Vol 06, Article 12519; January 2015
http://ijves.com

International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

A NOVEL DESIGN APPROACH TO INCREASE THE SPEED OF VLSI CIRCUITS IN MIXED-SIGNAL ENVIRONMENT

R.PRAKASH RAO[1], B.K.MADHAVI[2]

[1]Associate Professor, St.Peter's Engineering College,Near Forest Academy, Dulapally,Hyderabad,India.
[2]Professor, Sridevi Women's Engineering College, V.N.Pally, Near Gandipet, R.R. Dist,India.
[1] prakashiits@gmail.com, [2] bkmadhavi2009@gmail.com

ABSTRACT

Currently the mixed signal circuits like A/D and D/A converters are being designed either by MATLAB, C & C++ or VHDL-AMS. But, these languages do not give detailed information at the architecture level. Though VHDL-AMS(VHSIC Hardware Description Language-Analog & Mixed Signal Extensions) is used for hardware description for mixed signal circuits, it is used for simulation only but not used for synthesis. There are no synthesis tools for VHDL-AMS. Since, synthesis is not possible using VHDL-AMS, further optimization of the mixed-signal designs is also not possible. Hence, to eliminate this difficulty, mixed-signal circuits are to be implemented by VHDL but not by VHDL-AMS. Using VHDL, plenty of synthesis tools are available so that further optimization is possible. As further optimization is possible, speed of the mixed-signal design can be increased.

***Key words**: mixed signal circuits, A/D and D/A converters, VHDL-AMS, simulation, synthesis, VHDL, optimization.*

[1] INTRODUCTION

Design methodologies play the vital role in the designing of the VLSI integrated circuits [1]. There are different design methodologies or approaches for different VLSI circuit designs i.e., for digital designs, analog designs and mixed-signal designs the design approaches are different. If a new design methodology is proposed, to accept the new design methodology there should be a tool support also. For designing the mixed-signal circuits currently VHDL-AMS is used. But, VHDL-AMS is used for simulation only so that further optimization is not possible [2]. To resolve this difficulty, a novel design methodology is required for mixed-signal circuits, by which the design should be able to synthesizable. To meet the above requirement, the total work is explored in this paper as given below. Section II briefs the different design methodologies. Section III gives the development of library packages. Section IV briefs the design of DWT as mixed signal design. Section V deals the novel work to optimize the parameters. Section VI and VII explain the results and conclusions respectively.

2. DESIGN APPROCHES:

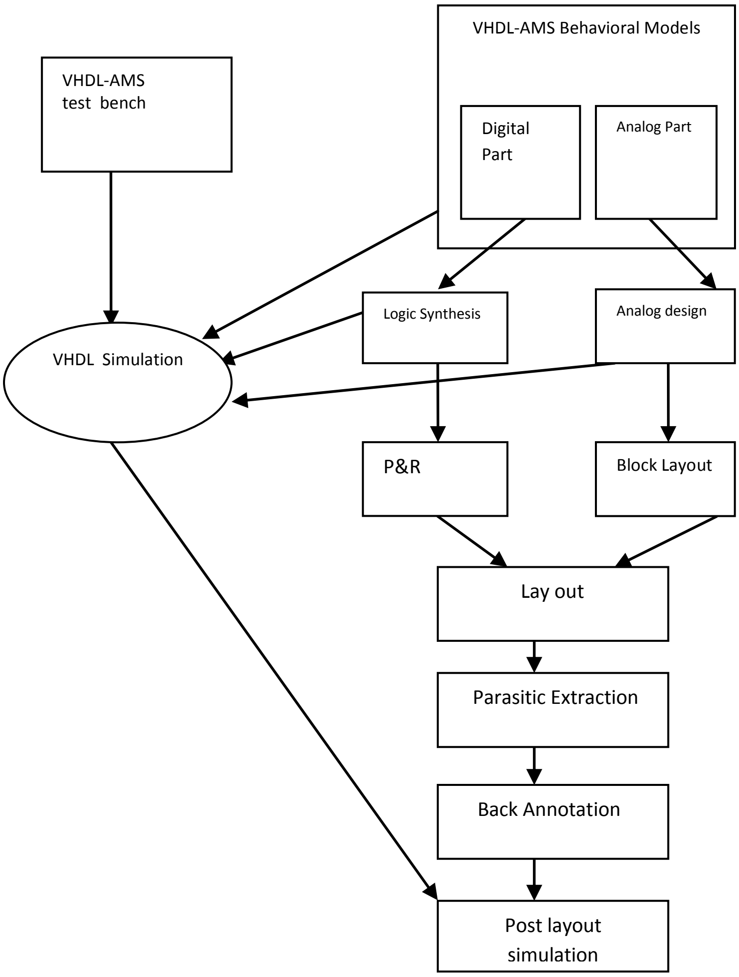
Currently mixed-signal circuits are being designed by VHDL-AMS. But VHDL-AMS is used for simulation only. Synthesis is not possible for the mixed-signal design developed by VHDL-AMS. To get synthesis a novel design approach is required. Hence, this section covers both the design approaches.

2a) CURRENT DESIGN APPROACH:

In this approach, the overall system is developed using behavioral model or RTL model and then verified [3]. In this methodology a system will be modeled using VHDL-AMS language. In this, both digital and analog parts will be analyzed separately as shown in below figure(1). The digital part will be analyzed using the RTL synthesizable code of the HDL language. Similarly the analog part is analyzed by dividing it into the functional blocks at the behavior level, that is as filters, VCO$_s$ etc. For the whole model the test benches will be written in VHDL-AMS [4].

After system level modeling using VHDL-AMS, the digital part of the system could be synthesized by a logic synthesizer. The logic synthesizer will convert the RTL code into a gate level net-list. Whereas analog blocks are designed at the transistor level individually. After completing the design of each and every block, to test the interaction among the various blocks, test benches will be developed at the earlier stage only. From the gate level net-list, by using place and route(P&R) tools, the layout of the digital part would be estimated. Similarly, the analog blocks layout is usually generated manually or by using the dedicated module generators.

At this layout stage, both the analog and digital parts will be combined to each other. To perform the schematic driven placement for the standard cells or devices and for routing the layout, cadence virtuoso-XL is used. For place and route, block layouts are used at the higher level. Hence, it is very important to maintain some consistency regarding the type metal layers used or where ground pins or power pins are located is required. After combining both analog and digital parts at the layout level, the next part is, extracting the parasitic elements from both. The parasitic elements are the R,L,C components which may provide the delay between any two layouts. The parasitic elements obtained from the digital part are used to compute the delays and these are stored in standard delay format (SDF) file. Whereas the parasitic elements related to analog part cannot be estimated at this stage. After extracting the parasitic elements, the next step is back annotation. After back annotation, the next step is post layout verification. The final simulation or overall design, at each and every step will be simulated by using the test-benches written using VHDL-AMS [5],[6].

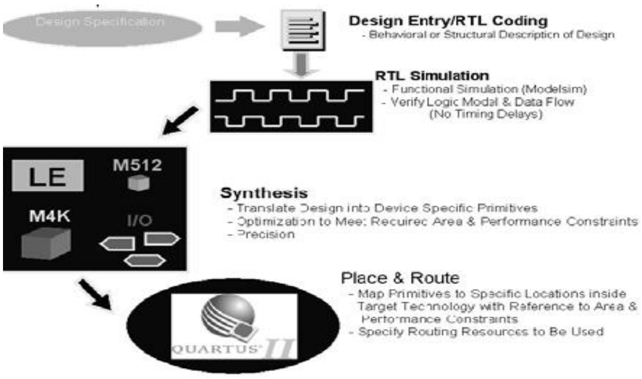


Figure(1):Mixed-signal design flow

2 b) NOVEL DESIGN APPROACH:

Design specifications:

In the novel mixed signal design methodology, design specifications is the first step [7]. The features and functionalities of the ASIC are defined in this step. For this step architecture and micro architecture functionalities are extracted from the specified features and functionalities. In general the features and functionalities include timing restrictions, power consumption, voltage reference, and performance criteria. From the above specifications the chip architecture is drafted. On the drafted architecture, architecture simulations are to be performed to meet the required specifications. If the simulation results are not met the defined specifications, the architectural definition is to be changed again and again till all the requirements are met.


Design Specification
Design Entry/RTL Coding
- Behavioral or Structural Description of Design
RTL Simulation
- Functional Simulation (Modelsim)
- Verify Logic Modal & Data Flow
(No Timing Delays)
LE
M512
M4K
I/O
Synthesis
- Translate Design into Device Specific Primitives
- Optimization to Meet Required Area & Performance Constraints
- Precision
Place & Route
- Map Primitives to Specific Locations inside
Target Technology with Reference to Area &
Performance Constraints
- Specify Routing Resources to Be Used
QUARTUS II

Figure(2): Novel design methodology for mixed- signal designs

Design entry RTL coding:
In this stage, the macro architecture derived from the specifications are converted to RTL synthesizable code. Various methods are there to obtain the RTL code. Some designers use graphical design entry tools. These tools allow the user to use flowcharts, truth tables and bubble diagrams to implement the macro architecture which generate the RTL code subsequently, either in VHDL or Verilog. Some other designers however, prefer writing the RTL code directly rather than using a graphic design entry tool. Either the approach ends at the same result. Hence the RTL synthesized code describes the logic functionality of the specification.

Simulation:
After RTL coding, the immediate step is the creation of test benches. A test bench is a program written by the user to inject a set of stimulus into the inputs of the design to check the o/p of the design to match the o/p to the designer's expectations or not. After that, using the HDL simulator the RTL code and the test bench are simulated. Some of the available simulators for Verilog as well as VHDL are- Mentorgrapghic's Modelsim & cadence's NCSIM are able to simulate both VHDL and Verilog. 'Scirocco' from synopsys is the example of a VHDL simulator. 'VerilogXL' from cadence and 'VCS' from Synopsys are some of the verilog simulations. Not only the above but plenty of simulators for HDL simulation are there. But whatever the simulator is used, the aim is to verify the RTL code of the design for the test bench written. The Designs to be debugged if the outputs don't match the designers expectations. Till bugs are not found, the designs are rewritten and need to rerun the simulation.

Synthesis:
After simulation, it is time to do the synthesis. Synthesis is the process through which the RTL code is to be converted into logic gates. The functionality described by these logic gates is the same as the functionality specified by the RTL code. The more common tools used for synthesis are Design-V compiler from the Synapsys, Ambit from Cadence, XST from Xilinx. To convert RTL code to logic gates the "technology library file" is used which contains standard cells. During the process using the available standard cells the RTL code will be converted into gate level. Upon final optimization, the designer is to verify the specified performance and area. If they are not met them again the micro architecture is to be reconsidered till the required specifications are met.

Place and Route:
In this step, the logic gates which are synthesized by the synthesis tool are placed and are routed. At this stage, the designer is having some degree of flexibility to place the logic gates according to the predefined floor plan. For the critical paths which are tight in terms of timing, the designer is to give highest priority than the other routing paths. For the automatic place and route, the designers are using APR tools. With built in self algorithm, most of the APR tools will be operated.

3. LIBRARY PACKAGE DEVELOPMENT TO CONVERT ANALOG TO DIGITAL

Since, any HDL simulation tool consists of standard HDL library and it supports for digital designs only, to introduce mixed signal designs in HDL simulation tool, user defined 'mathematical package' is developed using IEEE 754 16 bit floating point format to support for the basic arithmetic operations like adder, shifter and multiplier. Since, a HDL tool is not having the provision for floating point operations, With the name of math_pack1 a record is created with type real_single. To support the above methodology, this section covers briefly the algorithm to develop the 'mathematical package'.

3 a) PACKAGES AND LIBRARIES

Packages and libraries provide a convenient way of referencing frequently used functions and components. Packages are the only language mechanism to share objects among different design units. Usually, they are designed to provide standard solutions for specific problems (e.g., data types and corresponding subprograms like type conversion functions for a certain bus protocol, procedures and components (macros) for signal processing purposes, etc.). A package consists of a package declaration and an optional package body. The package declaration contains a set of declarations, which may be shared by several design units (for example: types, signals, components, function and procedure declarations). The body package usually contains the functions and procedure bodies. The syntax rule for a package declaration is:

package identifier is
{package declarations}
begin
{sequential_statement}
end [package] [identifier];

A package is analyzed separately and placed in the working library by the analyzer. Each package declaration that includes function and/or procedure declarations must have a corresponding package body. The syntax rule for a package body is:
package body identifier is
{package body declarations}
end [package body] [identifier];

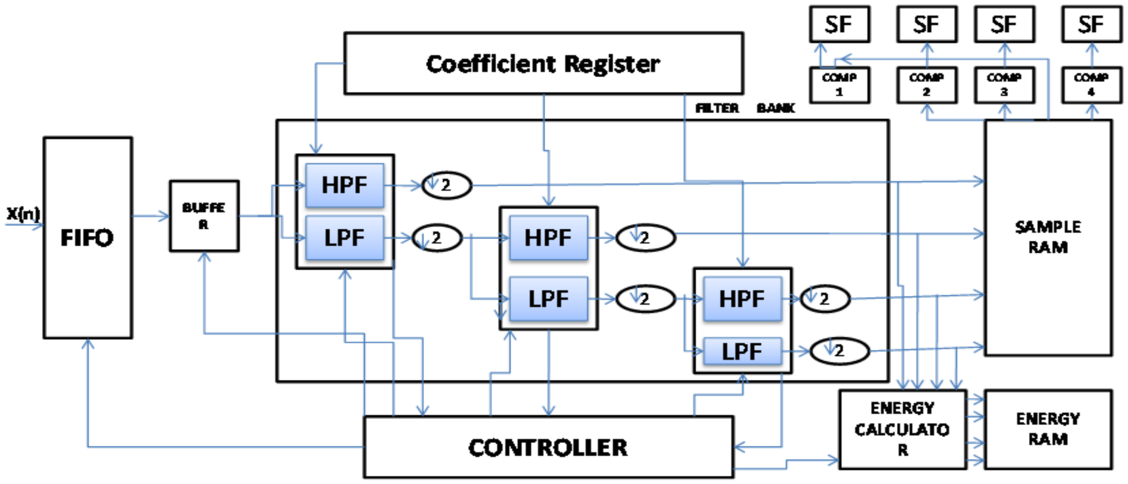
3 b) Algorithm to develop the 'mathematical package': The declarations of the mathematical package, as an algorithm is as given below

Start
⇩
use ieee standard logic 1164 pack
⇩
use ieee standard logic unsigned pack
⇩
use ieee standard logic arithmetic pack
⇩
create package with package name
⇩
record creation with type real_single
⇩
constant declarations
⇩
create ROM with constant
⇩
record creation with type complex
⇩
RAM type memory creation
⇩
constant declaration with above memory
⇩
constants declaration for low pass filter coefficients
⇩
constants declaration for high pass filter coefficients
⇩
creation of real and complex memories for real and imaginary signals
⇩
constant declaration for above memories
⇩
RAM creation for real and imaginary and SRAM signals
⇩
constant declaration for above memories
⇩
divide program with procedures
⇩
procedures declarations

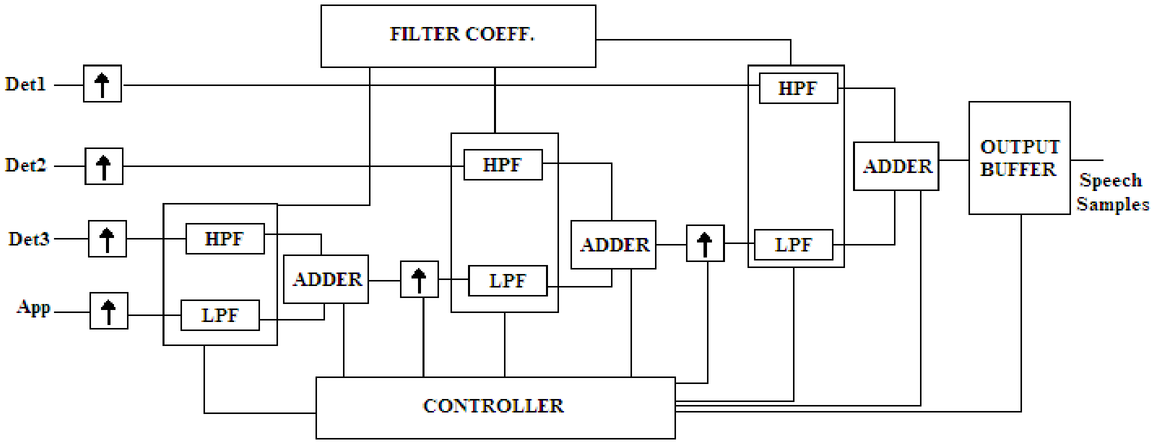
To support the above design methodology, any of the mixed-signal application can be developed [8]. As part of it, here a three stage Discrete Wavelet Transform for signal has been designed using VHDL and proved synthesis is possible, by which speed has been increased. The detailed approach has been explained in the upcoming sections.

4. DESIGN OF 3 STAGE DISCRETE WAVELET TRANSFORM(DWT) FOR SIGNAL

For the designed 16 bit floating point math_pack, any signal processing application can be developed. But, here a three stage DWT decomposer and re-constructor for signal have been developed.



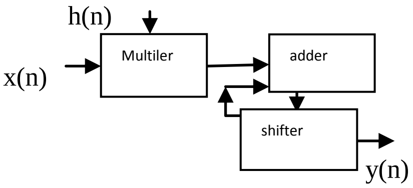
Figure(3): De-composer of Discrete Wavelet Transform for signal



Figure(4):Re-constructor of Discrete Wavelet Transform for signal

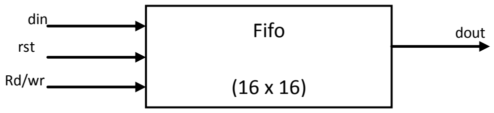
4 a) REALIZATION

The filter logics are realized using MAC (multiply and accumulate) operation where a recursive addition, shifting and multiplication operation are performed to evaluate the output coefficients. The recursive operation logic is as shown below.



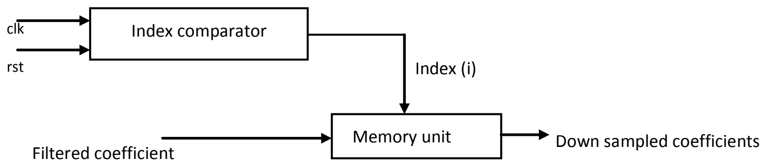
Figure(5): Realization of recursive MAC operation

Before passing the data to filter bank the fifo logic realized stores the data in asynchronous mode of operation [9], operating on the control signals generated by the controller unit. On a read signal the off-centered data is passed to the buffer logic. The fifo logic realized as shown below.



Figure(6): Realization of 16 x 16 fifo logic for coefficient interface

The obtained detail coefficients are down sampled by a factor of two to reduce the number of computation intern resulting in faster operation. To realize the decimator operation comparator logic with a feedback memory element is designed as shown below.



Figure(7): Architecture for decimation by 2 logic

The proposed design(DWT) is realized using VHDL language for its functional definition. The HDL modeling is carried out in top-down approach with user defined package support for floating point operation and structural modeling for recursive implementation of the filter bank logic. For the realization, a package is defined with user defined record data type like

```
type real_single is
record
sign : std_logic;
exp: std_logic_vector(3 downto 0);
mantissa: std_logic_vector(10 downto 0);
end record;
```

The floating notation is implemented using 16 bit IEEE-754 standard in which one bit is assigned to ‘sign’ , four bits are assigned to ‘exponent’ and rest of the eleven bits are assigned to ‘mantissa’. The floating-point addition, multiplication and shifting operation are implemented as procedures in the user defined package and are repeatedly called in the implementation for recursive operation.

4 b) IMPLEMENTATION

The proposed system is realized using VHDL language for it’s functional definition. The HDL modeling is carried out in top-down approach with user defined package support for floating point operation and structural modeling for recursive implementation of the filter bank logic. For the realization a package is defined with user defined record data type as

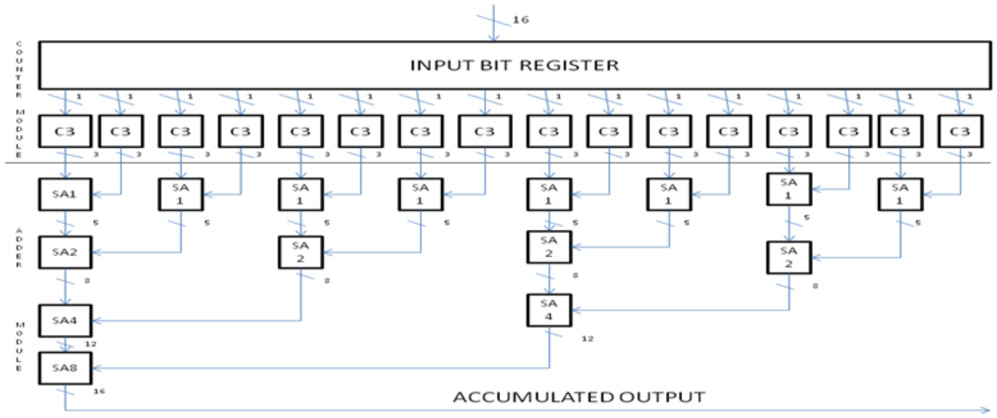
```
    type  real_single is
  record
        sign : std_logic;
        exp: std_logic_vector(3 downto 0);
       mantissa: std_logic_vector(10 downto 0);
     end record;
```

The floating notation is implemented using 16 bit IEEE-754 standards as presented below.

Sign. (1)	Exp. (4)	Mantissa (11)

The floating-point addition, multiplication and shifting operation are implemented as procedures in the user defined package and are repeatedly called in the implementation for recursive operation.

5. NOVEL APPROACH TO INCREASE THE SPEED OF DISCRETE WAVELET TRANSFORM



Figure(8):Multiple accumulations

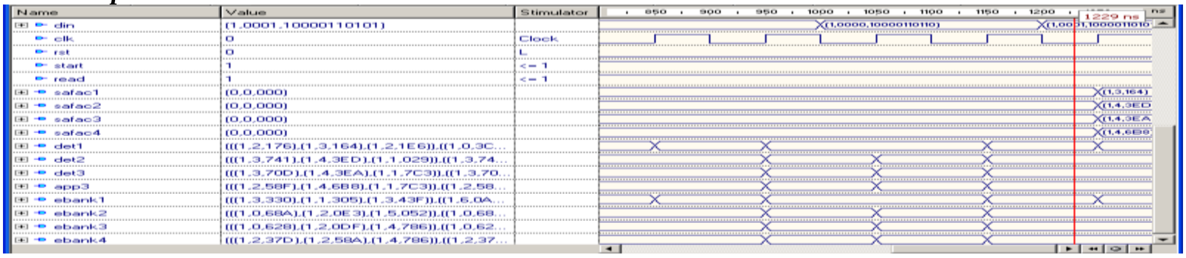
The above figure(8) shows the modified MAC. Repetitive multiple accumulations are frequently encountered in engineering applications

The MAC operation in DSP applications is performed by convolution operation. The convolution operation is given by y(n)=∑ x(n)h(n-1) or Y(n)=X(n)*H(n). This mathematical expression can be modeled as block diagram as shown in figure(5). It performs the accumulation operation in successive clocks of duration which amounts to that of an adding up of the time so that the accumulation of N successive input words are performed in N clock cycles. For large values of N, a computational latency of N addition-time to obtain an accumulated output could be too high to meet the timing requirement in real-time applications. Hence, to speed up the MAC, adder will be developed with the parallel processing. Due to parallel processing speed of the MAC is increased with the expense of little bit hardware.

6.RESULTS

6 a) SIMULATION RESULTS:

Without optimization

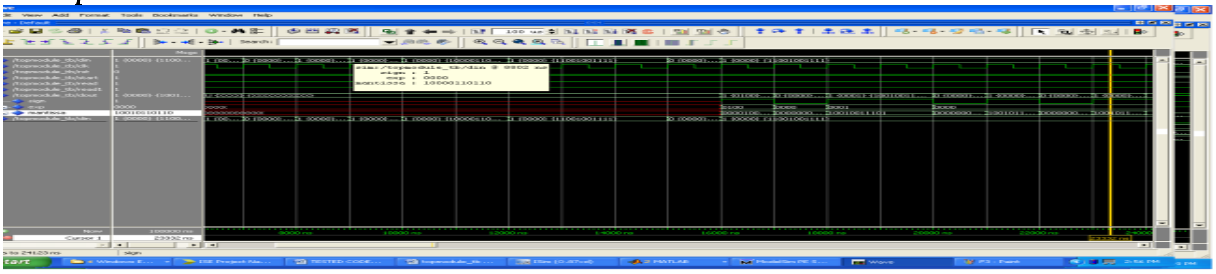


Name	Value	Stimulator
din	(1,0001,10000110101)	
clk	0	Clock
rst	0	L
start	1	<= 1
read	1	<= 1
safac1	(0,0,000)	
safac2	(0,0,000)	
safac3	(0,0,000)	
safac4	(0,0,000)	
det1	(((1,2,176),(1,3,164),(1,2,1E6)),((1,0,3C...	
det2	(((1,3,741),(1,4,3ED),(1,1,029)),((1,3,74...	
det3	(((1,3,70D),(1,4,3EA),(1,1,7C3)),((1,3,70...	
app3	(((1,2,58F),(1,4,6B8),(1,1,7C3)),((1,2,58...	
ebank1	(((1,3,330),(1,1,305),(1,3,43F)),((1,6,0A...	
ebank2	(((1,0,68A),(1,2,0E3),(1,5,052)),((1,0,68...	
ebank3	(((1,0,628),(1,2,0DF),(1,4,786)),((1,0,62...	
ebank4	(((1,2,37D),(1,2,58A),(1,4,786)),((1,2,37...	

Figure(9)Simulation result for the implemented sub-band

Figure.14 shows the input values passed to the sub-band module. The inputs are passed to the module in Floating point Excess-7 notation. The system is passed with a clock of 100Mhz system application frequency with reset signal low as the system considered being active low. The Signal 'start' and 'read' is fed high for making the system enable and to read the data from the buffer element. The result shows the scale factor obtained for each sub-band. The signal 'det1','det2','det3' and 'app3' gives three detailed coefficients and approximate coefficients for the input signal. Each sub-band constitute of 9 sub-samples for every packet of the data burst. The e-bank gives the energy values of each sub-band sample.

With optimization

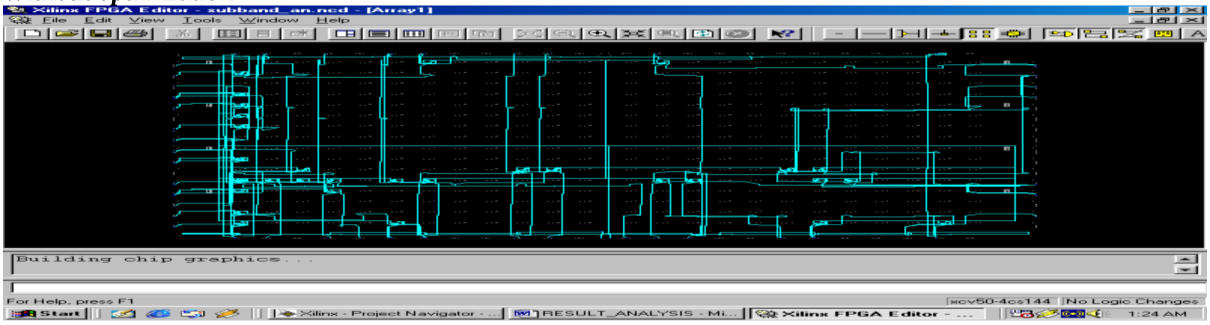


Figure(10): Input, ouput results with floating point values

The floating point values are applied as input to the decomposer of the DWT system and their corresponding output values can be observed at the re-constructor of DWT system. The input and output values of the DWT system at particular instant of time t1 is shown in figure(10)

6 b) SYNTHESIS REPORT:

Without optimization

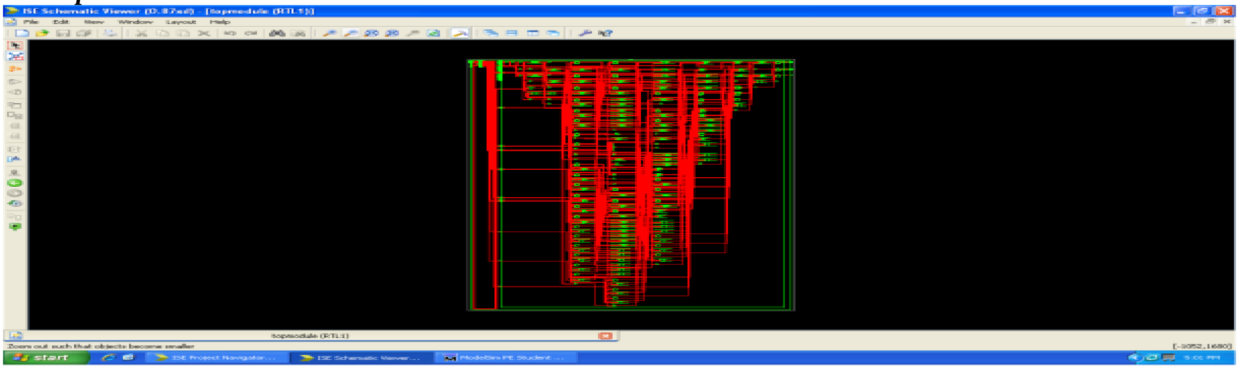


Xilinx FPGA Editor - subband_an.ncd - [Array1]
File Edit View Tools Window Help
Building chip graphics...
For Help, press F1
xcv50-4cs144
No Logic Changes
Start
Xilinx - Project Navigator - ...
RESULT_ANALYSIS - Mi...
Xilinx FPGA Editor - ...
1:24 AM

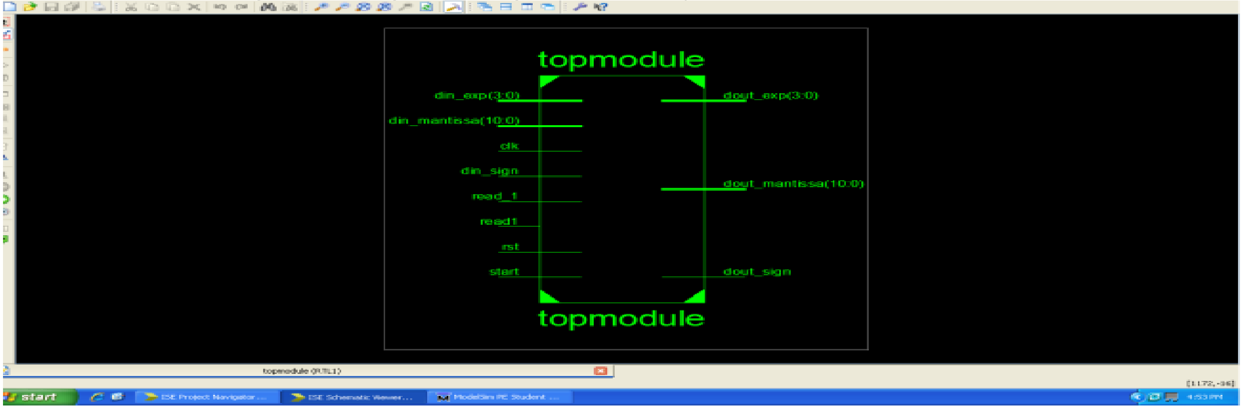
Figure(11): Logical routing of the implemented wavelet decomposing module targeting to Xc2s50e-ft256-7

Figure shows the logical routing of the implemented design targeting on to Xc2s50e-ft256-7 FPGA of Spartan family. The routing is carried out on Xilinx FPGA editor. The result obtained shows the real time FPGA interconnection of logics connected inside the FPGA.

With optimization


ISE Schematic Viewer (O.87xd) - [topmodule (RTL1)]
File
Edit
View
Window
Layout
Help
topmodule (RTL1)
Zoom out such that objects become smaller
start
ISE Project Navigator ...
ISE Schematic Viewer ...
ModelSim PE Student ...

Figure(12): Logical routing of the DWT system after modified MAC.



Figure(13): Top-level module of the 3 stage DWT system for signal

Top level module of the DWT. It shows the input and output pins which relates to the 3-stageDWT developed using 16-bit floating point notation.

Table (1): Design Statistics, Cell usage:

Parameters	**Resources of DWT without optimization**	**Resources of DWT with optimization**
Design statistics:# IOs	105	105
Cell usage:# BELs	205	236

Table (2): Device utilization summery (Selected device: Xc2s50e-ft256-6):

Parameters	**Resources of DWT without optimization**	**Resources of DWT with optimization**
Design statistics:# IOs	105 out of 182 57%	105 out of 182 57%
Cell usage:# BELs	205 out of 1728 11.86%	236 out of 1728 13.65%

Table (3): Timing summery (Speed Grade: -6):

Parameters	**Resources of DWT without optimization**	**Resources of DWT with optimization**
Minimum period	2.649 ns	1.937 ns
Maximum Frequency (speed)	377.501 MHz	516.262 MHz

Table(4): Power dissipation

Parameter	Before optimization	After optimization
Power	38.46mw	41.79mw

CONCLUSION

In real time applications, for mixed signal circuits, the designers are using VHDL-AMS. But, VHDL-AMS is used only for simulation [10]. It is not used for synthesis. Since, synthesis is not possible the mixed signal circuits are not being optimized. Hence, a new design approach for mixed-signal circuits has been proposed. To convert analog signal into digital, a floating point package in the IEEE 754 format has been developed and compiled in the Modelsim 10.3 *PE* . Now as the Modelsim10.3 *PE* tool has been updated for mixed signal designs. To support it a DWT was designed and to increase the speed of DWT a novel algorithm has been proposed with the little expense of hardware. Hence, it is the most appropriate methodology to design the mixed signal circuits with good optimization techniques for all signal processing applications.

REFERENCES

[1]J.-J. Charlot, N. Milet-Lewis, T. Zimmer and H. Lévi, "VHDL-AMS FOR MIXED TECHNOLOGY AND MIXED SIGNAL, AN OVERVIEW", France beams@ixl.u-bordeaux.

[2]Ken Kundert, Henry Chang, Dan Jefferies, Gilles Lamant, Enrico Malavasi, Fred Sendig, "Design of Mixed-Signal Systems on Chip", © 2006, Kenneth S. Kundert – All Rights Reserved.

[3]David Ian Long, "Behavioural Simulation of Mixed Analogue/Digital Circuits", April 1996.

[4]Hany G. Bakeer, Omar Shaheen, Haitham M. Eissa, Mohamed Dessouky, "Analog, Digital and Mixed-Signal Design Flows", 978-1 -4244-1 825-1/07/$25.OO ©2007 IEEE.

[5]Edward Murphy and Steven Bibyk, "Development Strategies for Durable Hardware Language Descriptions of Analog & Mixed Signal Circuits", bibyk@ee.eng.ohio-state.edu.

[6]Ramin Shariat-Yazdi, "Mixed Signal Design Flow A mixed signal PLL case study",Ramin Shariat-Yazdi 2001.

[7]H.E1 Tahaw, D. Rodriguez, S. Garcia-Sabiro, J.-J. Mayol, "VHDeLDO : A New Mixed Mode Simulation", 0-8186-435O-lf9$33 .00 0 1993 IEEE.

[8]Georges G. E. Gielen, Senior member, IEEE, and Rob A. Rutenbar, Fellow, IEEE, "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits", PROCEEDINGS OF THE IEEE, VOL. 88, NO. 12, DECEMBER 2000.

[9]Jung-Lin Yang, Wei-Che Hsu, and Sung-Min Lin, "A Case Study On Asynchronous VLSI Design Platform", 1-4244-0637-4/07/$20.OO ©C2007 IEEE.

[10]Rajeev Narayanan, Naeem Abbasi, Mohamed Zaki, Ghiath Al Sammane, Sofi`ene Tahar, "On the Simulation Performance of Contemporary AMS Hardware Description Languages", 1-4244-2370-5/08/$20.00 ©2008 IEEE.