IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 10 (October. 2013), ||V3|| PP 01-07

Implementation of Baugh-Wooley Multiplier Based on Soft-Core Processor

Indrayani Patle, Akansha Bhargav, Prashant Wanjari,
1Lecturer, RGCER,Nagpur,
2 Lecturer , Atharv College of Engg.,Mumbai,
3 Lecturer, RGCER Nagpur

Abstract: - This Paper presents the work on implementation of Baugh-Wooley multiplier based on soft-core processor. MicroBlaze soft core is high performance embedded soft core processor developed by XILINX Company. This soft core enjoys high configurability and allows designer to make proper choice based on his own design requirements to build his own hardware platform.

Custom hardware of power optimized Baugh-Wooley signed multiplier is interface with MicroBlaze soft core processor. The major objective for using hardware for realizing Baugh-Wooley multiplier is to utilize hardware for realizing fast and efficient processing capacity.

Keywords: - *VHDL, FPGA, MICRO-BLAZE, SCP , SOC, CAD Tool, EDK.*

I. INTRODUCTION

Multipliers play an important role in today’s digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact implementation.

The common multiplication method is “add and shift” algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To achieve speed improvements Baugh-Wooley algorithm can be used.

This multiplier subsystem is commonly implemented using an embedded processor [1] combined with specific hardware.

Field programmable gate arrays (FPGAs) provide designers with the ability to quickly create hardware of circuits. Increases in FPGA configurable logic capacity and decreasing FPGA costs have enabled designers to more readily incorporate FPGAs in their designs. While FPGAs with soft processor cores provide designers with increased flexibility.

Reconfigurable logic devices, such as field programmable gate arrays (FPGAs), have been very effective to implement dedicated multiplier architectures . Over the last few years, the huge increase in FPGA features made possible the implementation of a whole system in a single device: processor, peripherals, memories and so on. Nowadays, it is feasible to implement on an FPGA an entire multiplication algorithm based on a soft-core processor (SCP), which also includes multiplier cores for hardware acceleration .

There are a several soft core processors that are commonly used in SOC applications like PowerPC [5], NIOS3 [4], MicroBlaze [2], and free or open cores that may be used without the need to acquire a license, like LEON3 [6]. The main advantage of these processors are that they are usually well tested and optimized for a specific target hardware and provide a complete set of CAD tools to make the SOC design an easier process. For example, MicroBlaze from Xilinx is well integrated with the development platform from the same foundry, which leads to highly optimized designs at the cost of being bound to a particular technology (Xilinx Spartan and Virtex FPGA families) and a concrete set of tools (Xilinx ISE and EDK [3]).

1.1 Microblaze Soft-Core Processor [1,2]

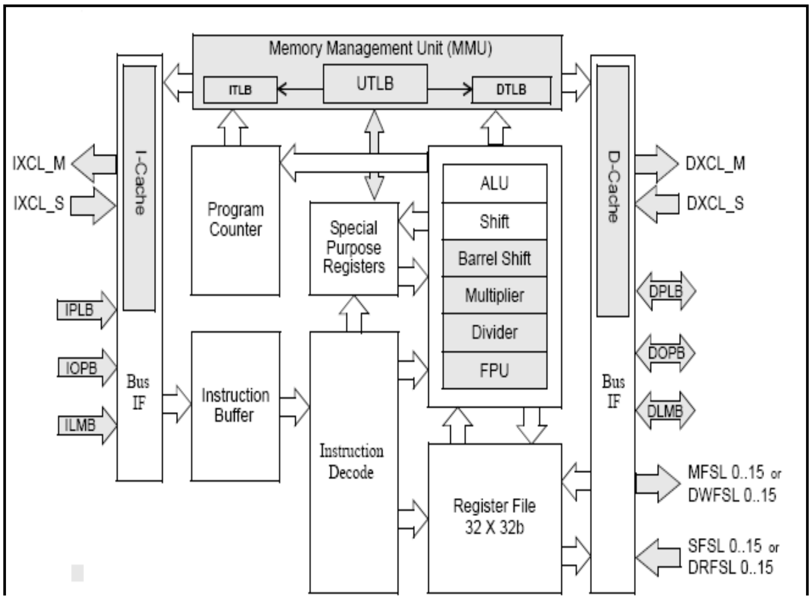


Figure 1. MicroBlaze's internal structure.

MicroBlaze soft core is highly simplified embedded processor soft core with relatively high performance developed by XILINX Company.[7] This soft core enjoys high configurability and allows designer to make proper choice based on his own design requirements to build his own hardware platform. The processor architecture includes thirty-two 32-bit general-purpose registers and an orthogonal instruction set. It features a three-stage instruction pipeline, with delayed branch capability for improved instruction throughput. As it is a SCP, the functional units incorporated into the processor architecture can be customized in order to fit as much as possible the target application. This soft core adopts RISC instruction set and Harvard architecture and has the following performance characteristics:
1) 32-bit general-purpose registers and 2 special register
2)32-bit instruction word length, 3 operands and 2 kinds of addressing modes.
3) Separated 32-bit instruction and data bus.
4) Complying with IBM OPB specification;
5) Local Memory Bus (LMB) enables direct access to on-chip block memory (BRAM), it provides high-speed instructions and data caching and features three-stage pipelined architecture;
6) Hardware debugging module (MDM) and eight input/output fast link interfaces (FSL) are available. Figure 1 shows MicroBlaze's internal structure.

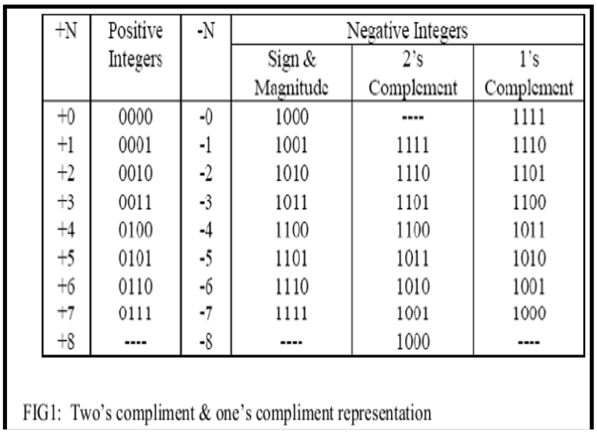
1.2 Multiplier [8]

Multiplication is a heavily used arithmetic operation that figures Multiplication is a heavily used arithmetic operation that figures prominently in signal processing and scientific applications. Multiplication is hardware intensive, and the main criteria of interest are higher speed, lower cost and lower power. The main concern in classic multiplication often realized by K cycles of shifting and adding, is to speed up the underlying multi-operand addition of partial products. A variety of multiplication algorithms and hardware designs are available.

1.3 Baugh-Wooley Multiplier [7]

2's Compliments is the most popular method in representing signed integers in Computer sciences. It is also an operation of negation(Converting positive to negative numbers or vice –versa) in computers which represent negative numbers using two's compliments. Its use is so wide today because it does not require the addition and subtraction circuitry to examine the signs of the operands to determine whether to add or subtract. Two's compliment and one's compliment representations are commonly used since arithmetic units are simpler to design. Figure 2 shows 2's compliment and one's compliment representations.
Baugh-Wooley Two's comp liment Signed numbers : Baugh-Wooley Two's compliment Signed multipliers is the best known algorithm for signed multiplication because it maximizes the regularity of the multiplier and allow all the partial products to have positive sign bits.Baugh–Wooley technique was developed to design direct multipliers for Two's compliment numbers.When multiplying two's compliment numbers directly, each of the partial products to be added is a signed numbers. Thus each partial produ ct has to be sign extended to the width of the final product in order to form a correct sum by the Carry Save Adder (CSA) tree. According to Baugh-

Wooley approach, an efficient method of adding extra entries to the bit matrix suggested to avoid having deal with the negatively weighted bits in the partial product matrix.

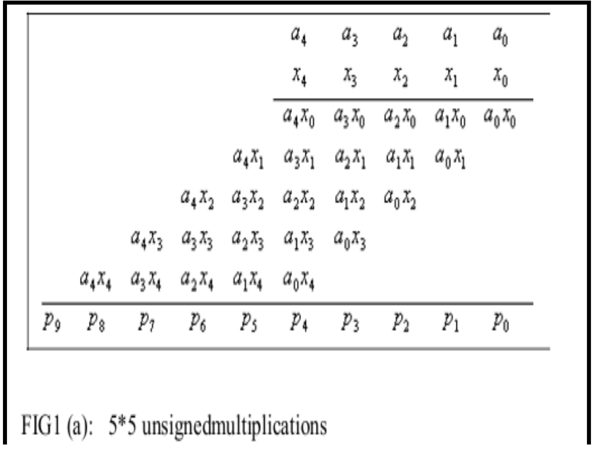


+N	Positive Integers	-N	Negative Integers		
			Sign & Magnitude	2's Complement	1's Complement
+0	0000	-0	1000	----	1111
+1	0001	-1	1001	1111	1110
+2	0010	-2	1010	1110	1101
+3	0011	-3	1011	1101	1100
+4	0100	-4	1100	1100	1011
+5	0101	-5	1101	1011	1010
+6	0110	-6	1110	1010	1001
+7	0111	-7	1111	1001	1000
+8	----	-8	----	1000	----

FIG1: Two's compliment & one's compliment representation

Figure 2. 2's compliment and 1's compliment representation

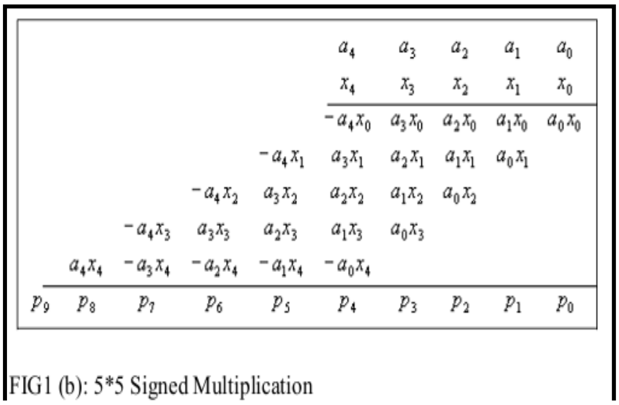
In figure 2 (a) & (b)partial product arrays of 5*5 bits Unsigned and Signed bits are shown:



$$\begin{array}{cccccccccc}
 & & & & & a_4 & a_3 & a_2 & a_1 & a_0\\
 & & & & & x_4 & x_3 & x_2 & x_1 & x_0\\
\hline
 & & & & & a_4x_0 & a_3x_0 & a_2x_0 & a_1x_0 & a_0x_0\\
 & & & & a_4x_1 & a_3x_1 & a_2x_1 & a_1x_1 & a_0x_1 & \\
 & & & a_4x_2 & a_3x_2 & a_2x_2 & a_1x_2 & a_0x_2 & & \\
 & & a_4x_3 & a_3x_3 & a_2x_3 & a_1x_3 & a_0x_3 & & & \\
 & a_4x_4 & a_3x_4 & a_2x_4 & a_1x_4 & a_0x_4 & & & & \\
\hline
p_9 & p_8 & p_7 & p_6 & p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}$$

FIG1 (a): 5*5 unsignedmultiplications

Figure 2.(a) partial product arrays of 5*5 bits Unsigned



$$\begin{array}{cccccccccc}
 & & & & & a_4 & a_3 & a_2 & a_1 & a_0\\
 & & & & & x_4 & x_3 & x_2 & x_1 & x_0\\
\hline
 & & & & & -a_4x_0 & a_3x_0 & a_2x_0 & a_1x_0 & a_0x_0\\
 & & & & -a_4x_1 & a_3x_1 & a_2x_1 & a_1x_1 & a_0x_1 & \\
 & & & -a_4x_2 & a_3x_2 & a_2x_2 & a_1x_2 & a_0x_2 & & \\
 & & -a_4x_3 & a_3x_3 & a_2x_3 & a_1x_3 & a_0x_3 & & & \\
 & a_4x_4 & -a_3x_4 & -a_2x_4 & -a_1x_4 & -a_0x_4 & & & & \\
\hline
p_9 & p_8 & p_7 & p_6 & p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}$$

FIG1 (b): 5*5 Signed Multiplication

Figure 2.(b) partial product arrays of 5*5 bits Signed bits

Figure 2 (c) shows how this algorithm works in the case of a 5x5 multiplication. The first three rows are referred to as PM (partial products with magnitude part) and generated by one NAND and three AND operations. The fourth row is called as PS (partial products with sign bit) and generated by one AND and three NAND operations with a sign bit. Consider the partial products of PM. Suppose b2= b0 in figure2 (c). Then the third row can be obtained by shifting the first row by 2 bits. Likewise, shift operation can be used to obtain a partial product of different bit level as in sign magnitude multiplication.

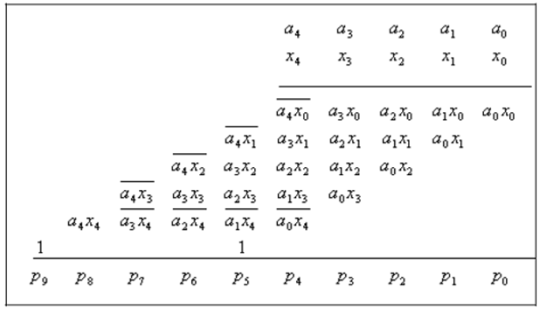

a_4 a_3 a_2 a_1 a_0
x_4 x_3 x_2 x_1 x_0
$\overline{a_4x_0}$ a_3x_0 a_2x_0 a_1x_0 a_0x_0
$\overline{a_4x_1}$ a_3x_1 a_2x_1 a_1x_1 a_0x_1
$\overline{a_4x_2}$ a_3x_2 a_2x_2 a_1x_2 a_0x_2
$\overline{a_4x_3}$ a_3x_3 a_2x_3 a_1x_3 a_0x_3
a_4x_4 a_3x_4 a_2x_4 a_1x_4 a_0x_4
1 1
p_9 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0

FIG1 (c): 5*5 Multiplication Example of Baugh-WooleyArchitecture

Figure 2(c) a 5x5 multiplication of Baugh-Wooley architecture

Baugh-Wooley schemes becomean area consuming when operands are greater than or equal to 32 bits. The rest of the paper is organised as follows. The baugh-Wooley architecture is explained in section 2. Implementation results in terms of power, area, and speed 4 bit multipliers and comparison are presented.

1.4 Baugh-Wooley Architecture

Hardware architecture for Baugh-Wooley multiplier is shown in figure 3.It follows left shift algorithm. Through mux we can select which bit will multiply. Suppose we are adding +5 and -5 in decimal we get ‘0’. Now, represent these numbers in 2’s complement form, and then we get +5 as 0101 and -5 as 1011. On adding these two numbers we get 10000. Discard carry, then the number is represented as ‘0’.

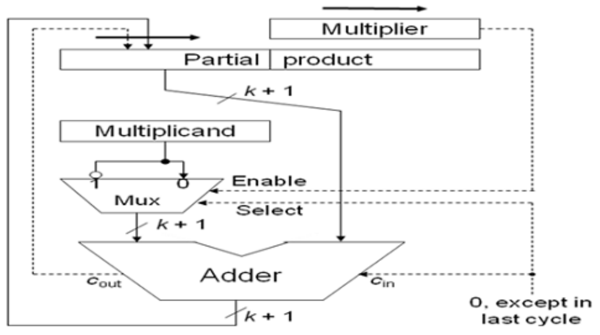


Fig 2: Signed 2’s-Complement Baugh-Wooley Hardware Multiplier

Figure 3. Hardware architecture for Baugh-Wooley multiplier

1.5 Baugh-Wooley Multiplier [7]:

Baugh-Wooley Multiplier is used for both unsigned and signed number multiplication. Signed Number operands which are represented in 2’s complemented form. Partial Products are adjusted such th at negative sign move to last step, which in turn maximize the regularity of the multiplication array. Baugh-Wooley Multiplier operates on signed operands with 2’s complement representation to make sure that the signs of all partial products are positive.

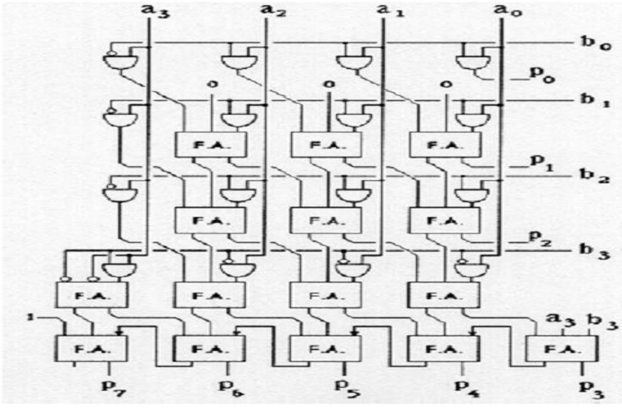


Figure4. Block diagram of a 4*4 Baugh-Wooley multiplier

Here are using fewer steps and also lesser adders. Here a0, a1, a2, a3& b0, b1, b2, b3 are the inputs. I am getting the outputs that are p0, p1... p7. As I am using pipelining resi ster in this architecture ,so it will take less time to multiply large number of 2's compliment but less than 32 bit .Above 32 bit Modified Baugh-Wooley Multiplier is used.

2.1 Implementation of Baugh-Wooley Multiplier

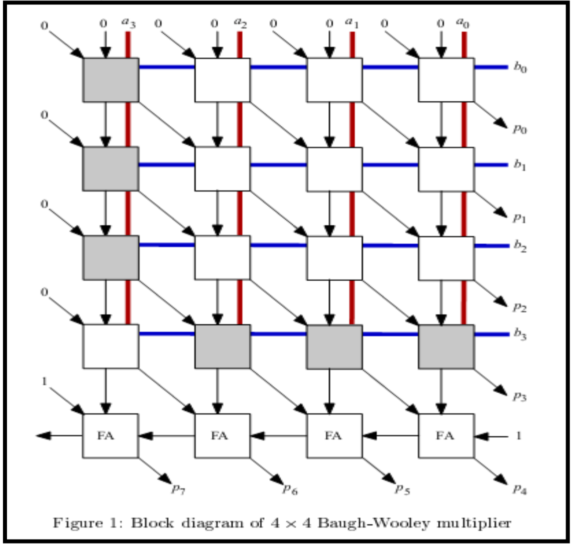
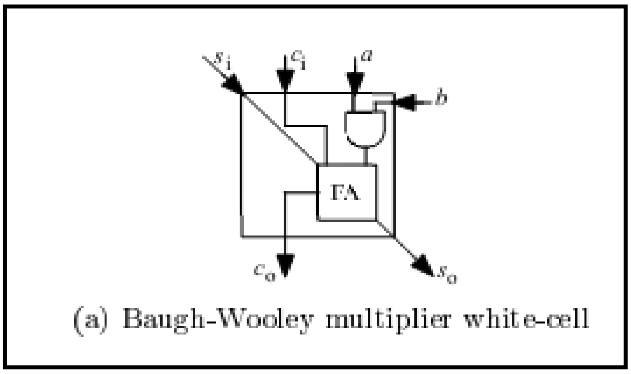


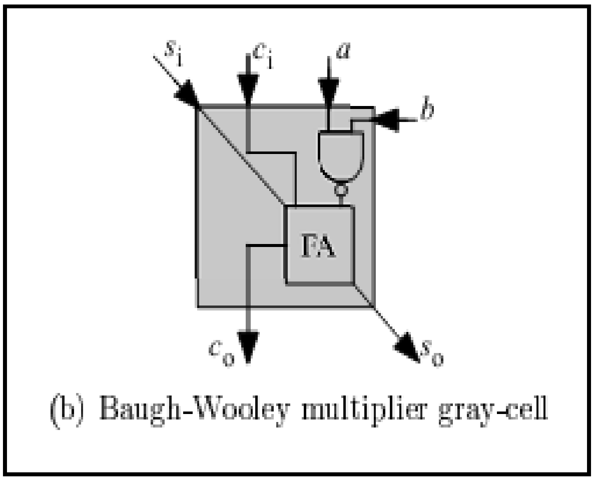
Figure 5. 4x4 Baugh-Wooley multiplier architecture

4x4 Baugh-Wooley multiplier architecture is shown in figure 5. And white and gray cell used in above architecture is shown in figure 5(A) ,figure 5(B).



(a) Baugh-Wooley multiplier white-cell

Figure 5(A)



(b) Baugh-Wooley multiplier gray-cell

Figure 5(B)

Architecture shown in figure is for 4x4 multiplications. Same architecture is replicate for 16x16 architecture.

2.1 Result of VHDL code 16-bit Baugh-Wooley multiplier

VHDL code of 16 bit Baugh-Wooley multiplier is shown in figure. a[15:0] is first 16- bit number and b[15:0] is second bit number p[31:0] is product of a and b.

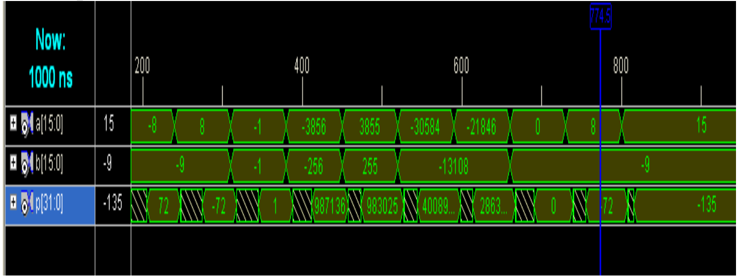
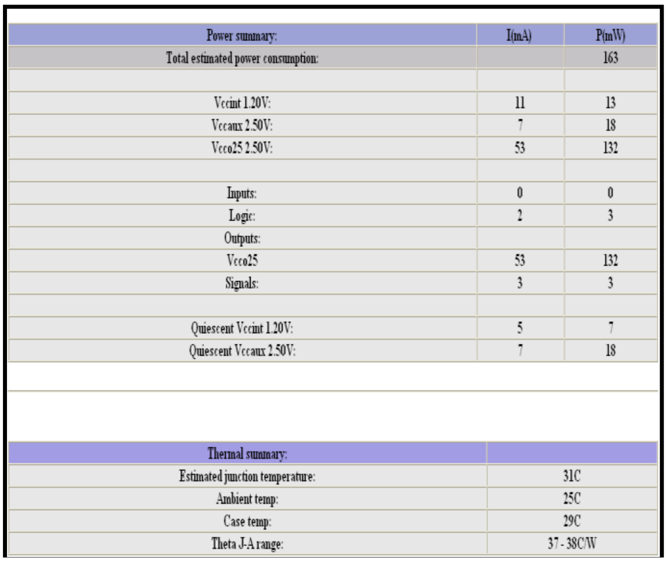


Figure. 6. Simulation Result of 16 bit Baugh-Wooley multiplier



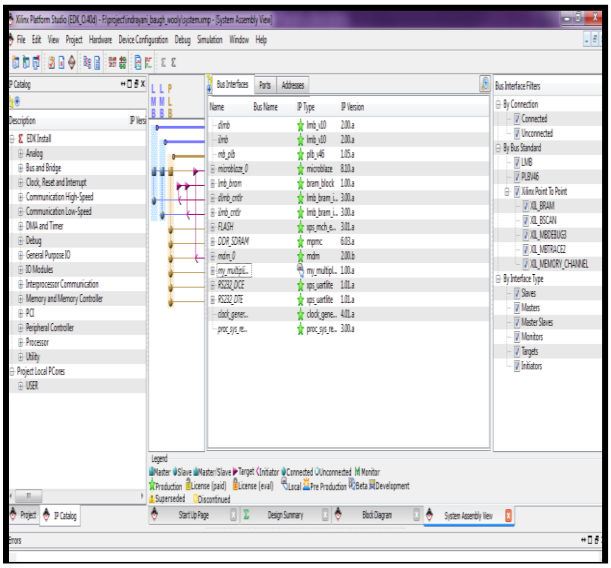
Power summary:	I(mA)	P(mW)
Total estimated power consumption:		163
Vccint 1.20V:	11	13
Vccaux 2.50V:	7	18
Vcco25 2.50V:	53	132
Inputs:	0	0
Logic:	2	3
Outputs:		
Vcco25	53	132
Signals:	3	3
Quiescent Vccint 1.20V:	5	7
Quiescent Vccaux 2.50V:	7	18

Thermal summary:	
Estimated junction temperature:	31C
Ambient temp:	25C
Case temp:	29C
Theta J-A range:	37 - 38C/W

Figure 7. Power report of 16-bit Baugh-Wooley multiuplier

Baugh-Wooley multiplier module in MicroBlaze Processor:

System assembly view of Baugh-Wooley multiplier on Microblaze Processor is shown in figure 8. my_multiplier is our custom hardware of 16- bit Baugh-Wooley multiplier.



Name	Bus Name	IP Type	IP Version
dlmb		lmb_v10	2.00.a
ilmb		lmb_v10	2.00.a
mb_plb		plb_v46	1.05.a
microblaze_0		microblaze	8.10.a
lmb_bram		bram_block	1.00.a
dlmb_cntlr		lmb_bram_i...	3.00.a
ilmb_cntlr		lmb_bram_i...	3.00.a
FLASH		xps_mch_e...	3.01.a
DDR_SDRAM		mpmc	6.03.a
mdm_0		mdm	2.00.b
my_multipli...		my_multipl...	1.00.a
RS232_DCE		xps_uartlite	1.01.a
RS232_DTE		xps_uartlite	1.01.a
clock_gener...		clock_gene...	4.01.a
proc_sys_re...		proc_sys_re...	3.00.a

Figure 8. System assembly view of Baugh-Wooley multiplier on Microblaze Processor

Block diagram of implementation of our architecture in microblaze processor is shown in figure 9.

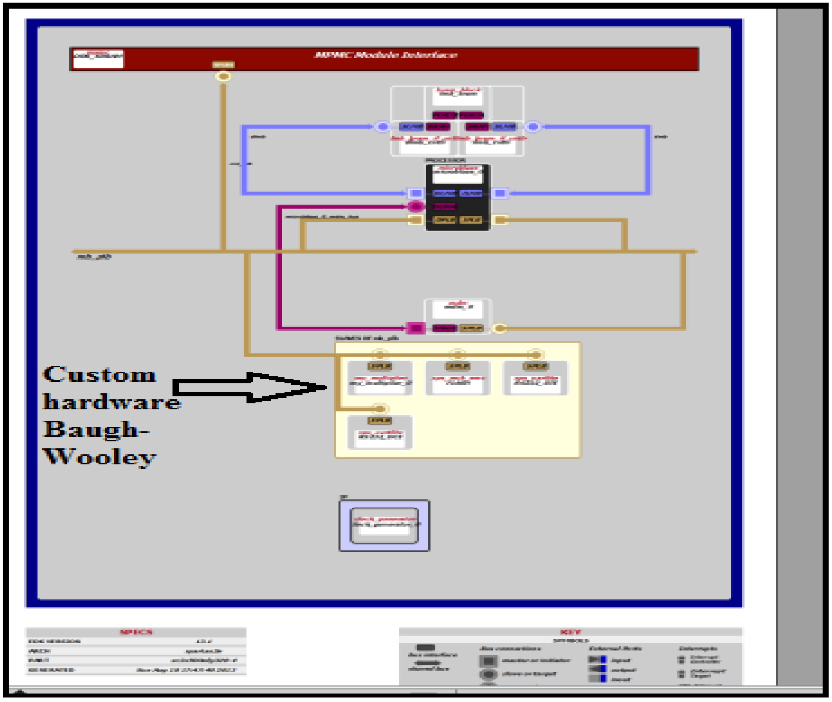


Figure8. Block diagram of implementation of our architecture in microblaze processor

II. CONCLUSION

This paper has described the process of implementation of Baugh-Wooley multiplier based on MicroBlaze soft core processor. Since software implementation results in slower speed, so to increase the computational speed, custom hardware of multiplier block is designed and interface with MicroBlaze processor. Also VHDL code of multiplier is power optimized, takes 163 mW of power.

This fast and power optimized Baugh –Wooly multiplier hardware block can be used in future for implementation of 8-bit FFT, 16 bit FFT, 32 bit FFT etc.

REFERENCES

[1] MicroBlaze Processor Reference Guide Embedded Development Kit EDK 13.1, www.xilinx.com

[2] "Xilinx FPGA Silicon Devices", Xilinx Inc. 2006, http://xilinx.com/products/silicon_solutions/fpgas/

[3] "Xilinx Logic Design and Embedded Design Tools", 2006, http://xilinx.com/products/design resources/design tool

[4] "NIOS 3.0 CPU Data Sheet", Altera Corporation, 2004 http:Hwww.altera.com/literature/ds/ ds_nios_cpu.pdf

[5] "IBM PowerPC Quick Reference Guide", IBM Corp. 2005.

[6] Jiri Gaisler, Sandi Habinc, Edvin Catovic: "GRLIB IP Library User's Manual", Gaisler Research, 2006,http:Hwww.gaisler.com/products/grlib/grlib.pdf

[7] PramodiniMohanty ,VLSIDesign, Department of Electrical &Electronics Engineering Noida Institute of Engineering & Technology 2011-2012:" An Efficient Baugh-WooleyArchitecture forBothSigned & Unsigned Multiplication", PramodiniMohanty et al./ International Journal of Computer Science & Engineering Technology (IJCSET)

[8] Mahzad Azarmehr, Supervisor: Dr. M. Ahmadi,:" Multipliers, Algorithms, and Hardware Designs", RESEARCH CENTRE FOR INTEGRATED MICROSYSTEMS - UNIVERSITY OF WINDSOR

[9] Laxman S, Darshan Prabhu R,Mahesh S Shetty ,Mrs.Manjula BM, Dr.Chirag Sharma:" FPGA Implementation of Different Multiplier Architectures", International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459,Volume 2, Issue 6, June2012)

[10] Wiatr, K.,:" Implementation of multipliers in FPGA structures " ,Quality Electronic Design, 2001 International Symposium on Digital Object Identifier: 10.1109/ISQED.2001.915265 Publication Year: 2001 , Page(s): 415 - 420 Cited by: Papers (3) IEEE Conference Publications