Tutorial 3 - Discrete Walsh Transform Processor

Philip Leong

May 30, 2014

1 INTRODUCTION

The discrete Walsh transform (DWT) takes $O(N^2)$ addition/subtraction operations to execute. This can be reduced to $O(N \log_2 N)$ using the fast Walsh transform (FWT). A similar algorithm is used in the fast Fourier transform. The explanation of the DWT below follows that of Beauchamp [1].

1.1 THEORY

Will be explained here.

2 LABORATORY QUESTIONS

1. Fast Walsh transform processor (50%). Make a combinatorial, parallel implementation of an $N = 64$ FWT processor for the Altera Cyclone V 5CSEMA5 FPGA used in the DE1-SoC board. Your inputs should be 16-bit integers in two's complement form, and your output represented as a two's complement fraction with sufficiently large wordlength that overflow cannot occur. Create a set of random test vectors and verify that your design is correct. Assuming that I/O bandwidth is sufficient to supply the processor with the input vector in parallel at the maximum speed of your FWT processor, what is the maximum throughput in terms of bytes/sec? How does performance compare with your DWT processor?

2. Pipelined FWT processor (50%). Modify your FWT processor so that it is pipelined. What is its maximum throughput in terms of bytes/sec?

3. Comparison (bonus 20%). Integrate the FWT processor with the ARM Cortex A9 processor. Measure the performance of a software implementation and compare it with yours. Identify the sources of bottlenecks in your design.

REFERENCES

[1] K.G. Beauchamp. *Walsh functions and their applications.* Academic Press, 1975.