ECE 734 FALL 2000

VLSI ARRAY STRUCTURES FOR DIGITAL SIGNAL PROCESSING

FINAL PROJECT REPORT

ONE & TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

IMPLEMENTATIONS

PART I : 2-DIMENSIONAL DCT PERFORMANCE SURVEY

PART II : 1-DIMENSIONAL DCT / IDCT HARDWARE DESIGN

VIJAY S. SRINIVASAN
ELECTRICAL AND COMPUTER ENGINEERING
UNIVERSITY OF WISCONSIN-MADISON

Introduction : Motivation and Background

The Discrete Cosine Transform has long been the basic transform coding method for the JPEG and MPEG standards. It helps separate the image into parts (or spectral sub-bands) of differing importance - with respect to the spatial quality of the image. In that respect it is similar to the Discrete Fourier Transform since it transforms a signal or image from the spatial domain to the frequency domain. However one primary advantage of the DCT over the DFT is that the former involves only real multiplications, which reduces the total number of required multiplications, unlike the latter. Another advantage lies in the fact that for most images much of the signal energy lies at low frequencies, and are often small - small enough to be neglected with little visible distortion. The DCT does a better job of concentrating energy into lower order coefficients than does the DFT for image data. This characteristic of the DCT, referred to as energy compaction efficiency, along with other advantages resulted in the JPEG and MPEG standards adopting the DCT as a standard for image compression.

The 2D DCT formula can be implemented in hardware or software, and numerous fast implementations exist today based on fast algorithms developed for computing the 1D DCT.

Two-Dimensional Discrete Cosine Transform Implementation Survey : Background

The first part of this project, the Survey, compiles together a list of recent DCT implementations in hardware and software. Implementations have been divided into 4 categories : ASIC based, FPGA based, DSP based and General Purpose Microprocessor based implementations. The first two are hardware and the latter two are software implementations. Performance figures are given for each implementation on a maximum of four parameters : speed, area, power and precision. Rankings based on the figures are estimated for each implementation inside its own category, specifically for example ASIC based designs are not ranked against FPGA based designs. The objective is to gain an insight into 2D DCT implementation platforms and styles, and primarily to get an idea of state-of-art performance figures available today.

One-Dimensional Discrete Cosine Transform Design & Implementation : Background

The second part of the project, the Implementation, involves the schematic based design of a 1D DCT straight from the formula (no fast algorithm) using Altera's MaxplusII FPGA design tools. The speed of computation and area usage is compared with two other IEEE documented 1D DCT implementations on Altera devices. Design files - schematics and simulation traces - have been included at the end of the report. The objective is to come up with a working 1D DCT implementation, not fast or optimal in any way (in fact slow and big), but correct and able to demonstrate why faster and smaller 1D DCT implementation is a big issue.

PART I

2-DIMENSIONAL DCT IMPLEMENTATION PERFORMANCE : A SURVEY

The survey compiled focuses primarily on the performance figures of the various implementations studied, and a brief description of the architecture of each implementation indicates methods followed for the implementation. An effort has been made to explain architectures without focusing on the details, and some frequently occurring terms have been explained below.

Brief explanations of frequently occurring terms

1D DCT usage in computing the 2D Discrete Cosine Transform : The formula for the 2D DCT is separable, which means that it can be broken into two sequential 1D DCT operations, one along the row vector and the second along the column vector of the preceding row vector results. Called the Row-Column decomposition method, this is the most common method deployed for computing the 2D DCT, and implementations usually focus on optimizing the 1D DCT so that the Row-Column 2D DCT implementation performs better when using the optimized 1D DCT block along rows and columns. Roughly ninety percent of the survey implementations below follow this method, and in the rest the implementations are either a direct 2D DCT computation using the 1D DCT in other ways or a recursive computation where a 1D DCT is used for 2x2 2D DCT, then this is used for a 4x4 2D DCT and so on.

Distributed Arithmetic : This is a technique very commonly used where Multiply-Accumulate plays a predominant role in the operation, especially true with signal processing applications. Typically it serves to eliminate multiplications and replace them with adds, which is useful since a multiplication consumes much more time than an add. An example of the result is a case where N multiplies followed by an N-input add has been replaced by a series of N-input adds followed by a single multiply.

Bit serial architecture : Primarily used in the context of multipliers, these are architectures where a single bit bit of each input word is transmitted during each processing cycle. This reduces I/O, however an n-bit word requires n-processing cycles for transmission. There is typically a large area overhead in latches associated with serial architectures.

Asynchronous designs : These do not employ global clocking, and instead rely on local handshaking protocols to activate only those components that are necessary to perform a given computation. This reduces switching capacitance and activity in general and presents power saving potential.