

Relatório 4: Maximal Overlap Discreat Wavelet Transform

=

1st Matheus Araújo de Oliveira

2nd Jessica Regina dos Santos

3rd Gustavo Hideki Guenka Vale

4th Fábio Augusto Dittrich

5th Guilherme Liberali

6th Gabriel Raul Marino

I. INTRODUCTION

O nosso trabalho foca em implementar uma transformada wavelet em hardware, mais especificamente uma MODWT (Maximun Overlap Discreat Wavelet Transform) [1]. Neste relatório primeiramente iremos falar sobre a arquitetura em hardware das transformadas, como foi feita a verificação do algoritmo e resultados de síntese no FPGA.

II. DESING E ARQUITETURA DO HARDWARE

O diagrama de blocos da figura 1 mostra o design de todos os componentes que foram utilizados no nosso projeto. Existem três componentes principais: a transformada direta (T-dir) que vai decompor o sinal de entrada em frequências altas e baixas, a transformada inversa (T-inv) que vai recompor o sinal que foi decomposto pela transformada inversa, e shift register (shift-reg) que vai alinhar as entradas de uma inversa que estão desalinhadas.

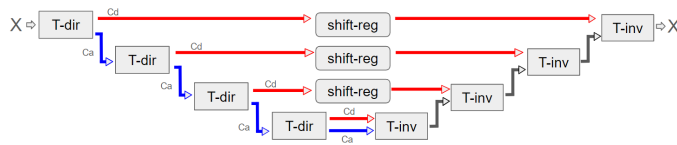


Fig. 1. Diagrama de blocos.

A. Transformada Direta e Inversa

Conforme [1] o algoritmo das transformadas corresponde a multiplicações e acumulações de um vetor de entrada unidimensional por coeficientes fixos, podemos ver o pseudo código da transformada direta descrito na figura 2.

Sendo assim com base nas implementações em hardware da DWT (Discreat Wavelet Transform) descritas em [2] decidimos utilizar a forma transposta de se calcular um filtro FIR (Finite Impulse Response) para computar o cálculo das transformadas wavelets.

A forma transposta de se calcular um filtro FIR está ilustrada na figura 3, nela podemos ver que todos os multiplicadores estão sendo usados em paralelo, o resultado dessas

multiplicações está sendo atrasado em um pulso de clock por um registrador, e então é feita a soma sucessiva desses resultados. Conforme [2] e [3] esta é uma das formas mais efetivas de se calcular um filtro FIR.

A única diferença entre o cálculo de um filtro FIR e uma MODWT é a decimação, conforme vemos na figura 2 o cálculo do vetor unitário de entrada tem como índice a equação (1), onde j é o nível da transformada wavelet.

$$k = 2^j - 1 \quad (1)$$

sendo assim tivemos que alterar a forma transposta acrescentando mais registradores entre as somas, e o número desses registradores é igual à k .

```

For  $t = 0, \dots, N - 1$ , do the outer loop:
  Set  $k$  to  $t$ .
  Set  $\tilde{W}_{j,t}$  to  $\tilde{h}_0 \tilde{V}_{j-1,k}$ , and set  $\tilde{V}_{j,t}$  to  $\tilde{g}_0 \tilde{V}_{j-1,k}$ .
  For  $n = 1, \dots, L_1 - 1$ , do the inner loop:
    Decrement  $k$  by  $2^{j-1}$ .
    If  $k < 0$ , set  $k$  to  $k \bmod N$ .
    Increment  $\tilde{W}_{j,t}$  by  $\tilde{h}_n \tilde{V}_{j-1,k}$ , and increment  $\tilde{V}_{j,t}$  by  $\tilde{g}_n \tilde{V}_{j-1,k}$ .
  End of inner loop.
End of outer loop.
    
```

Fig. 2. pseudocódigo para a transformada direta.

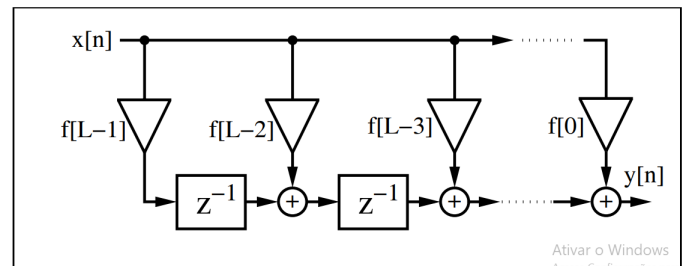


Fig. 3. Forma Transposta de se calcular um filtro FIR

B. Shift Register

Conforme em [2] e [3] a forma transposta atrasa a saída em alguns pulsos de clock, esses atrasos nós chamamos de

delay, portanto cada vez que o sinal de entrada passa por uma transformada direta ele é atrasado. Sendo assim necessário sincronizar os coeficientes de approximation e detail na entrada de cada inversa, para isso nós usamos shift registers.

III. SIMULAÇÃO E VALIDAÇÃO DO DESING

Em [1] temos o pseudo código e o código em matlab para a transformada wavelet, através dele nós geramos o nosso golden model. A Partir disso no nosso teste bench nós extraímos todos os coeficientes de decomposição do sinal e escrevemos esses coeficientes em um arquivo .hex, depois disso nós comparamos as forma de onda com o nosso golden model. A figura 4 contém as formas de onda dos coeficientes de approximation geradas pela nossa simulação no modelsim e a figura 5 contém a forma de onda gerada pelo nosso golden model em software.

Além disso, nós fizemos a diferença entre a entrada do algoritmo e a saída da última transformada inversa, vimos que a diferença foi aproximadamente zero.

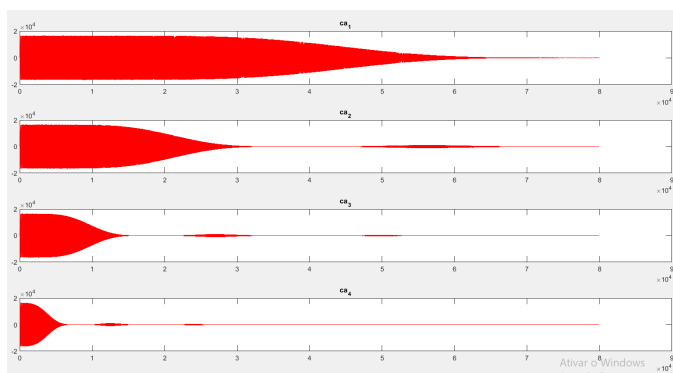


Fig. 4. Forma de onda geradas em Hardware.

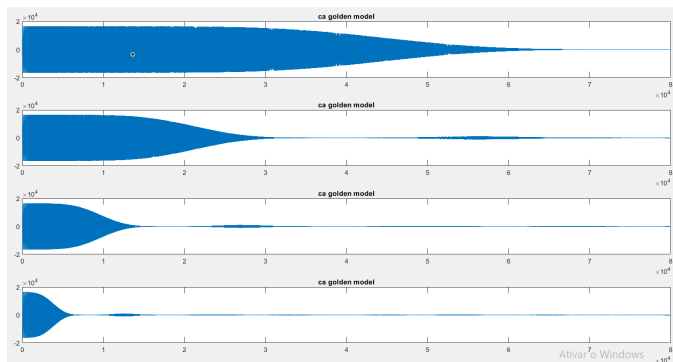


Fig. 5. Formas de onda geradas em Software.

IV. RESULTADOS DE SÍNTESE

O projeto foi compilado utilizando o dispositivo Cyclone IV EP4CE115F29C7 e os dados gerados estão na figura 6. A frequência maxima(Fmax) obtida foi de 92.61 Mhz.

Flow Status	Successful - Wed Nov 29 17:13:32 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	modwt
Top-level Entity Name	modwt
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	15,862 / 114,480 (14 %)
Total registers	4395
Total pins	165 / 529 (31 %)
Total virtual pins	0
Total memory bits	11,412 / 3,981,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

Fig. 6. Dados de Síntese.

V. CONCLUSÃO

Tivemos êxito em implementar uma MODWT em hardware, além disso obtivemos um bom resultado no sentido de ocupação de recursos da placa e frequência máxima, que são a principal motivação para a implementação desse algoritmo.

REFERENCES

- [1] D. Percival and A. Walden, *Wavelet Methods for Time Series Analysis*, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2000.
- [2] M. Bahoura and H. Ezzaidi, "Real-time implementation of discrete wavelet transform on fpga," 11 2010, pp. 191 – 194.
- [3] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, ser. Signals and Communication Technology. Springer Berlin Heidelberg, 2013. [Online]. Available: <https://books.google.com.br/books?id=MBz2CAAAQBAJ>