO et al., 2010; DINIZ et al., 2010; VANNE et al., 2006; YUFEI; XIUBO; QIN, 2007) não apresentam resultados de potência, não sendo possível uma comparação direta com este trabalho. Os trabalhos em (NALLURI; ALVES; NAVARRO, 2013) e (NALLURI; ALVES; NAVARRO, 2014) apresentam três diferentes níveis de paralelismo para

arquitetura de SAD. Entretanto, as arquiteturas são sintetizadas apenas para FPGA, o que é menos eficiente do que ASIC para resultados de avaliação de potência, o que não permite uma comparação justa com este trabalho. Além de todos estes fatores, os trabalhos relacionados não utilizam dados reais de vídeo para a estimação de potência, em oposição a este trabalho que utiliza dados de entradas reais de seis sequências de vídeos distintas. Além disso, neste trabalho é avaliado o uso de diferentes configurações de somadores compressores dentro da unidade de SAD e o impacto destas para a estimação de potência, os quais não são mostrados nos trabalhos relacionados.

6.5 Resumo

Neste capítulo foram apresentados todos os resultados obtidos neste trabalho em termos de área, frequência máxima, *throughput*, energia e dissipação de potência, assim como foram apresentados trabalhos relacionados ao tema e realizada as devidas comparações.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou um estudo sobre arquiteturas dedicadas para a compressão de vídeo do padrão HEVC. Em particular, foram exploradas algumas técnicas para o aumento de desempenho e redução de potência para arquiteturas dedicadas das métricas de similaridade SATD e SAD, que são partes integrantes dos módulos de Estimação de Movimento e Predição Intra na codificação do padrão HEVC. Devido à característica das Transformadas Hadamard, encontradas na métrica SATD, e da árvore de somadores encontrada na métrica SAD, que apresentam um elevado número de operadores aritméticos de soma e subtração, foram exploradas neste trabalho algumas técnicas de paralelismo do cálculo da HT 8x8, e de múltiplos tamanhos, que permitem a soma de vários operandos simultaneamente, assim como a utilização de somadores compressores eficientes para o cálculo da SAD.

Foram propostas arquiteturas de *hardware* em três diferentes níveis de paralelismo para o cálculo da HT 8x8, utilizando circuitos compressores, sendo essas Sequencial, Semi-Paralela e Totalmente Paralela. As arquiteturas semi-paralela e totalmente paralela reduzem o consumo de energia comparadas com a arquitetura sequencial. A arquitetura totalmente paralela apresentou eficiência de 62x e 37x, em termos de energia/*throughput*, em relação às duas arquiteturas encontradas em trabalhos relacionados, respectivamente. A aplicação de somadores e subtratores compressores possibilitou uma redução média de potência total nas arquiteturas HT8x8, sendo de 14,7% na sequencial, 19% para a semi-paralela e 23,6% para a paralela. Por outro lado, utilizando o somador/subtrator compressor, proposto no âmbito dessa dissertação, houve reduções médias de potência total de 16,3% para a sequencial, 21,1% para a semi-paralela e 26,6% para a paralela.

As arquiteturas de SATD baseadas em múltiplos tamanhos da HT apresentaram uma redução média de potência de 4,54% para as arquiteturas com somadores e subtratores compressores modificados e 10,07% para com somadores/subtratores compressores propostos.

Para a arquitetura de SAD foram exploradas 11 diferentes arquiteturas, sendo 7 delas com somadores compressores, que apresentaram os melhores resultados em termos de redução de potência. Em particular, quando analisando a energia por operação de SAD, as versões usando na linha de recombinação somadores do tipo Kogge-Stone (v10) e com o somador da ferramenta (v8) obtiveram os melhores resultados. Isto se explica pelo fato do somador Kogge Stone apresentar um melhor desempenho, visto que o *carry* é propagado de forma mais eficiente, e pelo fato do somador da ferramenta utilizar blocos otimizados.

7.1 Trabalhos Futuros

Os estudos realizados com as implementações de arquiteturas dedicadas para as métricas de similaridade SATD e SAD, oportunizaram publicações e submissões de artigos para eventos internacionais e revista internacional. Em relação ao trabalho com a métrica SAD, o trabalho intitulado: Power-Efficient Sum of Absolute Differences Architecture using Adder Compressors foi aceito para publicação no ICECS (International Conference on Electronics, Circuits and Systems), que será apresentado em dezembro de 2016.

Em relação aos resultados obtidos com a métrica SATD, sobre exploração arquitetural e com Transformadas Hadamard de múltiplos tamanhos, os seguintes artigos foram publicados ou submetidos para avaliação:

- SATD Hardware Architecture Based on 8x8 Hadamard Transform for HEVC Encoder (publicado no ICECS2015);
- Applying Adder and Subtractor Compressors to Sum of Absolute Transformed Differences Architecture for Low Power Video Encoding (submetido à revista IET - Electronic Letters);
- Low Power SATD Architecture Employing Multiple Sizes Hadamard Transforms and Adder Compressors (submetido ao ISCAS2017).

Diante dos resultados obtidos ao longo desta dissertação com resultados expressivos obtidos nas explorações das métricas SAD e SATD, que culminaram em importantes publicações e submissões para eventos qualificados, abre-se uma perspectiva de novos trabalhos neste linha. Desta forma, os novos desafios devem envolver:

- Exploração da implementação de somadores compressores 8:2, baseados em outros somadores compressores da literatura, tais como, 5:2 e o 7:2.
- Implementação de módulos de estimação de movimento e de predição intra do padrão HEVC, usando os módulos SATD e SAD desenvolvidos.
- Aplicação de operadores aritméticos imprecisos nas arquiteturas de SATD e SAD implementadas.

REFERÊNCIAS

- AGOSTINI, L. Desenvolvimento de arquiteturas de alto desempenho dedicadas à compressão de vídeo segundo o padrão h. 264/avc. **Universidade Federal do Rio Grande do Sul. Instituto de Informática. Programa de Pós-Graduação em Computação**, 2007.
- AGOSTINI, L. et al. High throughput multitransform and multiparallelism ip for h. 264/avc video compression standard. In: IEEE. **2006 IEEE International Symposium on Circuits and Systems**. [S.l.], 2006. p. 4–pp.
- BOSSEN, F. Common test conditions and software configurations. In: **JCT-VC document no. L1100**. [S.l.: s.n.], 2013.
- CANCELLIER, L. H. et al. Energy-efficient hadamard-based satd architectures. In: ACM. **Proceedings of the 27th Symposium on Integrated Circuits and Systems Design**. [S.l.], 2014. p. 36.
- CHEN, L.-F. et al. Analysis and architecture design of multi-transform architecture for h. 264/avc intra frame coder. In: IEEE. **2008 IEEE International Conference on Multimedia and Expo**. [S.l.], 2008. p. 145–148.
- CHEN, T.-C.; HUANG, Y.-W.; CHEN, L.-G. Fully utilized and reusable architecture for fractional motion estimation of h. 264/avc. In: IEEE. **Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on**. [S.l.], 2004. v. 5, p. V–9.
- CORREA, G. et al. Performance and computational complexity assessment of high-efficiency video encoders. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 22, n. 12, p. 1899–1909, 2012.
- CORRÊA, G. R. **Computational Complexity Reduction and Scaling for High Efficiency Video Encoders**. Thesis (PhD) — Universidade de Coimbra, 2014.
- COSTA, E. A. C. d. Operadores aritméticos de baixo consumo para arquiteturas de circuitos dsp. 2002.
- DINIZ, C. et al. Performance enhancement of h.264/avc intra frame prediction hardware using efficient 4-2 and 5-2 adder-compressors. In: **SBCCI**. [S.l.: s.n.], 2010. p. 157–162. ISBN 978-1-4503-0152-7.
- DINIZ, C. M. **Dedicated and reconfigurable hardware accelerators for high efficiency video coding standard**. Thesis (PhD) — Universidade Federal do Rio Grande do Sul, 2015.
- DOMINGUES, J. S. et al. High throughput 4x4 and 8x8 satd similarity criteria architectures for video coding applications. In: CITESEER. **2011 VII Designer Forum (DF)**. [S.l.], 2011. p. 115.
- ENCOUNTER, R. C. Compiler v. 8.10. **Available at: www.cadence.com**. Accessed **November**, v. 11, 2015.
- FRAUNHOFER. **HEVC Reference Software**. 2015. Available from Internet: <https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/>.

GONZALEZ, R.; WOODS, R. Processamento de imagens digitais, tradução do original digital image processing. **Edgard Blücher, Sao Paulo**, 2000.

HE, G. et al. High-throughput power-efficient vlsi architecture of fractional motion estimation for ultra-hd hevc video encoding. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, IEEE, v. 23, n. 12, p. 3138–3142, 2015.

HUANG, Y.-W. et al. Analysis, fast algorithm, and vlsi architecture design for h. 264/avc intra frame coder. **IEEE Transactions on Circuits and systems for Video Technology**, IEEE, v. 15, n. 3, p. 378–401, 2005.

HWANGBO, W.; KYUNG, C.-M. A multitransform architecture for h. 264/avc high-profile coders. **IEEE Transactions on Multimedia**, IEEE, v. 12, n. 3, p. 157–167, 2010.

ITU-T; 1, I. J. Advanced video coding for generic audiovisual services. **ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)**, 2011.

ITU-T; ISO/IEC. High efficiency video coding. **ITU-T Recommendation H.265 and ISO/IEC 23008-2**, 2013.

KAO, C.-Y.; WU, C.-L.; LIN, Y.-L. A high-performance three-engine architecture for h. 264/avc fractional motion estimation. **IEEE transactions on very large scale integration (VLSI) systems**, IEEE, v. 18, n. 4, p. 662–666, 2010.

KIM, T.; JAO, W.; TJANG, S. Arithmetic optimization using carry-save-adders. In: **ACM. Proceedings of the 35th annual Design Automation Conference**. [S.l.], 1998. p. 433–438.

KOGGE, P. M.; STONE, H. S. A parallel algorithm for the efficient solution of a general class of recurrence equations. **IEEE Transactions on Computers**, C-22, n. 8, p. 786–793, Aug 1973. ISSN 0018-9340.

LIN, Y. K. et al. A 140-mhz 94 k gates hd1080p 30-frames/s intra-only profile h.264 encoder. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 19, n. 3, p. 432–436, March 2009. ISSN 1051-8215.

LIU, J. et al. A full-mode fme vlsi architecture based on $8 \times 8/4 \times 4$ adaptive hadamard transform for qfhd h. 264/avc encoder. In: **IEEE. 2011 IEEE/IFIP 19th International Conference on VLSI and System-on-Chip**. [S.l.], 2011. p. 434–439.

MANOEL, E. T. M. et al. Codificação de vídeo h. 264: estudo de codificação mista de macroblocos. Florianópolis, SC, 2007.

NALLURI, P.; ALVES, L. N.; NAVARRO, A. A novel sad architecture for variable block size motion estimation in hevc video coding. In: **International Symposium on System on Chip (SoC)**. [S.l.: s.n.], 2013.

NALLURI, P.; ALVES, L. N.; NAVARRO, A. High speed sad architectures for variable block size motion estimation in hevc video coding. In: **ICIP**. [S.l.: s.n.], 2014. p. 1233–1237. ISSN 1522-4880.

NANGATE, S. California (2008). 45nm open cell library. URL {<http://www.nangate.com>, year=2008.

OKLOBDZIJA, V. G.; VILLEGER, D.; LIU, S. S. A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach. **IEEE Trans. on Computers**, v. 45, n. 3, p. 294–306, Mar 1996. ISSN 0018-9340.

PARHAMI, B. **Computer arithmetic**. [S.l.]: Oxford university press, 1999.

PORTO, M. et al. Motion estimation architecture using efficient adder-compressors for hdtv video coding. **Journal of Integrated Circuits and Systems**, v. 5, n. 1, p. 78–88, Mar 2010.

RICHARDSON, I. E. **H. 264 and MPEG-4 video compression: video coding for next-generation multimedia**. [S.l.]: John Wiley & Sons, 2004.

SEIDEL, I. et al. Análise do impacto de pel decimation na codificação de vídeos de alta resolução. 2014.

SHAFIQUE, M.; BAUER, L.; HENKEL, J. A parallel approach for high performance hardware design of intra prediction in h. 264/avc video codec. In: **IEEE. 2009 Design, Automation & Test in Europe Conference & Exhibition**. [S.l.], 2009. p. 1434–1439.

SHI, Y. Q.; SUN, H. **Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards**. [S.l.]: CRC press, 1999.

SILVEIRA, B. et al. Satd hardware architecture based on 8x8 hadamard transform for hevc encoder. In: **2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)**. [S.l.: s.n.], 2015. p. 576–579.

SULLIVAN, G. J. et al. Overview of the high efficiency video coding (hevc) standard. **IEEE Trans. Circuits Syst. Video Technol.**, v. 22, n. 12, p. 1649–1668, Dec 2012. ISSN 1051-8215.

SZE VIVIENNE; BUDAGAVI, M. G. S. **High Efficiency Video Coding (HEVC)**. [S.l.: s.n.].

VANNE, J. et al. A high-performance sum of absolute difference implementation for motion estimation. **IEEE Trans. Circuits Syst. Video Technol.**, v. 16, n. 7, p. 876–883, July 2006. ISSN 1051-8215.

VANNE, J. et al. Comparative rate-distortion-complexity analysis of hevc and avc video codecs. **IEEE Trans. Circuits Syst. Video Technol.**, v. 22, n. 12, p. 1885–1898, Dec 2012. ISSN 1051-8215.

WEINBERGER, A. 4-2 carry-save adder module. **IBM Technical Disclosure Bulletin**, v. 23, n. 8, p. 3811–3814, 1981.

WIEGAND, T. et al. Rate-constrained coder control and comparison of video coding standards. **IEEE Transactions on circuits and systems for video technology**, IEEE, v. 13, n. 7, p. 688–703, 2003.

YUFEI, L.; XIUBO, F.; QIN, W. A high-performance low cost sad architecture for video coding. **IEEE Transactions on Consumer Electronics**, v. 53, n. 2, p. 535–541, May 2007. ISSN 0098-3063.