VLSI IMPLEMENTATION OF LOW POWER 8 BIT DSP PROCESSOR

Kaustav Dash[1], Dr. P.R. Tripathy[2]

[1](Electronics & Communication Engineering, Gandhi Engineering College, Bhubaneswar)
[2](Electronics & Communication Engineering, Gandhi Engineering College, Bhubaneswar)

Abstract- Digital signal processing has become increasingly popular in multimedia systems, largely thanks to the advent of general digital signal processor (DSP) chips. For portable and home entertainment audio systems, however the general DSP chip based solution might consume too much power. Then a low-cost and low power solution for this purpose should be sought. There exist two possible approaches that can reduce the chip areaand power consumption. One is the simplification of the multiplier structure, and the other techniques include low power ALU, multiplier, adder, memory, supply voltage, etc. In This project we present a low-power digital signal processor that exploits the low power multiplier design. Power dissipation of integrated circuits is a major concern for VLSI circuit designers. Since power optimization is a major goal of this work we used Wallace tree multiplier structure that reduces power consumption over traditional array multiplier. Wallace tree multiplier is an improved version of tree based multiplier architecture. In this project we replace the multiplier module of the DSP processor with Wallace tree multiplier. In addition, the Wallace tree structure has been modified to include 4:2 compressors that are used to add the partial products generated from the multiplier unit. As 4:2 Compressors reduces number of adders needed in the addition process, it reduces number of full adders needed. Hence this further reduces the power consumption. The DSP processor is implemented using Spartan3 FPGA from Xilinx, and it could also implemented using Application specific Integrated Circuits (ASIC) when further reduction in power consumption is required. The result shows that the proposed architecture is faster than the conventional CMOS architecture, along with reduced powerconsumption.

I. INTRODUCTION

Digital signal processor (**DSP**) is a specialized \

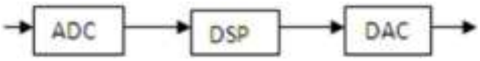


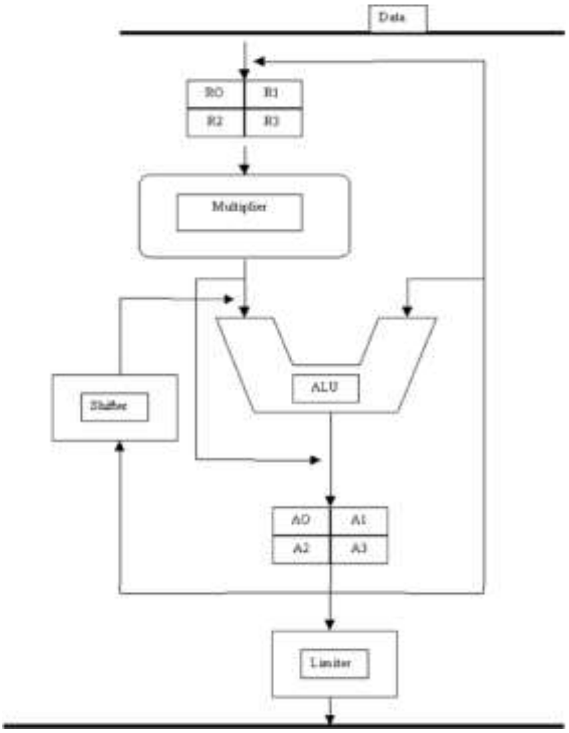
Fig1.1. *A typical digital signal processor*

Digital signal processing algorithms typically require a large number of mathematical operations to beperformed quickly and repeatedly on a set of data. Signals (perhaps from audio or video sensors) are constantly converted from analog to digital, manipulated digitally, and then converted backto

analog form. Many DSP applications have constraints on latency; that is, for the system to work, the DSP operation must be completed within fixed time, and deferred (or batch) processing is not viable. Most general-purpose microprocessors and operating systems can execute DSP algorithms successfully, but are not suitable for use in portable devices such as mobile phones and PDAs because of power supply and space constraints. A specialized digital signal processor, however, will tend to provide a lower-cost solution, with better performance, lower latency, and no requirements for specialized cooling or largebatteries

The architecture of a digital signal processor is optimized specifically for digital signal processing. Most also support some of the features as an applications processor or microcontroller, since signal processing is rarely the only task of a system. Some useful features for optimizing DSP algorithms are outlined below.

BLOCK DIAGRAM OF 8-BIT DSP PROCESSORDATAPATH:



II. MOTIVATION

Comparison between DSP &GPP:-

With the advent of Digital Signal Processors one can easily classify processors as follows. A General Purpose Processor (GPP) is very efficient in data manipulation. A Digital Signal Processor (DSP) is designed specially to efficiently perform mathematical calculation. DSPs have their architectures optimized so that all the operations it undertakes are finished within as low clock cycles as possible. A GPP cannot perform well for DSP applications. DSP based applications demand a lot of number crunching, very high data bandwidth, real time constraints, attention to subtle numeric effects (in fixed- point implementations), and specialized Peripherals / interfaces.A Digital Signal Processor's average speed performance is far better than a corresponding GPP clocked by the same clock. Thus any GPP architecture would want to move towards DSP architecture. Ageneral trend shows that GPP Architecture approaches to DSP: Initially we had Baseline GPPs (moderate performance, no DSP features) like the 8085,8086,80386 etc. These low/moderate performance GPPs with no DSP features perform poorly on DSP tasks because they have poor multiplication throughput, limited memory bandwidth, loop overhead, address generation overhead. Also, being fixed- point processors they lack hardware to support fast overflow protection, convergent rounding, etc. Then there were high-performance GPPs with few/no DSP- oriented features like Pentium (P54C).These processors performed well on DSP tasks as they had high clockrates (200+ MHz; 2- 5 X those of typical DSPs), multiplication and fast arithmetic operations, Good memory bandwidth, Loop overhead reduced via branch prediction, and floating point arithmetic block included. However, dynamic features complicate optimization of DSP code and make real- time developmentdifficult.

9

Then there were GPPs with major DSP- oriented features. For example, Intel MMX Pentium (P55C). These processors achieve outstanding DSP performance bycombining the features of "conventional" high performance GPPs with new SIMD (single instruction multiple data path) capabilities by partition existing data path (or adding a new partitioned datapath), Multiple operations/cycle on small data types (e.g., 4 multiplies), Single-cycle operations on various fixed- Point data types. In addition there were DSP co- processors like ARM Piccolo co-processor(for ARM7 designs), which gave better performance on DSP tasks as compared to their corresponding processor systems without a coprocessor. Thus it is observed a Digital Signal Processor is very important block not only for DSP applications but also for General Purpose Processors.

Powerconsumption

Microprocessors use multipliers within their arithmetic logic units, and digital signal processing systems require multipliers to implement DSP algorithms such as convolution and filtering. In most systems, the multiplier lies directly within the critical path due to which, the demand for high-speed multipliers is continuously increasing. However, due to portability and reliability issues, the power consumption of the multipliers has become equally important. In this paper,a new design technique for column compression (CC) multipliers is presented. Constraints for column compression with full and half adders are analyzed and, under these constraints, considerable flexibility for implementation of the CC multiplier, including the allocation of adders, and choosing the length of the final fast adder, is exploited. We show that architectures obtained from this new design technique are more area efficient and have shorter interconnections than the classical Dadda CC for the design of twos complement multipliers.

III. METHODOLOGIES

ExistingMethods

Programming DSP applications in a high-level language such as C is becoming more prevalent as applications become increasingly more complex. Current DSP compilers, however, are generally unable to exploit the DSP-specific features of a processor to produce good codes for most DSP applications. To explore the challenges and gain an understanding of generating code that is as good as handwritten assembly code, an optimizing C compiler for Texas Instruments' TMS320C25 DSP processor is developed. A pseudo C25 instruction set and GCC configuration files are also created for the C25 in this study. A number of benchmarkprogramsarecollectedandusedtoevaluate the quality of compiler-generated codes. Finally, this thesis shows that a modified GNU C compiler combined with a post-optimizer is able to utilize the specialfeatures of the TMS320C25 and to generate high-performance codes. The empirical results indicate that codesgenerated by the C25 optimizing compiler execute, on average, 1.9 times faster than the TI compiler-generated codes. In addition, for most of the benchmark programs, the performance of codes generated by the C25 optimizing compiler come, on average, within a factor of two of that of the hand-written assemblycodes.

Low power parallel array multiplier is proposed for both unsigned and two's complement signed multiplication. Modified Baugh-Wooley multiplier is further modified and if input numbers are not in two's complement form, proposed method makes the calculation of two's complement of the number redundant, thus reducing delay. Also power consumption has been found to be less than that of modified Baugh- Woolen multiplier. Multipliers are one of the most important units in microprocessors and DSPs and also a major source of power dissipation. Reducing the power dissipation of multipliers is key to satisfying the overall power budget of various digital circuits and systems. Power consumed by multipliers can be lowered at various levels of the design hierarchy, from algorithms to architectures to circuits, and devices. In this paper, we focus on power reduction for 10
both unsigned and signed multipliers. For Signed multiplier, the modified Baugh- Wooley algorithm is extended to obtain a power efficient multiplier. Inputs are the inverted bits of two's complement representation. The basic process of binary array multiplication involves the AND operation of multiplicand and multiplier bits and subsequentaddition.

IV. PROPOSED TECHNIQUE

Digital audio signal processing has become increasingly popular in multimedia systems, largely thanks to the advent of general digital signal processor (DSP) chips and high-precision oversampling A/D and D/A converters. For portable and home entertainment audio systems, however; the general-DSP-chip-based solution might be too costly and consume too much power: Then a low-cost and low power solution for this purpose should be sought. By carefully examining digital audio processing applications, we believe that there exist two possible approaches that can reduce the chip area and power consumption: one is the simplification of the audio processor structure, and the other is to reduce the chip supply voltage. This paper presents a low-powerand low-cost two-channel (stereo) digital audio processorthat exploits these twoapproaches.

To achieve the low-power and low-cost purposes, several considerations related to audio processing algorithms and the processor architecture has been made in the early stage of the chip design. First, because multipliers normally occupy large chip areas and, for CMOS, consume considerable power when they are clocked, multiplier-free algorithms are preferred. Second, an optimal memory address technique should be adopted in order to minimize the system hardware. Third, as compared with a general DSP, the function of the audio processor should be designed more specific, and in this design it is optimized for performing the convolution operation.

In general, the design follows the principle that, based on the characteristics of convolutions, makes the system as simple as possible. The power issue is another major concern in our design. It is well known that decreasing the power supply can considerablyreduce the system power dissipation, and it has become a popular power-saving technique. However, the system will

generally sacrifice its performance when supply voltage is reduced. To reach a good balance between power and performance, intensive simulations are made on difference circuit design unit to guarantee acceptable system performance.

ArrayMultiplier

Array multiplier is an efficient layout of a combinational multiplier. Multiplication of two binary number can be obtained with one micro-operation by using a combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array.In array multiplier, consider two binary numbers A and B, of m and n bits. There are mn summands that are produced in parallel by a set of mn AND gates. n x n multiplier requires n(n-2) full adders, n half-adders and n2 AND gates. Also, in array multiplier worst case delay would be (2n+1) td. Array Multiplier gives more power consumption as well as optimum number of components required, but delay for this multiplier is larger. It also requires larger number of gates because of which area is also increased; due to this array multiplier is less economical. Thus, it is a fast multiplier but hardware complexity ishigh.

11

ArrayStructure

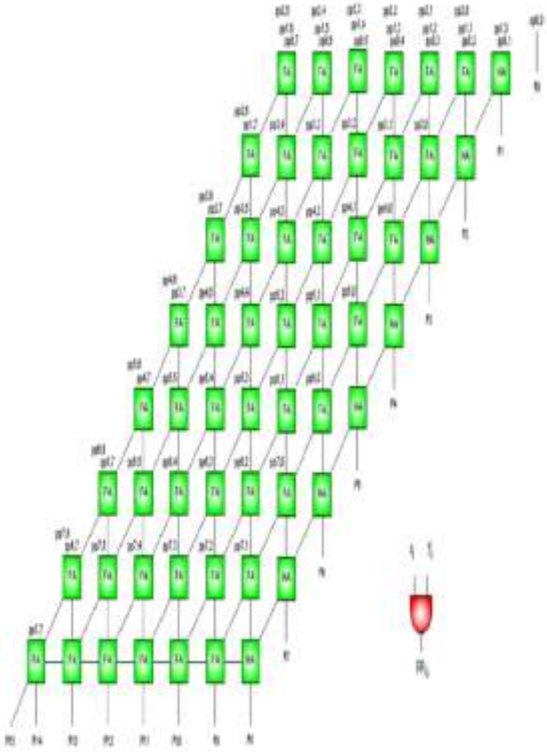


Fig: 1.6 Array Structure

supplied to the next stage of the full adder located at a one bit higher position.

ASIC

Any IC other than a general purpose IC which contain the functionality of thousands of gates is usually called an ASIC (Application Specific Integrated Circuit). ASICs are designed to fit a certain application. An ASIC is a digital or mixed-signal circuit designed to meet specifications set by a specific project.

Programming Language

Verilog is a HARDWARE DESCRIPTION LANGUAGE (HDL). A hardware description Language is a language used to describe a digital system, for example, a microprocessor or a memory or a simple flip- flop. This just means that, by using a HDL one can describe any hardware (digital) at any level.

Verilog is one of the HDL languages available in the industry for designing the Hardware. Verilog allows us to design a Digital design at Behaviour Level, Register Transfer Level (RTL), and Gate level and at switch level. Verilog allows hardware designers to express their designs with behavioural constructs, deterring the details of implementation to a later stage of design in the final design. Verilog like any other hardware description language, permits the designers to design a design in either Bottom-up or Top-down methodology.

3.1 ARRAY MULTIPLIER

Fig 1.7: Array multiplier

WALLACE TREEMULTIPLIER

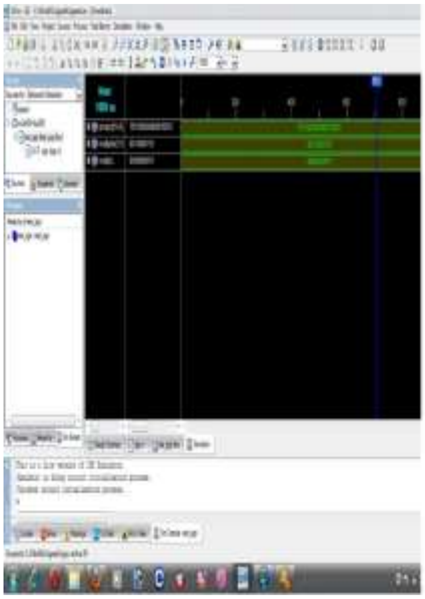
A Wallace tree is an efficient hardwires implementation of a digital circuit that multiplies two integers.

The Wallace tree has three steps:

- Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding n^2 results. Depending on position of the multiplied bits, the wires carry different weights, for example wire of bit carrying result of a_2b_3 is32.
- Reduce the number of partial products to two by layers of full and halfadders.
- Group the wires in two numbers, and add them with a conventionaladder.

A fast process for multiplication of two numbers was developed by Wallace. Using this method, a three step process is used to multiply two numbers; the bit products are formed, the bit product matrix is reduced to a two row matrixwhere sum of the row equals the sum of bit products, and the two resulting rows are summed with a fast adder to produce a final product. Three bit signals are passed to a one bit full adder (“3W”) which is called a three input Wallace tree circuit, and the output signal (sum signal) is supplied to the next stage full adder of the same bit, and the carry output signal thereof is passed to the next stage full adder of the same no of bit, and the carry output signal thereofis

SIGNEDMULTIPLIER



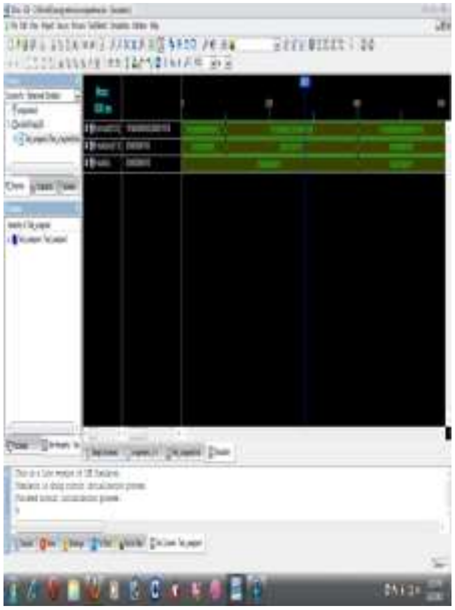


Fig 1.8: Signed multiplier *Fig 1.9: Unsigned multiplier*

MULTIPLICATION BY USING WALLACE TREE MULTIPLIERa7a6 a5 a4 a3 a2 a1b7 b6 b5 b4 b3 b2 b1 b0

P7 PP6 PP59 PP38 PP37 PP36 PP35 PP34 PP33 PP32 PP47 PP46 PP45 PP44 PP43 PP42 PP41 PP404 PP53 PP52 PP51 PP50 PP49 PP48 PP63 PP62 PP61 PP60 PP59 PP58 PP57 PP56

PP63 PP62 PP61 PP60 PP59 PP58 PP57 PP56 PP48 PP40 PP32 PP24 PP16 PP8

PP0 PP55 PP54 PP53 PP52 PP51 PP50 PP49 PP41 PP33 PP25 PP17 PP9PPPP47

PP46 PP45 PP44 PP4P42 PP34 PP26 PP18 PP10 PP2 PP39 PP38 PP37 PP36 PP35

PP24 PP19 PP11PFA comp comp comp comp comp comp comp comp comp FA

HA 4:24:2 4:2 4:2 4:2 4:2 4:2 4:2 4:2

PP31 PP30 PP29 PP28 PP20 PP12 PP4

PP23PP22 PP21 PP13 PP5

PP15 PP14 PP6

PP7

FA comp comp FA HA

4:2 4:2

PP63 S13 S12 S11 S10 S9 S8 S7 S6 S5 S4 S3 S2 S1 PP0

C13 C12 C11 C10 C9 C8 C7 C6 C5 C4 C3 C2 C1

PP31 S18 S17 S16 S15 S14 PP4

C18 C17 C16 C15 C14

```
        PP63 S13 S12 S11  S10  S9   S8   S7    S6 S5 S4    S3   S2   S1PP0
        C13   C12   C11 C10  C9C8   C7   C6   C5 C4 C3 C2CPP31  S18  S17  S16   S15   S14  PP4 C18 C17 C16HA
        FA comp comp comp comp comp FA  FA  HA HA   4:2 4:2  4:2  4:2  4:2
      ------------------------------------------------------
        S31  S30   S29   S28   S27   S26  S25   S24   S23    S22  S21  S20   S19  S1
   PP0    C31C29 C28 C27 C26 C25 C24 C23 C22 C21 C20C19
   C43 C42 C41 C40 C39 C38 C37 C36 C35 C34 C33 C32
      ------------------------------------------------------
   P15   P14 P13   P12  P11 P10P9   P8    P7    P6    P5    P4    P3 P2 P1P0
------------------------------------------------------------------------------------------------------------
```

COMPARISON OF MULTIPLIERS

Components	**Array Multiplier**	**Signed Multiplier**	**Unsigned Multiplier**	**Wallace tree Multiplier**
#IOs	32	31	32	32
# BELS	130	132	173	165
# IO Buffers	32	31	32	32
# IBUF	16	16	16	16
# OBUF	16	15	16	16
Number of Slices	74 out of 3584 2%	51 out of 3584 1%	60 out of 3584 1%	87 out of 3584 2%
Number of 4 input LUTs	129 out of 7168 1%	88 out of 7168 1%	108 out of 7168 1%	154 out of 7168 2%
Number of bonded IOBs	32outof 141 22%	31outof 141 21%	32outof 141 22%	32outof 141 22%
Gate count for design	777	653	836	957
JTAG gate count for IOBs	1,536	1,488	1,536	1,536
Peak Memory Usage	183 MB	182 MB	183 MB	184 MB

Table4.1. *Comparison of Multiplier*

V. CONCLUSION

We have successfully completed in which we are programming the Low power Wallace tree multiplier by using Verilog HDL Language some modules in our project is designed and tested by using Xilinx ISE 9.1i software.Verilog HDL coding of our project is downloaded into the FPGA or ASIC Hardware kit. Finally we got the expected output of our project.

REFERENES

[1] Wen-Yen Lin,"An Optimizing Compiler for the TMS320C25 DSPProcessor",1998.

[2] Tsung-En Andy Lee, 'Low power Digital Signal Processor Architecture For Multiple Standard wireless communications', March1998

[3] Damu Radhakrishnan,"PASS LOGIC CIRCUITS WITH REDUCED SWITCHING ACTIVITY FOR LOW POWER DSP PROCESSORS", IEEE2000.

[4]NMozaffar and N Z. Azeemi"Design and Implementation Of a SHARC Digital Signal Processor CoreInVerilogHDL",ProceedingsIEEEINMIC2003C.H.Gebotys, R.J.Gebotys "Designing for Low Power in Complex Embedded DSP Systems" ,IEEE 1999.

[5] Blair G.M. "Designing Low Power CMOS". *IEEE* Electronics and Coniniunicution Engineering Journal, 6:229-236,1994.

[6] Brzozowski I. and Kos A. “Minimization of Power Consumption in Digital Integrated Circuits by Reduction of Switching Activity”. Proceedings qf 2SIh EuronticroConference, Milan, Italy, 1999, vol. 1, pages 376-380.

[7] Guyot A. and Abou-Samara S. “Power Consumption in Digital Circuits”. Proceedings of 3 International Conference on ASIC, Beijing,1998.

[8] Pedron C. and Stauffer A. “Analysis and Synthesis of Combinational Pass Transistor Circuits”. *IEEE* Transactions on Computer-Aided Design, 7:775-786, 1988.

[9] Radhakrishnan D., Whitaker S.R., and Maki G.K. “Formal Design Procedures for Pass Transistor Switching Circuits”.*IEEE Journal of Solid-StateCircuits*, SC-2053 1-536,1985.

[10] Radhakrishnan D. “Design of CMOS Circuits”. *IEEProceedings - Circuits Devices and Systems*, 138333- 90,1991.

[11] Shams A.M. and Bayoumi M.A. “A Structured Approach for Designing Low Power Adders., 1998, pages 757-761.

[12] Erskine, C.. and S. Magar. “Architecture and Applications of a Second-Generation Digital Signal Processor,” Proceedings of IEEE International Conference on Acoustics, Speech, and SignalProcessing, USA,1985.

[13] Lin. K., G. Frantz. and R. Simar, Jr.. “The TMS320 Family orDigital Signal Processors,” Proceedings of the IEEE, USA, Volume 75, Number 9, pages 1143-1159, September1987.

[14] G. Jiangmin, G. Jiangmin, and C. Chip-Hong,"Low voltage, low power (5:2) compressor cell for fast arithmetic circuits", 2 ed. C. Chip-Hong, Ed. 2003, p.11- 4.

[15] C. Chip-Hong, G. Jiangmin, and Z. Mingyan, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," Circuits and Systems I: Regular Papers, IEEE Transactions on vol. 51, no. 10, pp.1985-1997,2004.

[16] R. Menon, R. Menon, and D. Radhakrishnan, "High performance 5:2 compressor architectures," Circuits, Devices and Systems, lEE Proceedings -, vol. 153,no. 5, 2