

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

The Discrete Wavelet Transform provides a multiresolution representation of signals. The transform can be implemented using filter banks. In this thesis, different architectures for the Discrete Wavelet Transform have been implemented. For each of them, parameters such as area, performance and power consumption were discussed. Based on the application and the constraints imposed, the appropriate architecture can be chosen. For the Daubechies length-4 orthogonal filter, three architectures were implemented, i.e., the polyphase architecture, the polyphase with fully parallel DA architecture, and the polyphase with modified DA architecture. It is seen that, in applications which require low area and power consumption, e.g., in mobile applications, the polyphase with modified DA architecture is most suitable and for applications which require high throughput, e.g., real-time applications, the polyphase with DA architecture is more suitable.

The biorthogonal wavelets, with different number of coefficients in the low pass and high pass filters, increases the number of operations and the complexity of the design, but they have better SNR than the orthogonal filters. For the Daubechies 9/7 biorthogonal filter, two different architectures were implemented, i.e., the polyphase architecture, and the polyphase with modified DA architecture. It is seen that the polyphase architecture has better throughput while the polyphase with modified DA architecture has lower area and lower power consumption.

A scalable architecture for computation of higher octave DWT has been presented. The architecture was implemented using the Daubechies length-4 filter for a signal length of 15. The simulation results verify the functionality of the design. The proper scheduling of the wavelet coefficients written to the RAM ensures that, when the coefficients are finally read back from the RAM, they are available in the required order for further processing. The proposed architecture is simple since further levels of decomposition can be achieved using identical processing elements. It is easily scalable to different signal lengths and filter orders for use in different applications. The architecture enables fast computation of DWT with parallel processing. It has low memory requirements and consumes low power.

6.2 Future Work

Synthesis filter banks to compute the inverse DWT, i.e., IDWT can be implemented using similar architectures for the corresponding analysis filter banks.

The architectures of the filter banks can be further improved using techniques such as Reduced Adder Graph, Canonic Signed Digit coding and Hartley's common subexpression sharing among the constant coefficients. Also, in the case of orthogonal filters with mirror coefficients, the transpose form of the filters yields a good architecture; this can be implemented and compared with the others.

The proposed higher octave DWT architecture can be extended to include symmetric signal extension. The use of symmetric extension in image compression applications reduces the distortion at boundaries of reconstructed image and provides improved SNR.

In memory intensive applications such as image and video processing, memory accesses could be the dominant source of power dissipation, as reading and writing to memory involves switching of highly capacitive address busses. Methods such as gray

code addressing can be incorporated into the architecture to reduce this power dissipation.

As the DWT hierarchy increases, the required precision of the wavelet coefficients also increases. In the proposed architecture, the coefficients at all levels are scaled to have the same precision. While this reduces the hardware requirements, the accuracy of the coefficients is compromised as the number of levels increases. Therefore, the architecture can be modified to allow increased precision as the DWT level increases so as to achieve higher accuracy.

The proposed architecture can also be extended to 2-dimensional DWT computation. This can be achieved by computing the 1-dimensional DWT along the rows and columns separately. This operation requires large amount of memory and involves extensive control circuitry.