IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 4, Ver. II (Jul-Aug. 2014), PP 18-22
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

VLSI Realization of Area efficient FIR Filters

Neha Bhardwaj[1], Vipin Gupta[2]
[1]*Master of Technology (Scholar), Suresh Gyan Vihar University, Jaipur, India*
[2] *Assistant professor, Suresh Gyan Vihar University, Jaipur, India*

Abstract: *Arithmetic circuit Multiplication happens oftentimes in finite impulse response (FIR) filters, quick Fourier transforms, distinct trigonometric function transforms, convolution, and to avoid wasting vital temporal order consumption of a VLSI style. Here, we have a tendency to propose planning of FIR filter exploitation high speed & space economical multiplier factor adopting the new implementing approach exploitation changed radix-4 booth multiplier factor. The multiplier factor is intended by a changed Booth formula that is controlled by a detection unit exploitation associate degree logic gate. The changed booth formula can scale back the amount of partial merchandise generated by an element of two.*

Keywords: *DSP, FIR Filter Design, Booth algorithm, Radix no. system, Xilinx tools, area, delay.*

I. Introduction

Since the speculation of digital signal process (DSP) is developed and applied to the engineering science world, digital filtering continually plays a really vital role. Digital filtering techniques is employed to suppress noise, enhance signal in hand-picked frequency ranges, constrain information measure, take away or attenuate specific frequencies and different special operations. Digital filters square measure classified into finite impulse response (FIR) and infinite impulse response (IIR) filters. FIR digital filters will have specifically linear section response and a really regular design, and suffer less from the consequences of finite word length as compared with IIR digital filters. This thesis presents the planning an implementation of such a filter supported redundant binary variety systems. The most parts of FIR filter square measure adder, multiplier factor and delay. The carry propagation delay could be a limiting issue of the adder and multiplier factor. Supported redundant variety, adders and multipliers square measure designed in such some way that the propagation delay is reduced of the FIR filter.

The advantages of full custom square measure most circuit performance, minimum style size, and minimum high-volume cost. Finally, planning box-car FIR filter and 3-tap FIR (multiplier constant 4-bit) filter will function a basic of IC style students to figure with as a tool in their understanding of digital style. It's additionally a stepping-stone for college kids in planning different CMOS chips victimization the 90nm CMOS technology and to encourage them to form enhancements within the style.

The basic operation in digital signal process is filtering. This operation is wide utilized in several electronic devices to cancel a part of signal that's redundant or damages the signal. The digital filter is delineate by distinction equation in time domain and by transfer operate in frequency domain. There square measure 2 basic sorts of digital filters: Finite impulse response (FIR) filters and infinite impulse response (IIR) filters. The each sorts of filters have some blessings and drawbacks that square measure summarized in Table one.

Type of filter	Advantages	Disadvantages
FIR	Linear phase behavioral Stability guaranteed Low quantization noise Simple implementation	High order of filter
IIR	Low order of filter	Stability is not guaranteed Complicated implementation Limit cycles

Table 1: Advantages and disadvantages of digital filters

The main disadvantage of a FIR filter is high order of the filter as compared with IIR filter with just about an equivalent frequency response. that the size, consumption of power and computing time of a FIR filter square measure on top of IIR filter. The one in all the solutions of this drawback is use interpolated FIR (IFIR) filter that considerably reduces order of the filter. Structure of IFIR filter is delineate below. Additional details

regarding options of FIR and IIR filters is found in [3].

II. Motivation and goals

Area, delay (performance) and power square measure the 3 vital style constraints for planning an embedded period of time digital signal process systems. The real m constraint is obligatory primarily by concerns of value. Space economical implementation ends up in a smaller die size and therefore becomes additional value effective. It additionally allows group action additional practicality on one chip. The performance needs of a system square measure driven by its processing wants. For DSP systems, outturn is that the primary performance criterion. The performance constraint is therefore passionate about the speed at that the input signals square measure sampled and on the complexness of process to be performed. Low power dissipation could be a key demand for transportable, battery operated systems because it extends battery life. Low power dissipation additionally helps cut back the packaging value (plastic rather than ceramic), eliminate / cut back cooling (heat sinks) overhead and increase the responsibleness of the device.

For the need of high-speed and low-power applications, the event and implementation of high-speed FIR digital filters want each augmented similarity and reduced complexness so as to fulfill each rate and power dissipation goals. During this thesis, FIR filter is meant supported RNS to realize high speed operation. Bottom-up style flow is employed for optimum circuit performance, minimum style size, and minimum high-volume cost.

III. VLSI Design Flow

3.1 Introduction on Bottom-up and Top-down Design Flow

The styleer sometimes follows some design phases to form his project. At the start the designer needs to specify the practicality of the system. Basic blocks of the hardware square measure known and their interfaces, composed of information and management signals, are fixed. Today, there square measure 2 principal ways that to style a VLSI circuit with ancient tools that are developed in these last years. The styleer will opt for at discretion AN approach bottom-up or a top-down flow however generally the selection is forced in consequence of explicit design needs logic gate structure. Top-down could be a method of unvarying refinements. The designer starts with a prime read of the system and decomposes single blocks into smaller ones. Bottom-up flow starts with low-level building blocks and interconnects them to larger ones. In reality, these 2 techniques don't seem to be terribly incompatible and, for example, the designer can even prefer to use explicit successful cells and to try to not bit their structure at intervals a top-down approach [10]. The approach bottom-up is desirable in digital style if the designer wishes to plain a specific cell achieving carrying out with transistors full-custom designed so he needs to duplicate this structure in his project.

3.2 Bottom-Up Design Flow

The bottom-up style flow is given in Fig one.3.1. The bottom-up style flow starts with a collection of style specifications. The “specs” usually describe the expected practicality of the designed circuit additionally as different properties like delay times, area, etc. To fulfill the assorted style specifications sure style trade offs (area verses delay) square measure needed [10].

A. Schematic Capture

A Schematic Editor is employed for capturing (i.e. describing) the transistor-level style. The Schematic Editors give straightforward, intuitive means that to draw, to position and to attach individual parts that form up the planning. The ensuing diagram should accurately describe the most electrical properties of all parts and their interconnections. Additionally enclosed within the schematic square measure the provision connections (VDD and gnd), additionally as all pins for the input and output signals of the circuit. From the schematic, a netlist is generated, that is employed in later stages of the planning. The generation of an entire circuit schematic is thus the primary vital step of the transistor-level style.

B. Symbol Creation

A symbol read of the circuit is additionally needed for a few of the next simulation steps or for documentation functions. Thus, the schematic capture of the circuit topology is sometimes followed by the creation of an emblem to represent the whole circuit. The form of the icon to be used for the image might recommend the operate of the module (logic gates – AND, OR, etc.), however the default image icon could be a straightforward rectangular box with input and output pins. The image creation will facilitate the circuit styleer to form a system level design consisting of multiple hierarchy level.

IV. Multipliers

Multipliers are the key element in the most arithmetic units in microprocessors and DSPs. It is also a major source of power dissipation. To reduce the overall power budget of various digital circuit and systems, it is important to

reduce power dissipation of multipliers. Power consumption of multipliers can be reduce at various levels of design hierarchy. Using different algorithm, power consumption can be reduced. Multiplication is a process of adding an integer to itself a specified number of times. Multiplicand is added to itself a number of times as specified by Multiplier to form the result that we called product. The multiplicand is multiplied by each digit of multiplier starting with rightmost bit i.e. LSB. Intermediate results means partial products are placed on the top of the other. The final product is determined by summation of all the partial products.

Different computer arithmetic technique can be used to implement a digital multiplier. Most techniques involve computation of partial product and summation of partial product together.

4.1 Booth Multiplier

Though Wallace Tree multipliers were quicker than the normal Carry Save methodology, it conjointly was terribly irregular and therefore was difficult whereas drawing the Layouts. Slowly once multiplier factor bits gets on the far side 32-bits giant numbers of logic gates square measure needed and therefore conjointly a lot of interconnecting wires that makes chip style giant and slows down operational speed Booth multiplier factor may be employed in totally different modes like radix-2, radix-4, radix-8 etc. However we have a tendency to set to use Radix-4 Booth's algorithmic program as a result of range of Partial product is reduced to n/2. The decision to use a Radix-4 changed Booth algorithmic rule instead of Radix-2 Booth algorithm is that in Radix-4, the quantity of partial product is reduced to n/2.

4.2 Modified Booth Encoder (MBE)

Modified Booth secret writing is most frequently won't to avoid variable size partial product arrays. Before planning a MBE, the multiplier factor B must be regenerate into a Radix-4 variety by dividing them into 3 digits severally in keeping with Booth Encoder Table given later. Before convert the multiplier factor, a zero is appended into the smallest amount vital Bit (LSB) of the multiplier factor. The figure higher than shows that the multiplier factor has been divided into four partitions and hence that mean four partial merchandise are going to be generated exploitation booth multiplier factor approach instead of eight partial merchandise being generated exploitation standard multiplier factor.

Zn = -2* Bn+1 + Bn + Bn-1

Lets take an example of converting an 8-bit number into a Radix-4 number. Let the number be -36 = 1 1 0 1 1 1 0 0. Now we have to append a '0' to the LSB. Hence the new number becomes a 9-digit number, that is 1 1 0 1 1 1 0 0 0. This is now further encoded into Radix-4 numbers according to the following given table. Starting from right we have 0*Multiplicand, -1*Multiplicand, 2*Multiplicand, -1*Multiplicand.

Bn+1	Bn	Bn-1	Zn	Partial Product	1M	2M	3M
0	0	0	0	0	1	1	0
0	0	1	1	1×Multiplicand	0	1	0
0	1	0	1	1×Multiplicand	0	1	0
0	1	1	2	2×Multiplicand	1	0	0
1	0	0	-2	-2×Multiplicand	1	0	1
1	0	1	-1	-1×Multiplicand	0	1	1
1	1	0	-1	-1×Multiplicand	0	1	1
1	1	1	0	0	1	1	0

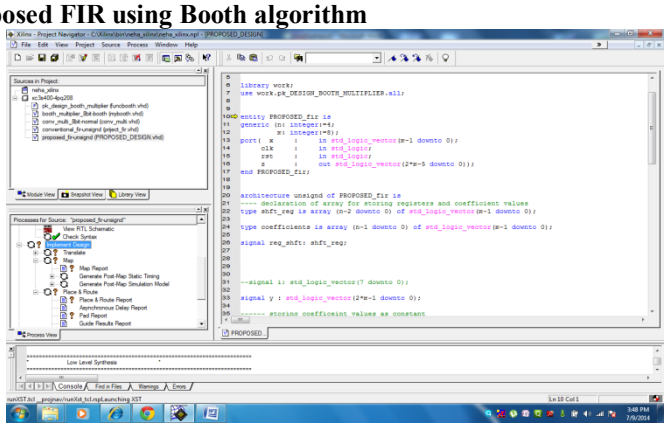
Table 4.1 Modified Booth Encoder's table to generate M, 2M, 3M control signal

Table 4.1 shows Bn+1, Bn and Bn-1 which are three bits wide binary numbers of the multiplier Bin which Bn+1 is the most significant bit (MSB) and Bn-1 is the least significant bit (LSB). Zn is representing the Radix-4 number of the 3-bit binary multiplier number. For example, if the 3-bit multiplier value is "111", so it means that multiplicand A will be 0.And it's the same for others either to multiply the multiplicand by -1, -2 and so on depending on 3 digit number. And thing to note is generated numbers are all of 9-bit. From the table 4.1, the M, 2M and 3M are the elect control signals for the partial product generator. It will determine whether the multiplicand is multiplied by 0,-1, 2 or -2. M and 2M are designed as an active low circuit which means if let's say the multiplicand should be multiplied by 1 then the M select signal will be set to low "0" whereas If the multiplicand should be multiplied by 2 then the 2M select signal will be set to low "0". The 3M is representing

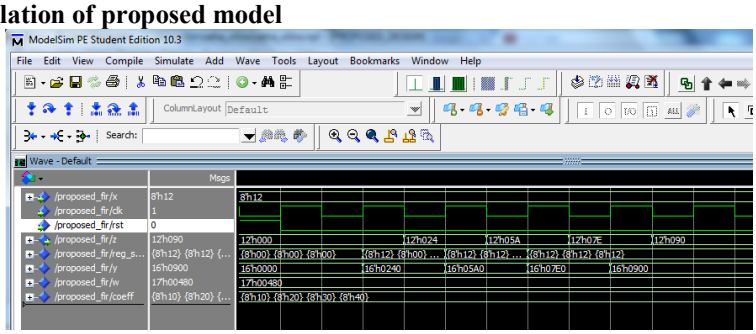
the sign bit control signal and its active high circuit which means if the multiplicand should be multiplied by -1 or -2, then the sign, 3M will be set to high "1".

V. Result And Analysis

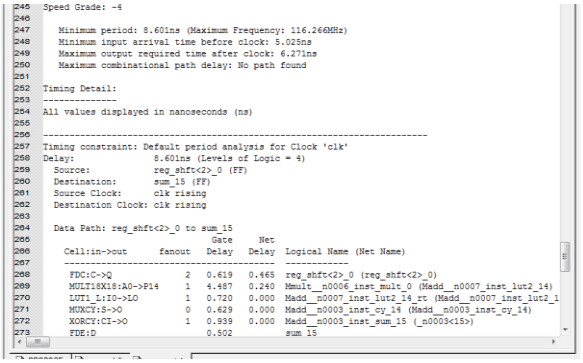
5.1 HDL Design of Proposed FIR using Booth algorithm



5.2 Functional simulation of proposed model



5.3 Timing Report of proposed Model



```
Speed Grade: -4

   Minimum period: 8.601ns (Maximum Frequency: 116.266MHz)
   Minimum input arrival time before clock: 5.025ns
   Maximum output required time after clock: 6.271ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

--------------------------------------------------------------------------------
Timing constraint: Default period analysis for Clock 'clk'
Delay:               8.601ns (Levels of Logic = 4)
  Source:            reg_shft<2>_0 (FF)
  Destination:       sum_15 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: reg_shft<2>_0 to sum_15
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDC:C->Q             2    0.619   0.465  reg_shft<2>_0 (reg_shft<2>_0)
     MULT18X18:A0->P14    1    4.487   0.240  Mmult__n0006_inst_mult_0 (Madd__n0007_inst_lut2_14)
     LUT1_1:I0->LO        1    0.720   0.000  Madd__n0007_inst_lut2_14_rt (Madd__n0007_inst_lut2_1
     MUXCY:S->O           0    0.629   0.000  Madd__n0003_inst_cy_14 (Madd__n0003_inst_cy_14)
     XORCY:CI->O          1    0.939   0.000  Madd__n0003_inst_sum_15 (_n0003<15>)
     FDE:D                     0.502          sum_15
```

VI. Conclusion And Future Work

In this thesis, I have proposed an algorithm based on Booth's Encoded Pattern to obtain Non-adjacent form with minimal hamming weight by recoding the integer key in Signed-Digit Representation to compress the on chip area. For utilization this algorithm we have introduce a basic phenomena of FIR filter with specify coefficient. I have used a booth multiplier in FIR filter to implement multiplication of coefficient and input signal.In the table 5.1 I have verify the area resizing between conventional and booth multiplication process and up to 75% area has been reduced over conventional method. With the reference of this analysis I have design area efficient FIR filter basic implementation and functional simulation with RTL Description was shown in chapter 5.

6.1 FUTURE WORK: Multipliers play a vital role in DSP and other digital calculation. Here we proposed a application of FIR filter to introduce a new booth multiplier. In proposed design a fix no. of coefficient was used and given a input sequence of 1 byte. For the brief description refer chapter no 5 and chapter 6. This work can be extend and some unified algorithm can be implement for achieve multiple no of coefficient to create multi order FIR application using modified booth algorithm. Timing performance and area density can be measure for

various order FIR filter in same logic.To obtain high precise value or performance modified booth algorithm can be implement on radix-8 or radix-16 platform and new method of 2's complement (achieving negative no.) can be merge to achieve low area.RTL design can be implementing on physical tool on 45 nm library whereas real data analysis of power consumption can be obtain.

Refrences

[1]. J.Douša, VHDL Language, Textbook CTU FEE 2003, in Czech
[2]. A.Pluha_ek, Computer Logic Design, Textbook CTU FEE 2003, in Czech
[3]. L.Ru_kay, Implementation of the IIR filter (Biquadratic Section) on the FPGA. Diploma
[4]. Thesis, CTU FEE Department of Circuit Theory, February 2005, in Czech
[5]. Richard S. Juszkiewicz, An Analysis of Interpolated Finite Impulse Response Filters
[6]. Their Improvements, IEEE Signal Processing Magazine, November 2005
[7]. Arsén Ivanóv, Case Study of Recent Improvements in Interpolated Finite Impulse
[8]. Response (IFIR) filter Design Methods, IEEE Signal Processing Magazine, April 2007
[9]. Richard Lyons, Interpolated FIR filters. Global DSP Magazine, June 2003
[10]. L.Ru_kay, Z05-02: FPGA Implementation of the MAC unit, Research report, FEE CTU