ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

HDL Code Generation of Efficient DCT Architecture Using MATLAB HDL Coder

Sawan Singh

IV[th] Year Student, Department of ECE, UIET, CSJM University, Kanpur, India

ABSTRACT: This paper leads a new and easy way to develop HDL codes for an efficient DCT architecture. We replace multipliers by shift multipliers as multipliers takes a lot of power and space. A special property of DCT is used which is it is even and odd symmetric depending on the row. A simple 2x2 DCT coefficient matrix is used to develop all the other higher order matrices. For generating HDL codes we have to write a code which can be converted to fixed point not all the functions of MATLAB are currently supported by HDL coder.

KEYWORDS: MATLAB; HDL Coder; HDL Code Generation; DCT; Shift Multipliers

I. INTRODUCTION

HDL codes are extensively used to simulate and physically realize the hardware. HDL comprises of VHDL(VHSIC Hardware Description Language**)** and Verilog. Writing HDL is not easy as it contains many things like clock, input ports, output ports etc. So if we learn how to generate the HDL codes from MATLAB it will be of great help. As MATLAB is a new generation language which is easy to work upon. [1] DCT is a very important for any data compression or coding application, therefor it is very necessary to develop a low cost low power and very efficient hardware for DCT. While working with DCT we mainly uses the DCT coefficients which are multiplied with the data to get there transforms. While doing so we require multiplier in huge amount but we know that the multipliers take a lot of space and consume a lot more power. Therefore for efficient DCT in this paper I develop a method to generate the HDL code by using HDL coder of MATLAB and replacing multipliers by shift multipliers. HDL codes require a main source code and with this we also need a test bench in order to test all the possible outputs of the design. HDL Coder generates the main source code with test bench and also with the report which contains the number of devices used such as multiplier adder etc. The HDL code generation starts with converting the floating point program to the fixed point code. All the function in MATLAB is not supported by the HDL coder. The HDL coder does not support multi dimension input and output because it is physically not possible.[2]The discrete cosine transform (DCT) is one of the major components in image and video compression system. In practice DCT is the best replacement for the Karhunen Loeve Transform (KLT). The DCT is given by the following eq(1)

$$[4] \quad Tdct(u,v) = a(u)a(v) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} X(i,j) \sin\left[\frac{\pi(2i+1)u}{2N}\right] \qquad \text{eq. (1)}$$

Where

$$a(u) = a(v) = \sqrt{\frac{1}{N}} \; for \; u,v = 0 \qquad \text{eq. (2)}$$

Otherwise

$$\sqrt{\frac{2}{N}}$$

First the 1D DCT of the rows are calculated and then the 1D DCT of the columns are calculated. The 1D DCT coefficients for the rows and columns can be calculated by separating the above equation in two parts.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

II. RELATED WORK

In [3] authors develop the same technique of shifting the bits left and adding to get any multiplication result. In [3] author develops a hardware using registers, multiplier and logic gates. Connecting various component will not result in a small and efficient design therefore in this paper we develop HDL codes so that a simple hardware can be generated which result in low power and space.

In [4] author has proposed a pipelined architecture for the DCT. In this the architecture takes 8 input and provide a single DCT output. The problem with this design is that it first computes the 1D DCT and then again perform the 1D DCT to the intermediate result. The intermediate result is stored in the RAM therefore it requires an extra hardware.

In [5] author first convert the 2D DCT into 1D case and then utilize the difference time and frequency index partition methods to reduce the computational complexity. Furthermore the common factors in the decomposition matrix during the algorithm derivation are factored out in order to save the number of multipliers, but some numbers of multiplexers are still used which make this design less area and power efficient.

In [6] author used the concept of distributed arithmetic which is a bit level rearrangement of a multiply accumulate to hide the multiplication. Its powerfully reduce the size of the design which is well suited to the FPGA design. This concept can be extended to the complex multiplication. This concept uses a look up table and accumulators instead of multipliers.

In [7] author uses a zigzag and alternate scan conversion circuits. Hardware cost and performance of this design are mainly affected by the 2D DCT. Author uses row-column decomposition technique. But this design required very precise timing schedules.

In [8] author uses the coefficient distributed arithmetic (CoDA) scheme as opposed to prevalent data distributed arithmetic(DDA) schemes to achieve low power consumption. The author uses the adder arrays to compute the different operations.

In [9] the author uses the CORDIC (coordinate rotation digital computer) algorithm. The proposed algorithm is based on an indirect approach for computing the DCT so that the vector rotations are completely separated from the other operations and placed at the end of the DCT unit.

III. PROPOSED ALGORITHM

A. *Design Considerations:*

Generally multiplier consumes large space and power in a digital circuit. To overcome this problem we use shifting number logically in order to multiply them with $2^n$where n is the number of bit shift towards left. DCT coefficient matrix has a special property. [3]Given below is a 4x4 matrix here odd rows have the same absolute value.

$$\begin{bmatrix} d_{16} & d_{16} & d_{16} & d_{16} \\ d_8 & d_{24} & -d_{24} & -d_8 \\ d_{16} & -d_{16} & -d_{16} & d_{16} \\ d_{24} & -d_8 & d_8 & -d_{24} \end{bmatrix}$$

If we consider row 0 which is even than the first two elements are same as the second value, but in the second row $d_8 = -d_8$and $d_{24} = -d_{24}$ so we can see that the even rows are even symmetric and the odd are odd symmetric. So we can develop a higher order DCT coefficient matrix using the lower order matrix.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

B. *Description of the Proposed Algorithm:*

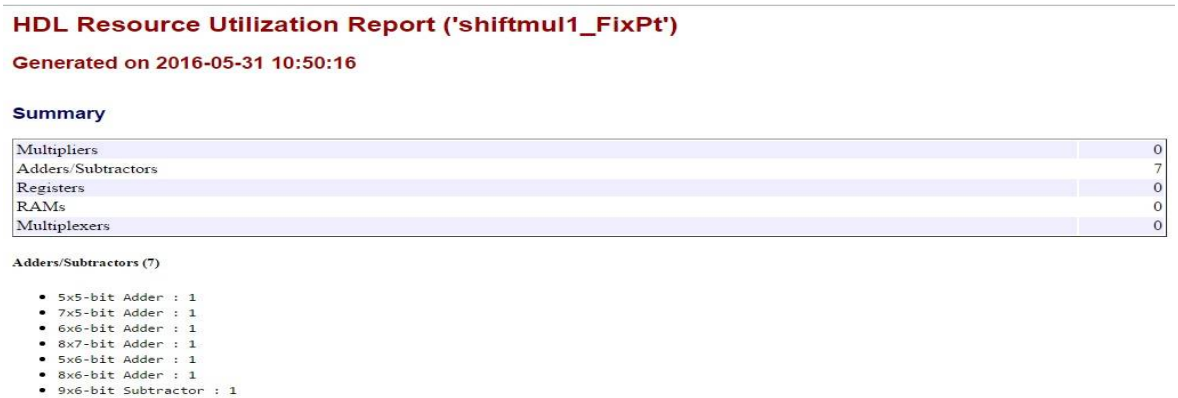
The inputs are given to a function which 1st check the order of the DCT and after that it shifts the bits to the left using MATLAB function *bitsll()*. [10] Digital signal processing can be divided into two categories fixed and floating points.This refers to format used to store and manipulate no. within devices. Fixed point DSPsare usually representing no. in bit form of different length (16 bit or more). Floating point DSP refers to the values where variable size is not fixed. MATLAB code is a floatingpoint design and HDL code is a fixed point design. We add/subtract all the coefficient values in order to the the output. We only code for the odd outputs because due to even symmetry of the even rows we can directly get there values.[11] The MATLAB/Simulink models are optimized and converted to target independent, specific and traceable Very High Speed Integrated Circuit Hardware Description Language (VHDL) code or Verilog code for FPGA programming.[12] This proposed design can also be made using Simulink and then can be converted to VHDL or Verilog code.

IV. SAMPLE CODE

A simple code is given below to generate a 4x4 DCT coefficient without using multiplier. In this x0,x1,x2,x3 are inputs and Y1 ,Y3 are the odd outputs. We know the even rows have the same values so Y0, Y4 are same as the other half of the coefficient matrix.

```
function [Y1,Y3] = shiftmul1(x0,x1,x2,x3)
data1=x0-x3;
data2=x1-x2;
  k2 = double(2);
  k6 = double(6);
  k5 = double(5);
  k4 = double(4);
  k1=  double(1);
  d1 = int16(data1);
  d2 = int16(data2);
  Bn1=bitsll(d1,k6)+bitsll(d1,k4)+bitsll(d1,k2);
  Bn2=bitsll(d2,k5)+bitsll(d2,k1)+d2;
  Y1=Bn1+Bn2;
  Bn12=bitsll(d1,k5)+bitsll(d1,k1)+d1;
  Bn11=bitsll(d2,k6)+bitsll(d2,k4)+bitsll(d2,k2);
  Y3=Bn12-Bn11;
 End
```

V. SIMULATION RESULTS


HDL Resource Utilization Report ('shiftmul1_FixPt')

Generated on 2016-05-31 10:50:16

Summary

Multipliers	0
Adders/Subtractors	7
Registers	0
RAMs	0
Multiplexers	0

Adders/Subtractors (7)

- 5x5-bit Adder : 1
- 7x5-bit Adder : 1
- 6x6-bit Adder : 1
- 8x7-bit Adder : 1
- 5x6-bit Adder : 1
- 8x6-bit Adder : 1
- 9x6-bit Subtractor : 1

Fig. 1. Utilization report for 4x4 DCT shift multiplier.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

This is the report for the 4x4 DCT shift multiplier we can see that there is no multiplier used in this design. There is Adder and subtractors. Similarly given below is the report for 8x8 DCT which again confirms that no multipliers are used in this design.

HDL Resource Utilization Report ('shiftmul8_FixPt')

Generated on 2016-06-01 12:00:44

Summary

Multipliers	0
Adders/Subtractors	18
Registers	0
RAMs	0
Multiplexers	0

Adders/Subtractors (18)

- 2x2-bit Adder : 2
- 6x2-bit Adder : 1
- 9x3-bit Adder : 1
- 12x3-bit Adder : 1
- 3x7-bit Adder : 1
- 7x10-bit Adder : 1
- 10x13-bit Adder : 1
- 13x15-bit Adder : 1
- 4x8-bit Adder : 1
- 8x11-bit Adder : 1
- 14x16-bit Adder : 1
- 5x3-bit Subtractor : 1
- 6x3-bit Subtractor : 1
- 12x2-bit Subtractor : 1
- 2-bit Subtractor : 1
- 3-bit Subtractor : 1
- 11x14-bit Subtractor : 1

Fig. 2. Utilization report for 8x8 DCT shift multiplier.

VI. CONCLUSION AND FUTURE WORK

This simulation results show that this HDL code results in efficient DCT design without using a single multiplier. Every design can be implemented in MATLAB and then can be converted into HDL code to relies it as a hardware. This may leads to a revolution which results in a very efficient and small world of electronics. Every design should be amended such that it consumes low power and are very small in space.

REFERENCES

1. Madhukar Budagavi, Arild Fuldseth, Gisle Bjontegaard, Vivienne Sze, Mangesh Sadafale, 'Core Transform Design in The High Efficiency Video Coding(HEVC)' in IEEE Journal Of Selected Topics In Signal Processing, Vol.3, 2013.
2. Vaithiyanathan Dhandapani, Seshasayan Ramachandran, 'Area and power efficient DCT architecture for image compression' in EURASIP Journal On Advances In Signal Processing, 2014.
3. Pramod Kumar Mehar, Sang Yoon Park, Basant Kumar Mohanty, Khoon Seong Lim, Chuohao Yeo, 'Efficient Integer DCT Architectures for HEVC' inIEEE Transactions On Circuit And Systems For Video Technology,2014.
4. Jagriti Sahu,'Design and Implementation Efficient DCT Architecture', 11thIRF International Conference, 2016.
5. Shen-Fu Hsiao, Wei-Ren Shiue,' New Hardware Efficient Algorithm and Architecture for the Computation of 2-D DCT on a Linear Systolic Array', IEEE, 1999.
6. Venkata Vinetha Kasturi, Y. Syamala,' VLSI Architecture for DCT Based on Distributed Arithmetic', International Journal of Engineering Research and Technology, 2013.
7. Kyeounsoo Kim, Jong Seog Koh,' An Area Efficient DCT Architecture for MPEG-2 Video Encoder', IEEE Transactions On Consumer, 1999.
8. S. Gosh, S. Venigalla, M.Bayoumi,'Design and Implementation of a 2D DCT Architecture Using Coefficient Distributed Arithmetic.',IEEE Computer Society Annual Symposium on VLSI, 2005.
9. Sungwook Yu, 'A Scaled DCT Architecture with the CORDIC Algorithm', IEEE transactions on Signal Processing,2002.
10. Harshit Pant, Himanshu Bourai, Gaurav Singh Rana,' Conversion of MATLAB Code in VHDL using HDL Coder and Implementation of Code on FPGA', HCTL Open International Journal of Technology Innovation and Research, 2015.
11. Yam P. Siwakoti, 'Design of FPGA Controlled Power Electronics and Drives Using MATLAB Simulink' ,ECCE Asia Downunder(ECCE Asia), 2013.
12. Gatis Valters,'Initial Version of MATLAB/SIMULINK based tool for VHDL code Generation and FPGA implementation of Elementary Generalized Unitary Rotaion', NORCHIP, 2011.

BIOGRAPHY

Sawan Singh is a 4th year student of B.Tech at ECE department UIET CSJM university Kanpur. Sawan has completed many successful project based on human and machine integration. His areas of interest are signal processing, human machine interface, gesture control , HDL code generation etc.