Research Article

Implementation of New VLSI Architecture of Multiplier and Accumulator using Carry Save Adder

Authors:

[1] S.N.V.P.Ravi Teja, [2] M. Chaitanya Kiran, [3]A.Narayana Kiran

Address for Correspondence:

[1]M.Tech II Year, ECE Dept. ASR Institute of Technology, Tadepalligudem, A.P
[2]Assistant Professor, ECE Dept., ASR Institute of Technology, Tadepalligudem, A.P
[3] ECE Dept., Aditya Engineering College, Surampallem, A.P

Abstract: In this paper, we proposed a new architecture of multiplier-and-accumulator (MAC) for high-speed arithmetic. By combining multiplication with accumulation and devising a hybrid type of carry save adder (CSA), the performance was improved. Since the accumulator that has the largest delay in MAC was merged into CSA, the overall performance was elevated. The CSA propagates the carries to the least significant bits of the partial products and generates the least significant bits in advance to decrease the number of the input bits of the final adder. Also, the proposed MAC accumulates the intermediate results in the type of sum and carries bits instead of the output of the final adder, which made it possible to optimize the pipeline scheme to improve the performance. Based on the theoretical and experimental estimation, we analyzed the results such as the amount of hardware resources, delay, and pipelining scheme.

Keywords: Modified Booth multiplier, CLA, multiplier and accumulator (MAC).

1. Introduction

In the majority of digital signal processing (DSP) applications the critical operations usually involve many multiplications and/or accumulations. For real-time signal processing, a high speed and high throughput Multiplier-Accumulator (MAC) is always a key to achieve a high performance digital signal processing system.

In the last few years, the main consideration of MAC design is to enhance its speed. This is because; speed and throughput rate is always the concern of digital signal processing system. But for the epoch of personal communication, low power design also becomes another main design consideration. This is because; battery energy available for these portable products limits the power consumption of the system. Therefore, the main motivation of this work is to investigate various Pipelined multiplier/accumulator architectures and circuit design techniques which are suitable for implementing high throughput signal processing algorithms and at the same time achieve low power consumption.

A conventional MAC unit consists of (fast multiplier) multiplier and an accumulator that contains the sum of the previous consecutive products. The function of the MAC unit is given by the following equation: $F = \Sigma A$.The main goal of a DSP processor design is to enhance the speed of the MAC unit, and at the same time limit the power consumption. Ina pipelined MAC circuit, the delay of pipeline stage is the delay of a1-bit full adder. Estimating this delay will assist in identifying the overall delay of the pipelined MAC. In this work, 1-bit full adder is designed. Area, power and delay are calculated for the full adder, based on which the pipelined MAC unit is designed for low power.

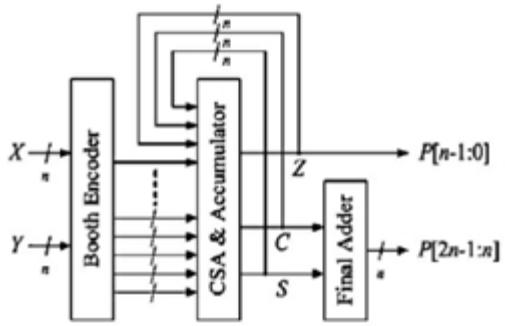


Fig 1: Hardware architecture of the proposed MAC.

2. **Derivation of MAC Arithmetic**

If an operation to multiply two -bit numbers and accumulates into a 2-bit number is considered, the critical path is determined by the 2-bitaccumulation operation. If a pipeline scheme is applied for each step in the standard design of Fig 1, the delay of the last accumulator must be reduced in order to improve the performance of the MAC. The overall performance of the proposed MAC is improved by eliminating the accumulator itself by combining it with the CSA function. If the accumulator has been eliminated, the critical path is then determined by the final adder in the multiplier. The basic method to improve the performance of the final adder is to decrease the number of input bits. In order to reduce this number of input bits, the multiple partial products are compressed into a sum and a carry by CSA. The number of bits of sums and carries to be transferred to the final adder is reduced by adding the lower bits of sums and carries in advance within the range in which the overall performance will not be degraded. A 2-bit CLA is used to add the lower bits in the CSA. In addition, to increase the output rate when pipelining is applied, the sums and carrys from the CSA are accumulated instead of the outputs from the final adder in the manner that the sum and carry from the CSA in the previous cycle are inputted to CSA. Due to this feedback of both sum and carry, the number of inputs to CSA increases, compared to the standard design and In order to efficiently solve the increase in the amount of data, a CSA architecture is modified to treat the sign bit.

3. Equation Derivation:

The aforementioned concept is applied to to express the proposed MAC arithmetic. Then, the multiplication would be transferred to a hardware architecture that complies with the proposed concept, in which the feedback value for accumulation will be modified and expanded for the new MAC. First, if the multiplication in (4) is decomposed and rearranged, it becomes

$$X \times Y = d_0 2Y + d_1 2^2 Y + d_2 2^4 Y + \ldots + d_{N/2-1} 2^{N-2} Y.$$

If this is divided into the first partial product, sum of the middle partial products, and the final partial product, it can be re expressed as. The reason for separating the partial product addition as is that three types of data are fed back for accumulation, which are the sum, the carry, and the preadded results of the sum and carry from lower bits. Now, the proposed concept is applied.

$$Z = \sum_{i=0}^{N-1} z_i 2^i + \sum_{i=N}^{2N-1} z_i 2^i.$$

$$\sum_{i=N}^{2N-1} z_i 2^i = \sum_{i=0}^{N-1} z_{N+i} 2^i 2^N = \sum_{i=0}^{N-2} (c_i + s_i) 2^i 2^N.$$

$$Z = \sum_{i=0}^{N-1} z_i 2^i + \sum_{i=0}^{N-2} c_i 2^i 2^N + \sum_{i=0}^{N-2} s_i 2^i 2^N.$$

$$P = \left(d_0 2^0 + \sum_{i=1}^{N/2-2} d_i 2^{2i} Y + d_{N/2-1} 2^{N-2} Y \right) + \left(\sum_{i=0}^{N-1} z_i 2^i 2^0 + \sum_{i=0}^{N-2} c_i 2^i 2^0 + \sum_{i=0}^{N-2} s_i 2^i 2^0 \right).$$

$$P = \left(d_0 2^0 + \sum_{i=1}^{N-1} z_i 2^i \right) + \left(\sum_{i=1}^{N/2-1} d_i 2^{2i} Y + \sum_{i=0}^{N-2} c_i 2^i 2^0 \right) + \left(d_{N/2-1} 2^{N-2} Y + \sum_{i=0}^{N-2} s_i 2^i 2^0 \right).$$

If is first divided into upper and lower bits and rearranged, (8) will be derived. The first term of the righthand side in (8) corresponds to the upper bits. It is the value that is fed back as the sum and the carry. The second term corresponds to the lower bits and is the value that is fed back as the addition result for the sum and carries the MAC arithmetic is

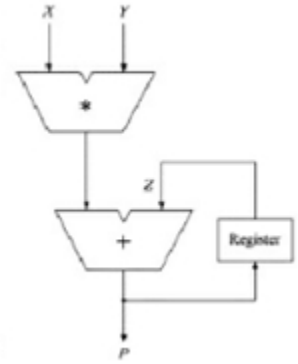


Fig 2 : Hardware architecture of general MAC

4. **Proposed CSA Architecture**

The architecture of the hybrid-type CSA that complies with the operation of the proposed MAC is shown in Fig. 5, which performs 8-bit operation. In Fig. 2.11Si is to simplify the sign expansion and Ni is to compensate 1's complement number into 2's complement number. S[i] and C[i] correspond to the ith bit of the feedback sum and carry. Z[i] is the ith bit of the sum of the lower bits for each partial product that were added in advance and Z'[i] is the previous result. In addition, Pj[i]corresponds to the ith bit of the jth partial product. Since the multiplier is for 8 bits, totally four partial products are generated from the Booth encoder. This CSA requires at least four rows of FAs for the four partial products. Thus, totally five FA rows are necessary since one more level of rows are needed for accumulation. For an -bit MAC operation, the level of CSA is (n/2+1). The white square in Fig. 2.11 represents an FA and the gray square is a half adder (HA). The rectangular symbol with five inputs is a 2-bit CLA with a carry input

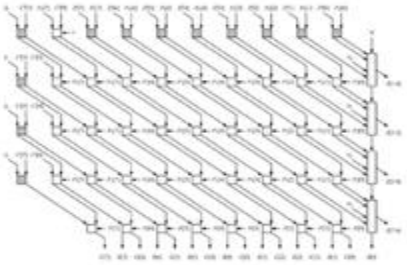


Fig 3: Architecture of the proposed CSA tree:

The critical path in this CSA is determined by the 2-bit CLA. It is also possible to use FAs to implement the CSA without CLA. However, if the lower bits of the previously generated partial product are not processed in advance by the CLAs, the number of bits for the final adder will increase. When the entire multiplier or MAC is considered, it degrades the performance. In Table I, the characteristics of the proposed CSA architecture have been summarized and briefly compared with other architectures. For the number system, the proposed CSA uses 1'scomplement, but ours uses a modified CSA array without sign extension. The biggest difference between ours and the others is the type of values that is fed back for accumulation. Ours has the smallest number of inputs to the final adder

5. Hardware and software used:

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by the customer or designer after manufacturing. Hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare). FPGAs can be used to implement any logical function that an ASIC could perform.

FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together" somewhat like a one-chip programmable breadboard. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. The area of field programmable gate array (FPGA) design is evolving at a rapid pace. The increase in the complexity of the FPGA's architecture means that it can now be used in far more applications than before. The newer FPGAs are steering away from the plain vanilla type "logic only" architecture to one with embedded dedicated blocks for specialized applications.

Definitions of Relevant Terminology are

- Field-programmable Device (FPD) — a general term that refers to any type of integrated circuit used for implementing digital hardware, where the chip can be configured by the end user to realize different designs.

- PLA — a Programmable Logic Array (PLA) is a relatively small FPD that contains two levels of logic, an AND-plane and an OR-plane, where both levels are programmable.

- PAL— a Programmable Array Logic (PAL) is a relatively small FPD that has a programmable AND-plane followed by a fixed OR-plane. SPLD — refers to any type of Simple PLD, usually either a PLA or PAL. CPLD — a more Complex PLD that consists of an arrangement of multiple SPLD-like blocks on a single chip.

6. The FPGA Landscape

In the semiconductor industry, the programmable logic segment is the best indicator of the progress of technology. No other segment has such varied offerings as field programmable gate arrays. It is no wonder that FPGAs were among the first semiconductor products to move to the 0.13µm technology, and again recently to 90nm technology.

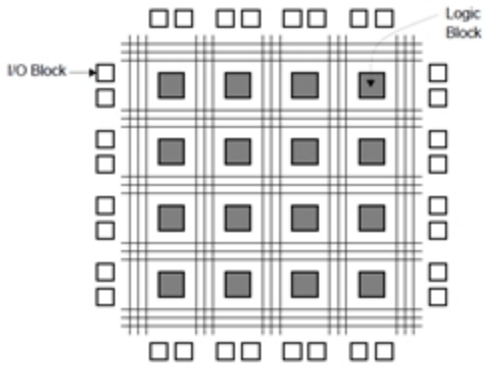


Fig 4: Structure of an FPGA

The players in the current programmable logic market are Altera, Atmel, Actel, Cypress, Lattice, Quick logic and Xilinx. Some of the larger and more popular device families are: Stratix™ from Altera, Accelerator from Actel, is XPGA™ from Lattice and Virtex™ from Xilinx. Between these FPGA devices, many major electronics applications such as communications, video, image and digital signal processing, storage area networks and aerospace are covered.

7. **FPGA synthesis: The Vendor-Independent Approach**

Dedicated memory blocks offer data storage and can be configured as basic single-port RAMs, ROMs (read only memory), FIFOs (first in first out), or CAMs (Content Addressable m\Memory). Data processing or the logic fabric of these FPGAs varies widely in size with the biggest Xilinx Virtex-II Pro™ offering up to 100K LUT4s. The ability to interface the FPGA with backplanes, high-speed buses, and memories is possible by the availability of various single-ended and differential I/O standards support. Many of the major electronics applications such as communications, video, image and digital signal processing; storage area networks and aerospace are covered between the above-mentioned FPGA devices. In a similar manner, for programmable systems applications requiring embedded processors, the Virtex-II Pro™ with its 32-bit RISC processor (PowerPC 405) would be an ideal choice.

	Xilinx virtex II Pro	Altera Stratix	Actel Axcelerator	Lattice is pXPGA
Clock management	DCM Up to 12	PLL Up to 12	PLL Up to 8	Sys CLOCK PLL up to 8
Embedded memory blocks	Block RAM Up to 10 M bit	Tri Matrix Memory Up to10 M bit	Embedded RAM Up to 338K	Sys MEM Blocks Up to 414K
Data processing	CLB and 18-bitx 18-bit Multipliers	LE's and embedded multipliers	Logic modules (C-cell &R-cell)	PFU based
Programmable I/O s	Select IO	Advanced IO Support	Advanced IO Support	Sys IO
Special features	Embedded power PC405 Cores	DSP blocks	Per pin FIFO's for bus application	Sys Hs 1 for high speed serial interface

Table 4.1 Features Offered In FPGA

8. Applications of FPGAs

A list of typical applications includes: random logic, integrating multiple SPLDs, device controllers, communication encoding and filtering, small to medium sized systems with SRAM blocks, and many more.

9. Software's used

We have used Modelsim, and QuartersII. Let us see in brief**.**

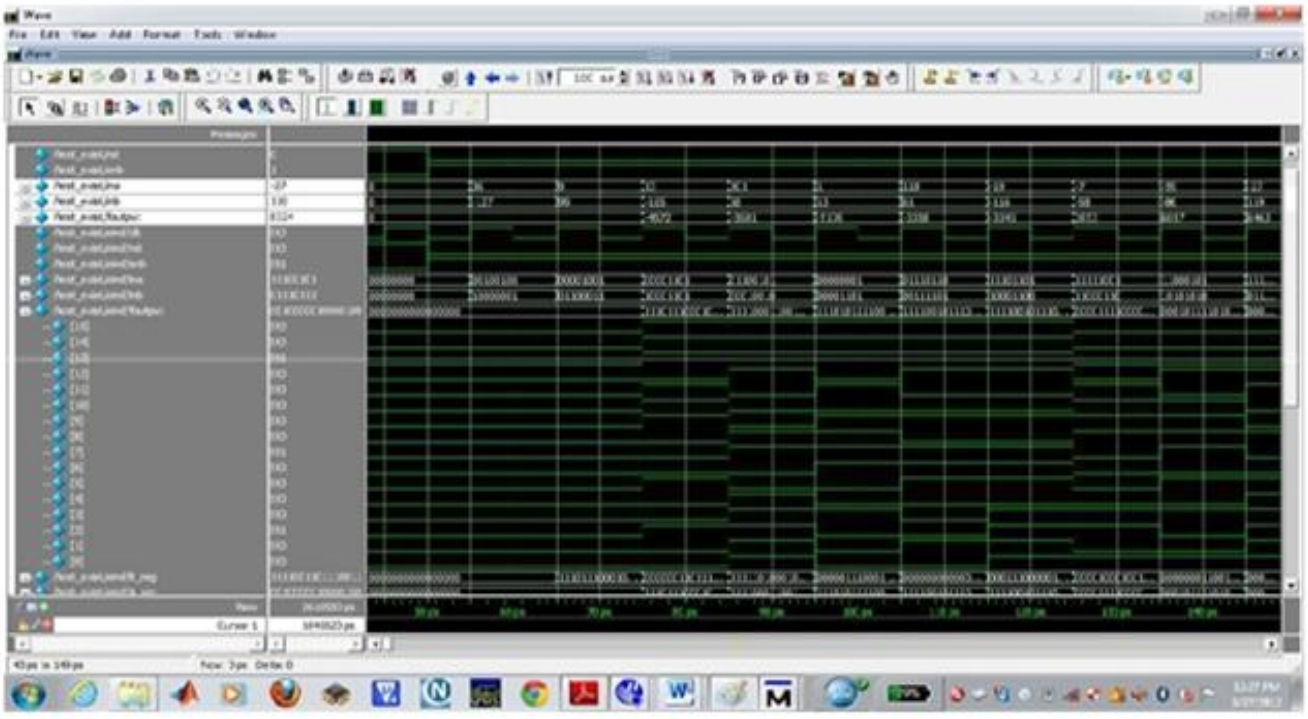
9.1 *Model sim:*

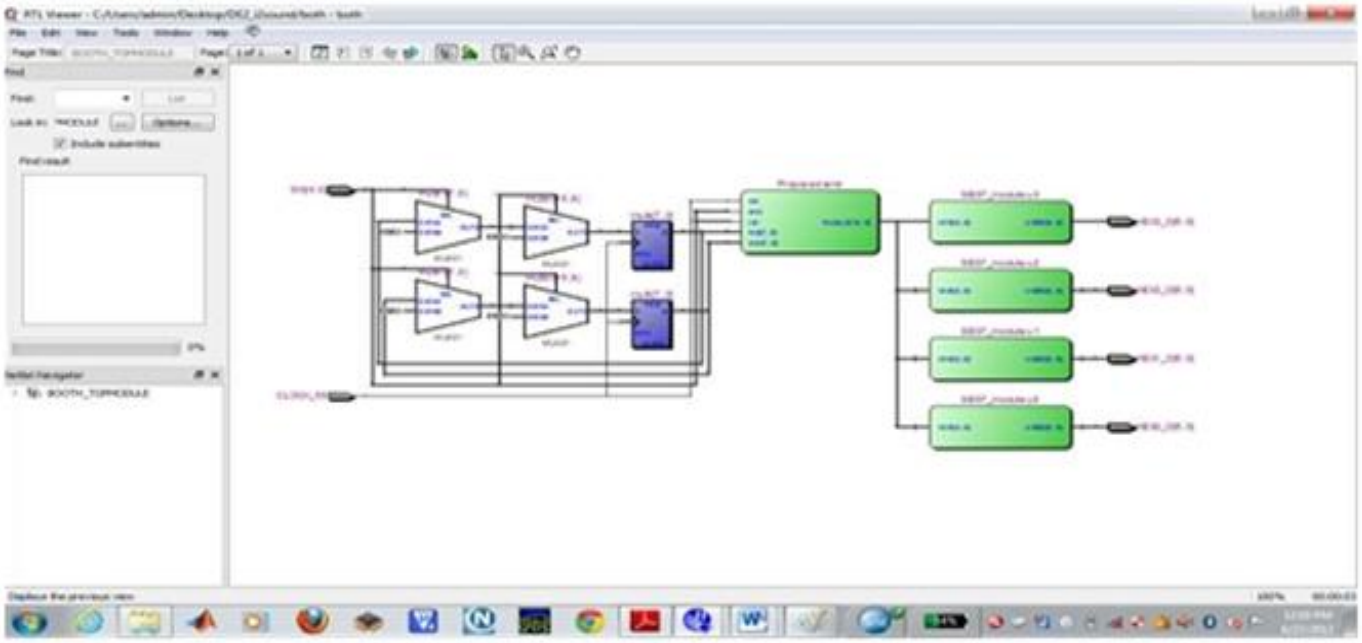
High Performance and Capacity Mixed HDL Simulation - Model Sim Mentor Graphics was the first to combine single kernel simulator (SKS) technology with a unified debug environment for Verilog, VHDL, and System C. The combination of industry-leading, native SKS performance with the best integrated debug and analysis environment make ModelSim the simulator of choice for both ASIC and FPGA design. The best standards and platform support in the industry make it easy to adopt in the majority of process and tool flows.

9.2 Q*uartus II:*

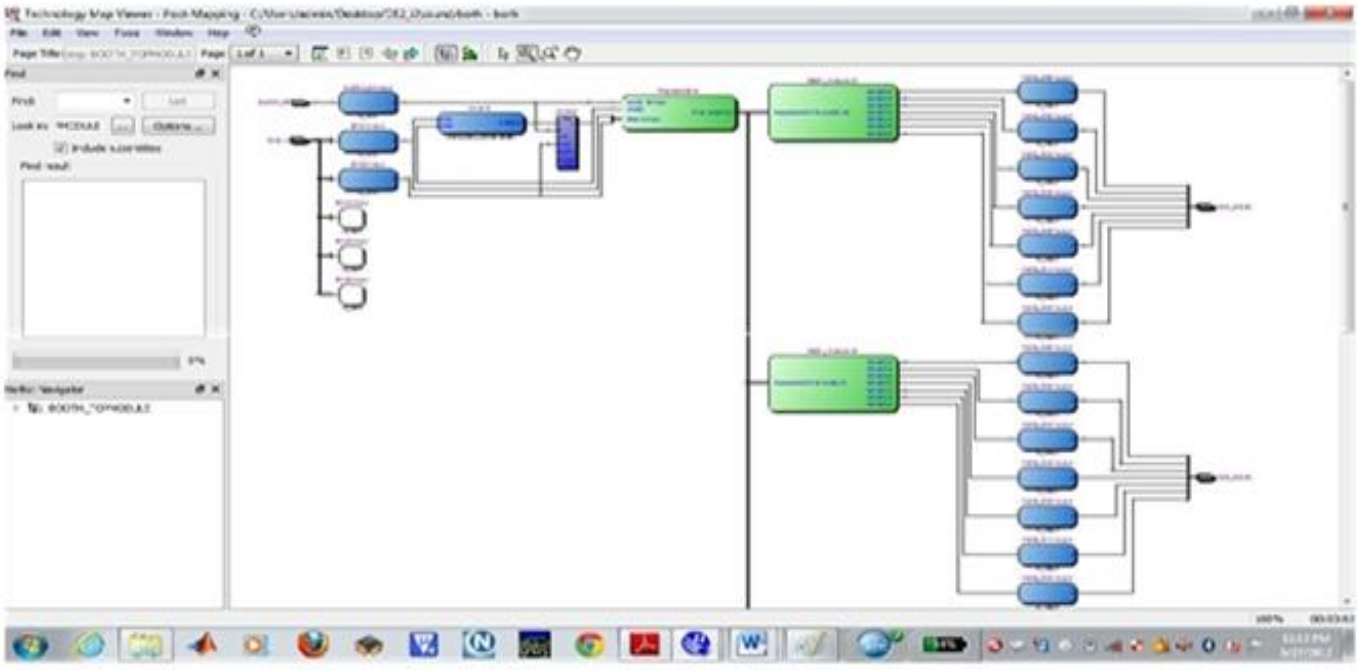
Quartus II is a software tool produced by Altera for analysis and synthesis of HDL designs, which enables the developer to compile their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

Simulation Results

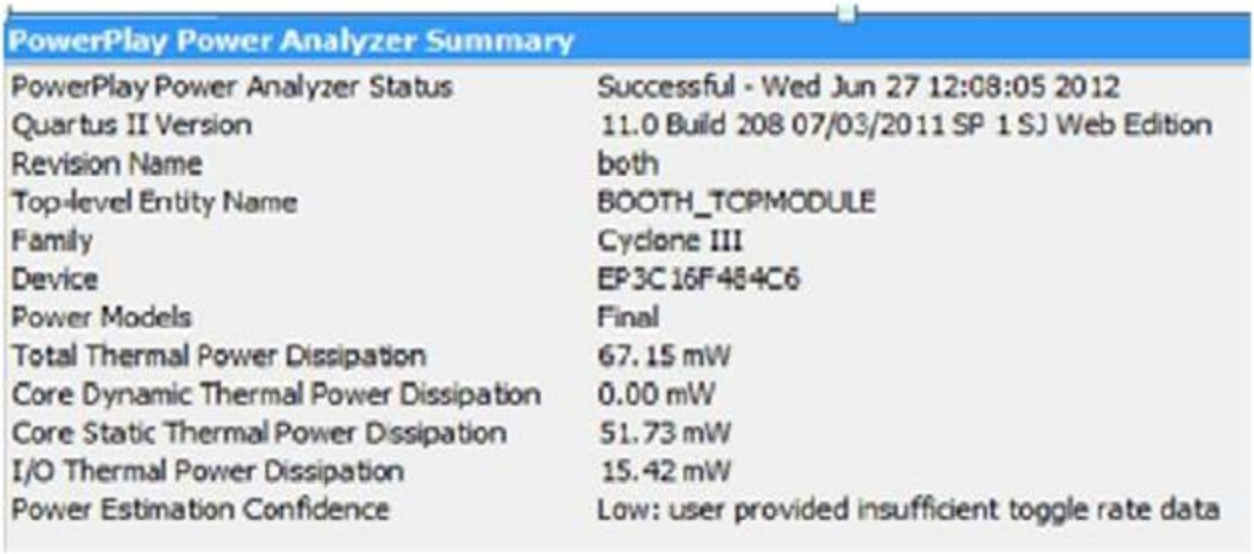




RTL Schematic



Technology map viewer



PowerPlay Power Analyzer Summary	
PowerPlay Power Analyzer Status	Successful - Wed Jun 27 12:08:05 2012
Quartus II Version	11.0 Build 208 07/03/2011 SP 1 SJ Web Edition
Revision Name	both
Top-level Entity Name	BOOTH_TCPMODULE
Family	Cyclone III
Device	EP3C16F484C6
Power Models	Final
Total Thermal Power Dissipation	67.15 mW
Core Dynamic Thermal Power Dissipation	0.00 mW
Core Static Thermal Power Dissipation	51.73 mW
I/O Thermal Power Dissipation	15.42 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Power dissipation report

10. **Conclusions and Future work:**

In this paper, a new MAC architecture to execute the multiplication-accumulation operation, which is the key operation, for digital signal processing and multimedia information processing efficiently, was proposed. By removing the large number of Partial products that has the largest delay ,we proposed high radix booth radix algorithm in order to reduce the partial products, the overall MAC hardware efficiency is increased in almost twice as much as in the previous work.

11. References:

[1] F. Elguibaly, "A fast parallel multiplier- accumulator using the modified Booth algorithm," IEEE Trans. Circuits Syst., vol. 27, no. 9, pp. 902-908, Sep. 2000

[2] Information Technology-Coding of Moving Picture and Associated Audio, MPEG-2 Draft International Standard, ISO/IEC 13818-1, 2, 3, 1994.

[3] J. J. F. Cavanagh, Digital Computer Arithmetic. New York: McGraw-Hill, 1984

[4] JPEG 2000 Part I Fina1119l Draft, ISO/IEC JTC1/SC29 WG1.

.[5] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," IEEE J.