IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. I (Nov – Dec. 2014), PP 01-13
www.iosrjournals.org

A Modified Approach to Low Power Digital Modeling of Recursive Filters For Analog Signal

R. Rajendra prasad[1] Dr.M.V.Subramanyam[2] Dr. K. Satya Prasad3
[1]*Asso.prof. ECE Dept, N.B.K.R.IST,Vidyanagar,Nellore ,AP, India,*
[2]*Principal and Professor, ECE Dept, Santhi Ram Engineering College,Nandyal, AP,India.*
3Professor, ECE Dept., JNTU Kakinada, Kakinada,AP, India.

Abstract: *The focus of this paper is towards developing an application specific design methodology for low power solutions in recursive coding system in which filtration process is performed by using MAC approach. For the realization of this MAC operation a recursive multiplication and addition operation is carried out. Theconventional recursion results in multiple accumulation operation in successive clocks of duration amounting to that of an addition-time, so that the accumulation of N successive input words are performed in Nclock cycles. For large values of N, a computational latency of Naddition-time over head and high power consumption and accumulation process to obtain an accumulated output could be too high to meet the timing requirement in real-time application. so in order to over this problem in this paper a new scalable Repetitive multiply accumulates (MACs) is proposed.From the simulation results the proposed MAC has less delay complexity and low power consumption when compared to the conventional approaches.*

Keyword*: Recursive coding, VLSI, Application specific integrated circuits, digital signal processing chips , digital arithmetic.*

I. Introduction

Application specific design is a process of alternative design evaluations and refined implementations at various abstraction levels to provide efficient and cost effective solutions. In hand held and portable applications like cellular phones, modems, video phones and laptops, batteries contribute a significant fraction of its total volume and weight [1, 2]. These applications demand low power solutions to increase the battery life.Low power design of digital [1, 15] integrated circuits has emerged as a very active and rapidly developing field. Power estimation can be done at different stages of system design as illustrated in [3-16]. These estimations can drive the transformations to choose low power solutions. Transformations have done at the higher level of abstraction namely the system level have a greater impact, whereas estimations are generally more accurate at the lower levels of abstractions like the transistor level and the circuit level. Some researchers also presented a high level DSP design methodology raising the design entry to algorithmic and behavioral level instead of specifying a detailed architecture on register transfer level. They propose a design methodology using communication applications for performance analysis. Their research focuses on the development of methodologies and supporting tools to enable the modeling of future heterogeneous reconfigurable platform architectures and the mapping of applications onto these architectures. Recursive codes were presented in 1993, and since then these codes have received a lot of interest from the research community as they offer better performance than any of the other codes at very low signal to noise ratio. Advances in third generation (3G) systems and beyond will cause a tremendous pressure on the underlying implementation technologies. The main challenge is to implement these complex algorithms with strict power consumption constraint. Recursive coding is a forward error correction (FEC) scheme. Iterative decoding is the key feature of recursive codes [17, 18]. Recursive codes consist of concatenation of two convolution codes. Recursive codes give better performance at low SNRs (signal to noise ratio) [19, 20]. Interestingly, the name Recursive was given to these codes because of the cyclic feedback mechanism (as in Recursive machines) to the decoders in an iterative manner.Recursive codes with short delay are being heavily researched. Recursive codes generally outperform convolutional and block codes when interleavers exceed 200 bits in length [21]. The device integrates a 32x32-bit interleaver and performance is reportedly at least as good as with conventional concatenated codes using a Reed-Solomon outer code and convolutional inner code [22]. Power savings is important for towers in especially remote areas, where recursive codes are used. [23]. Recursive codes used by these two systems are very similar, the differences lie in the interleaving algorithm, the range of allowable input size and the rate of constituent RSC encoders [24].

There are three types of algorithms used in recursive decoder namely MAP, Max-Log-MAP and Log-MAP. theMax-Log-MAP algorithm in recursive decoder and for the design process of analog signal processing in digital domain and the process of filter designs approach using wavelet for such a analog data is proposed[25]. inturbocoding while processing filtration is observed.. The filtration logics are realized using MAC (multiply and accumulate) operation where a recursive addition, shifting and multiplication operation is

performed to evaluate the output coefficients [26].As the conventional methods utilize much time and power consumption. So in order to over come this problem in this paper a new scalable Repetitive multiply accumulates (MACs) is proposed.To care the above issues: in this work A system design of the recursive decoder module is described in Section II; and the filterdesigning of the conventional MAC in the filter banks is illustrated in section IIIand proposed recursive MAC approach is illustrated in Section IV. The hardware complexities, time and power consumption of the proposed Recursive MAC operation results are discussed in Section V Conclusions are placed in section VI and followed by reference in section VII.

II. System Design

As soon as the decoding commences, the encoded data information is demultiplexed and separated into the systematic received data (ys), parity data elements from the encoder1 (yp1) and the parity data values from the encoder2 (yp2). The systematic information is to be interleaved, since this interleaved data is one of the inputs to the decoder2 section. Essentially in the data supply unit modules the data inputs that are required by the decoder1 and the decoder2 are ready for usage.The main blocks in the decoder1 are the γ, α, β, llr, extrinsic unit, intermediate storage units, intermediate storage units and the associated feedback units, namely the extrinsic interleaver and its storage units. The storage units are modeled as FIFO/LIFO following the data access pattern analysis that was presented in the previous chapter.The main blocks in the decoder2 are the same as the decoder1, except for the different inputs at the various computational blocks. Another addition is the decision unit, which gives the final estimates of the message that is retrieved. The output of the decoder2 is stored to be fed back to the decoder1 in the next iteration. The typical algorithmic behavior of the recursive decoder requires a specific selector module to continue from second to the sixth iteration. When the encoded data is received at the input of the decoder, first it is multiplexed and then respective systematic, parity1, parity2 and interleaved systematic data are stored in the first iteration. During the second iteration till the sixth iteration only the decoder1 and the decoder2 are iteratively operating. Hence the selector module (multiplexer) during the first iteration takes the input signal from the data supply unit and gives a start signal to the first module of the decoder1 (unit). During the subsequent iterations, the selector module receives the input signal from the last computational unit of the decoder2 (extrinsic interleaver unit) and enables the unit.The top-level controller, whose purpose is to activate the modules in proper order, manages the recursive decoding process. The recursive decoder which constitutes of the two decoders is iterative in nature. The simulations have to be performed for six iterations. Hence a top-level finite state machine (FSM) controller unit is essential. A FSM consists of a set of states, a start state and a transition signal to move from the current state to the next state (in recursive decoder from one iteration to the next iteration). The FSM design appears explicitly in the recursive decoder design system for the control and the sequencing of the iterations. At the data flow level, iteration control and the data computation of the decoder system are separated then the description of the FSM has a close correspondence to the typical behavior of the algorithm.

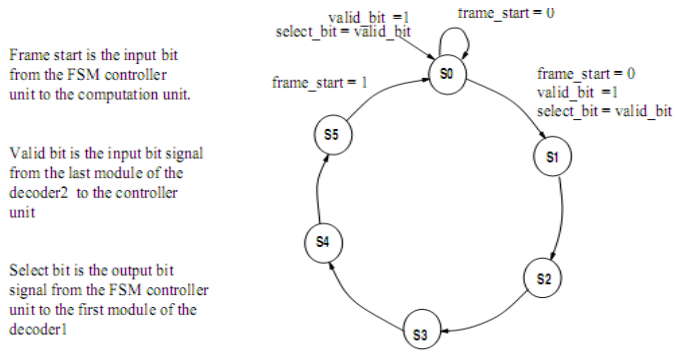


Fig.1: State diagram representation of FSM used in the recursive decoder design

Fig.1 depicts the state diagram of the FSM used in the recursive decoder design to control the number of iterations. The state diagram has state S0, which corresponds to the first iteration. After six iterations, next block of data has to be received and processed. If the frame start signal is equal to zero then the decoder process does not commence. If the frame start signal is equal to one and also the decoder module gets the enable signal, the decoding starts. The last module in the decoder2 gives a valid bit signal to the FSM controller unit. The FSM iteration controller unit sends an enable signal to the decoder1 so that the next iteration starts. In this way the number of iterations is controlled using the FSM unit. The recursive decoder requires a number of memory modules in its design. This is due to the filtration operation overhead.

III. Filter designing

For many signals, the low-frequency content is most important part. It gives the signal its identity. The high-frequency content, on the other hand, provides the add on properties to it. Eg.in a human voice if high-frequency components are removed, the voice sounds different, but, if enough of the low-frequency components are removed the complete signal may be lost. In wavelet analysis, decomposition tree can be obtained by the implementation of a chain of High pass and low pass filter banks, which consist of three banks of high pass and low pass filters. The filters decompose the input signal samples into different subbands depending upon the filter coefficients passed to it.The filter logics are realized using MAC (multiply and accumulate) operation where a recursive addition, shifting and multiplication operation is performed to evaluate the output coefficients. The recursive operation logic is as shown below.

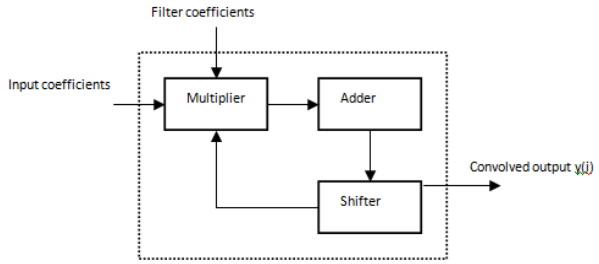


Fig 2: realization of recursive MAC operation

Before passing the data to filter bank the fifo logic realized stores the data in asynchronous mode of operation, 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.

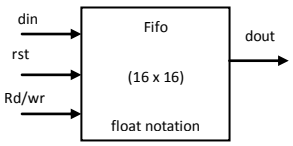


Fig 3: realization of 16 x 16 fifo logic for coefficient interface

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.

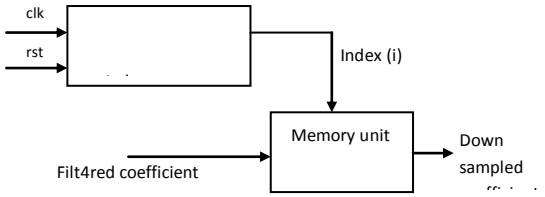


Fig 4: architecture for decimation by 2 logic

The conventional 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. for the realization of this MAC operation a recursive multiplication and addition operation is carried out. This recursion results in multiple accumulation operation in successive clocks of duration amounting to that of an addition-time, so that the accumulation of N successive input words are performed in Nclock cycles. For large values of N, a computational latency of Naddition-time over head and high power consumption and slows the accumulation process to obtain an accumulated output could be too high to meet the timing requirement in real-time application.so in order to over this problem in this paper a a new scalable Repetitive multiply accumulates (MACs) is proposed which is brifl;y discussed in section IV.

IV. Modified MAC operator

The proposed structure-1, as shown in Fig. 5(a), consists of two stages. The first stage is implemented by a counter module consisting of eight 3-bit ripple counters C3 [shown in Fig. 5(b)]. The second stage is implemented by an adder module consisting of a shift-add (SA)-tree, as shown in Fig. 5(a).

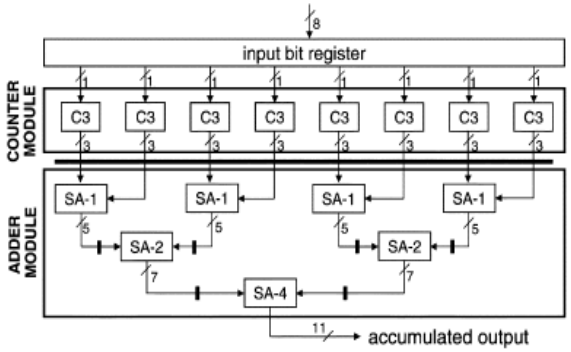


Fig. 5 Proposed Recursive MAC structure (a) Accumulator structure. (b) Function of three-bit counter C3

The time period of the clock used for the counter module is the same as the worst case propagation delay of a T flip-flop . On the first clock cycle, all the flip-flops of all the counters are reset and each counter is fed with a bit of input word available from the input register. After 7 clock cycles, the states of flip-flops of the counter represent the intermediate operand for (which equals to the number of 1's on the bit stream of a given place value of all the seven inputwords). Outputs of the counters are latched to four SA-1 cells at the first level of SA-tree. Each SA-1 shifts its vertical-input by one location to left; and adds its right-input with the left-shifted word. The second level of SA-tree consists of two SA-2 which shifts its left-input by two locations to left and adds that to its right-input. The last/third level of SA-tree consists of one SA-4 which shifts its left-input by four locations to left and adds that to the right-input

The SA-tree has a combinational path of duration$T_1 = T_{A3} + T_{A5} + T_{A7}$ in non-pipelined form to add the eight 3-bit outputs of counter module, where$T_{A3}, T_{A5}$and$T_{A7}$are, respectively, the time required for 3-bit, 5-bit and 7-bit additions. Total accumulation-time of the structure for a non-pipelined implementation amounts to the sum of the delays of counter module and the adder module$T_c + T_1$, , where$T_c = N \cdot T_{TF}$ is the time involved in counter module. The counter module and the adder module could also be implemented in two separate pipelined stages, with a pipeline period$= max\{T_c, T_1\}$. Duration of T_c in general, depends on the number of input word$N$, while T_1 depends on N as well as the word-length .For large values of$L$ the depth of SA-tree becomes large and word-length grows as we go down on the SA-tree. The computation of adder module, therefore, becomes larger for larger values of . The adder module in that case could be implemented by a pipelined SA-tree to increase the throughput rate. Similarly, when$N$ is large compared with$L$, the bit-level accumulation-time in counter module becomes bigger than the adder module delay. For very large values of$N$, it should be preferred to preform the accumulation in multiple pipelined stages of the proposed accumulators. In any case, the clock period of adder module, however, could be made an integer multiple of the clock period of counter module for efficient derivation of clock signals.

V. Simulation Observation

The design is first modeled in VHDL and then synthesized using Leonardo tools. XPower brings an extra-level of design assurance to the low-power device analysis. Accurate power estimation during programmable design is done with XPower. XPower reads in either pre-routed or post-routed design data and provides accurate device power estimation either by net, or for the overall device. Results are provided either in report or graphic format. The VHDL designs were simulated and only synthesized. The estimates are for 0.18 m technology. The voltage that is used in our estimation is 2.5V. There is a facility to set the input capacitance and frequency of operation for which the design power estimation is to be done. There is a provision to give a specific switching activity for the design power estimation. The estimates for 20 and 50 activity factor is found. XPower calculates the power as a summation of the power consumed by each component in the design. The power consumed is the product of capacitance, square of the voltage, activity factor and frequency of operation and is reported in mW.

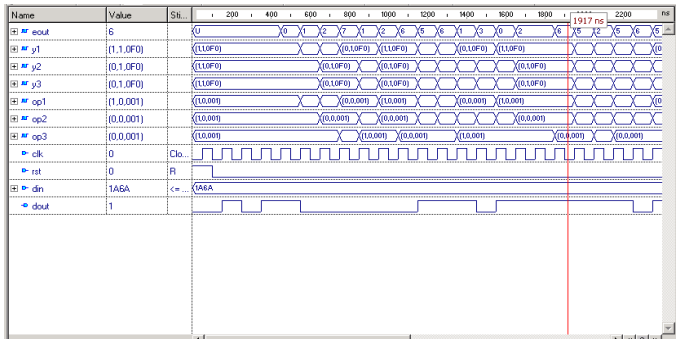


Fig.6: simulation observation illustrating timing result for the developed encoding unit

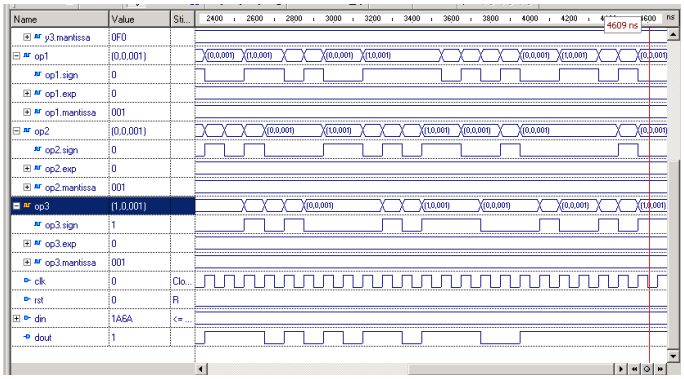


Fig.7: simulation timing observation for the decoding logic developed for the implemented system.

For the realization of the developed logic and its evaluation in physical environment in this work the developed system is synthesized on Xilinx ISE synthesizer targeting on to virtex2p device part number 2vp100ff1696-6. The obtained parameter for this device is given below,

Design Statistics: # IOs : 19

Cell Usage : # BELS : 6549

Maximum Frequency: 141.012MHz

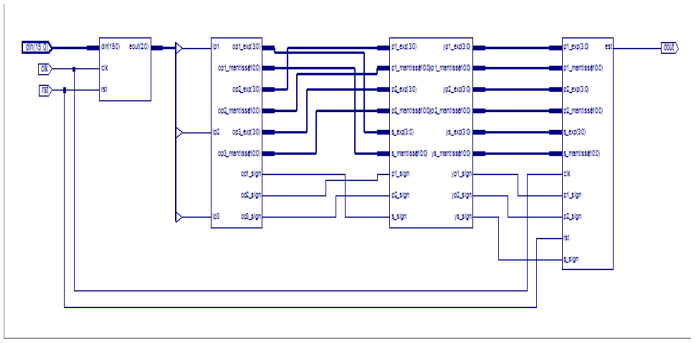


Fig.8: obtained RTL realization for the encoding unit

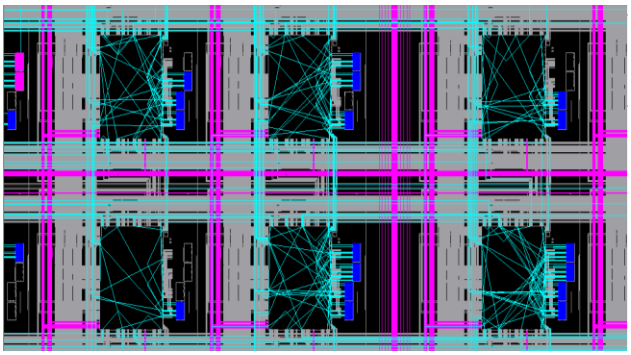


Fig. 9. the routed logical operation for the developed system onto the targeted FPGA device.

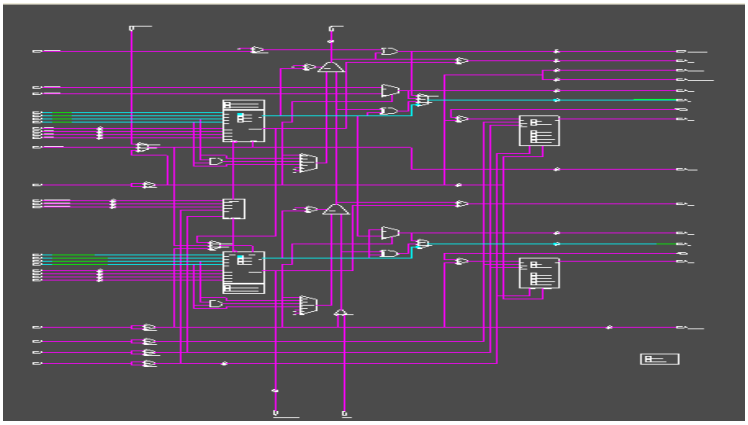


Fig.10. Logical placement of the logical blocks on a targeted FPGA logicFor the evaluation of the suggested design methodology a analog signal is taken and processed, the observations obtained are as illustrated below,

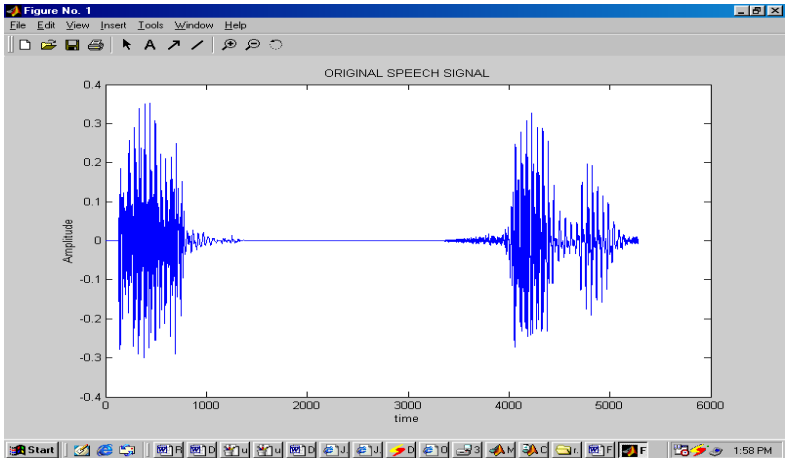
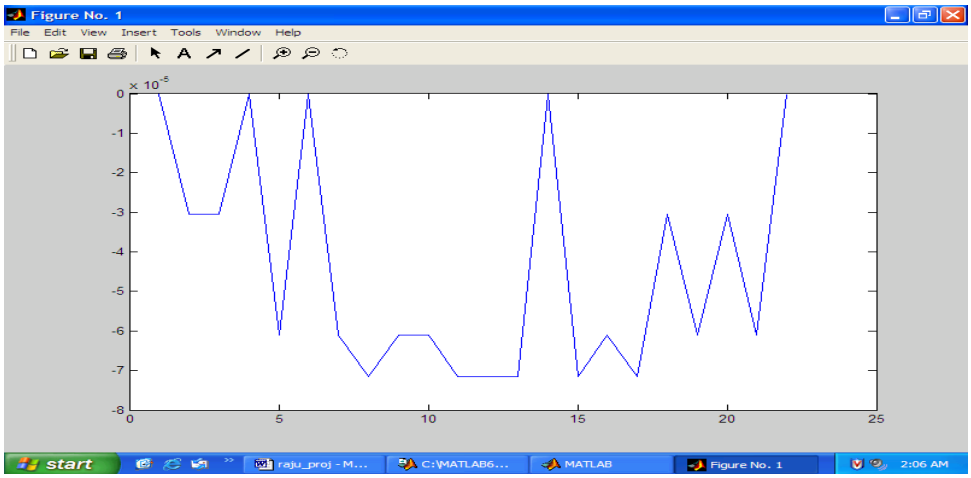


Fig.11Tst signal sample for observation2

Samples Considered

0---------- - "0000000000000000"
-3.0518e-05--------"1000110000110101"
-3.0518e-05--------"1000110000110101"
0---------- - "0000000000000000"
-6.1035e-05-------"1000010000110110"
0---------- - "0000000000000000"
-6.1035e-05-------"1000010000110110"
-7.1553e-05-------"1000011001001111"
-6.1035e-05-------"1000010000110110"
-6.1035e-05-------"1000010000110110"
-7.1553e-05-------"1000011001001111"
-7.1553e-05-------"1000011001001111"
-7.1553e-05-------"1000011001001111"
0---------- - "0000000000000000"
-7.1553e-05-------"1000011001001111"
-6.1035e-05------"1000010000110110"
-7.1553e-05-------"1000011001001111"
-3.0518e-05------"1000110000110101"
-6.1035e-05-------"1000010000110110"
-3.0518e-05------"1000110000110101"
-6.1035e-05-------"1000010000110110"
0---------- "0000000000000000"



Plot for the considered input signal sample

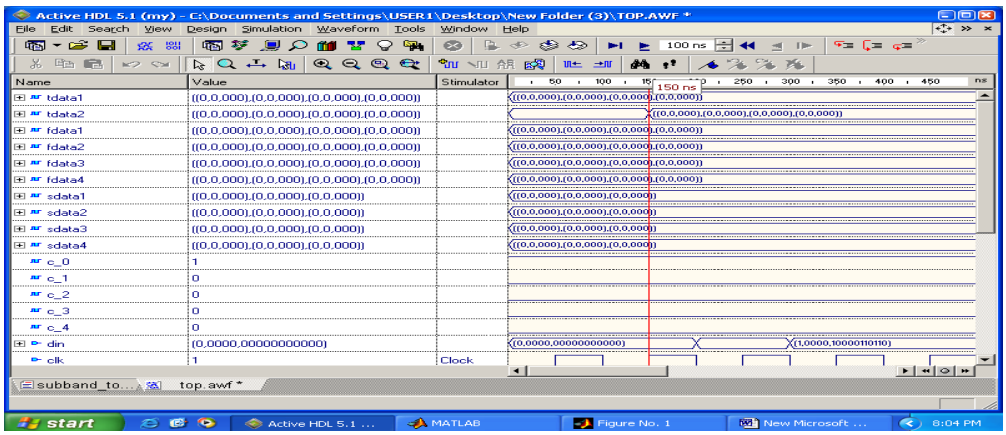


Fig.12 Simulation result for the implemented subband design

Figure shows the simulation result obtained for the implemented design on consideration to the signal samples considered as shown above. The simulation result shows signal 'tdata' as the temporary signal used to pass the data from input buffer to the filter bank. Signal 'fdata' carries the data from filter bank to each filter where the input data get convolves with the filter coefficients passed from the package as given above. Control signal 'c_0','c_1', 'c_2','c_3','c_4' are the control signal generated from the controller unit for the controlling of the filter bank operation. Signal 'din' shows the input data fed to the module. The signal is defined as a record type with 1st bit as sign bit next 4 bit as exponent bits followed by 11 bits as mantissa bits. Signal 'clk' shows the system clock fed to the module for the synchronization operation. The signal clock is passed to each filter bank for the synchronous operation

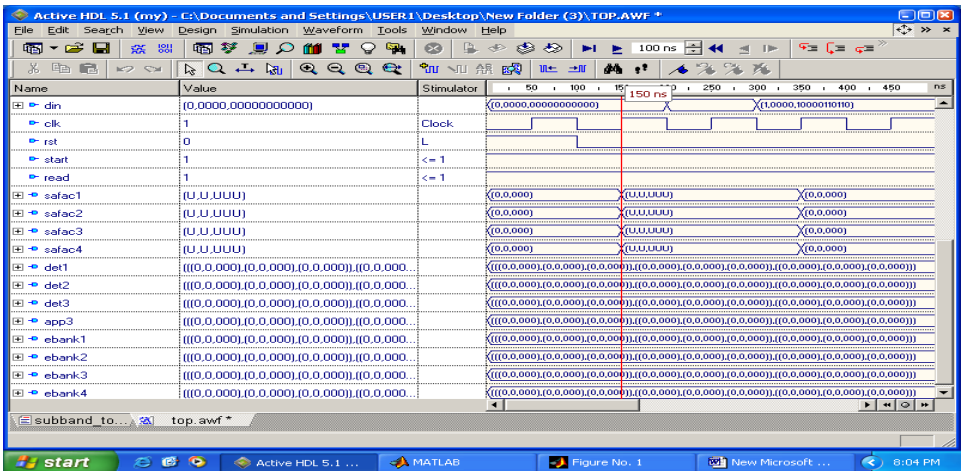


Fig. 12 Simulation result for the implemented subband design (cont.fig 11)

The simulation result shown above shows the global signal reset 'rst' passed to the system. The system is considered to be active low with the system getting activated in the lower value of the reset signal. On the initialization of the system the system get reset by applying reset as high on the first clock pulse. Under reset condition all the signal get cleared. Control signal 'start' is passed to the system as enable signal making the system enable whenever the signal goes high.
Control signal 'read' is used for reading of the data stored into input-buffer. On the rising value of read signal the content of the input buffer is read and passed down to filter bank for further processing. Signal 'sfac1', 'sfac2', 'sfac3', 'sfac4' shows the scale-factors obtained for the sub-band samples. Signal 'ebank1', 'ebank2', 'ebank3', 'ebank4' gives the energy content of each sub-band sample obtained after the decomposition. The sub-band sample energy gives the energy spectral of the sub-band samples.

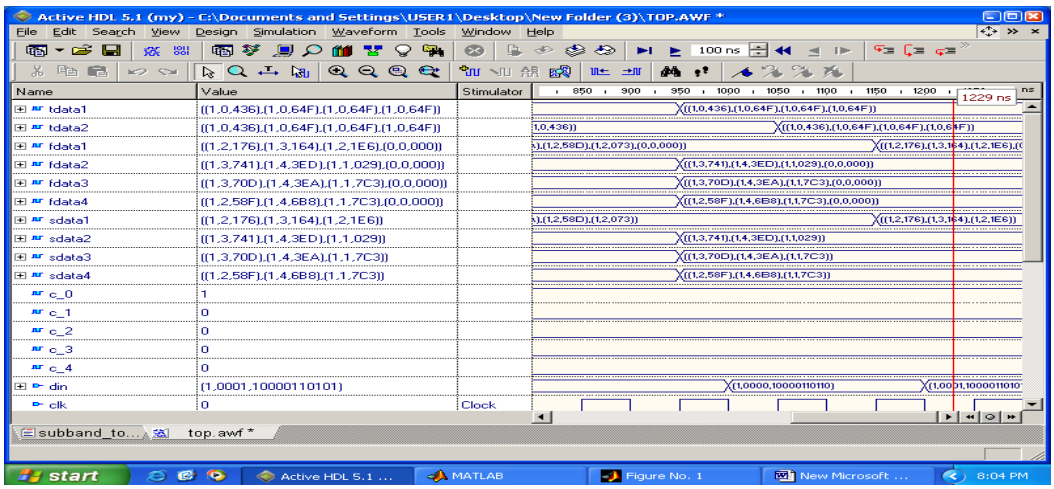


Fig.13 Simulation result for the implemented subband design (cont.fig 12)

Figure shows the simulation result for the implemented design under processing. The figure shows the signal values carried from the input buffer to the filter input via the signal tdata. The proposed design decomposes the signal into four distinct subband thus four unique temporary signals are used for the transfer of data to the filter bank. Signal fdata shows the data transferred to filter module for the processing. Sdata shows the sub-sampled data obtained for each sub-band.

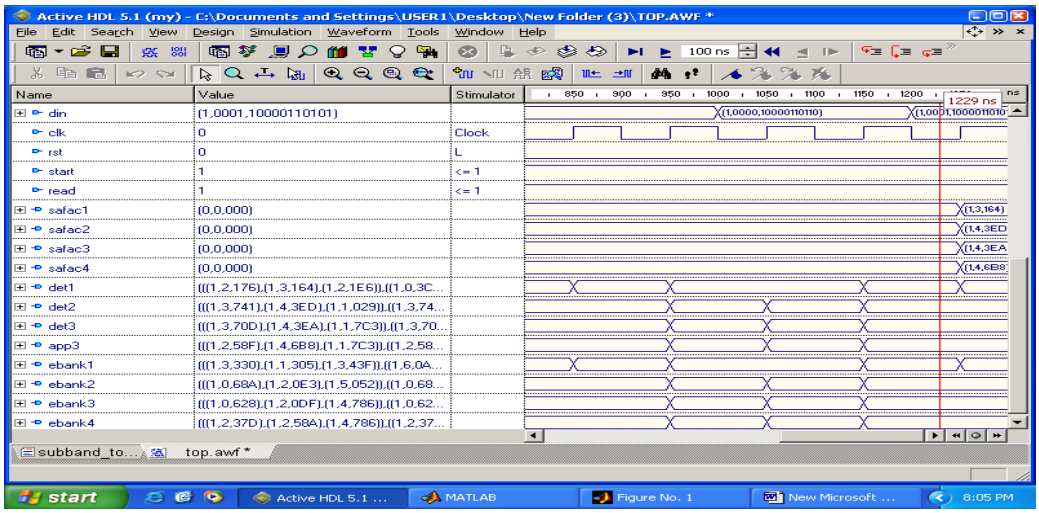


Fig.14 Simulation result for the implemented subband design (cont.fig 13)

Figure 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 ebank's gives the energy values of each subband sample for each subband

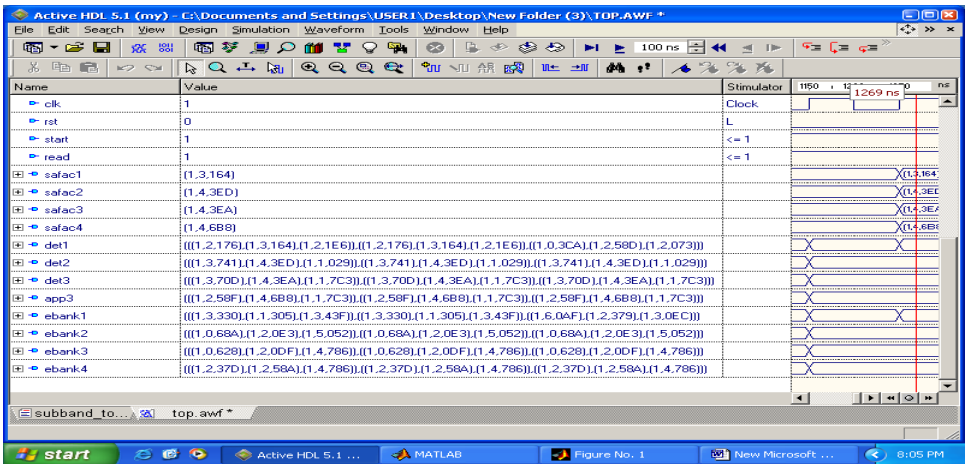
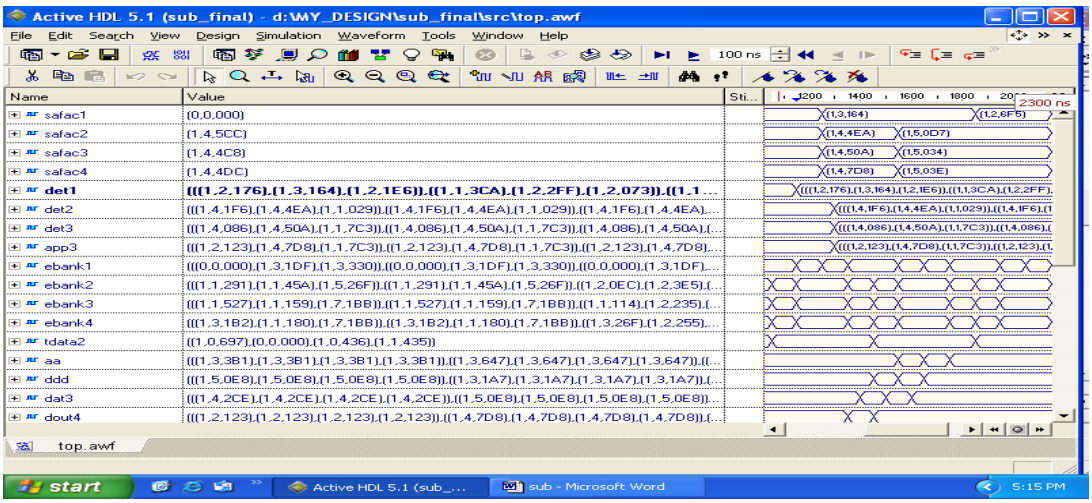


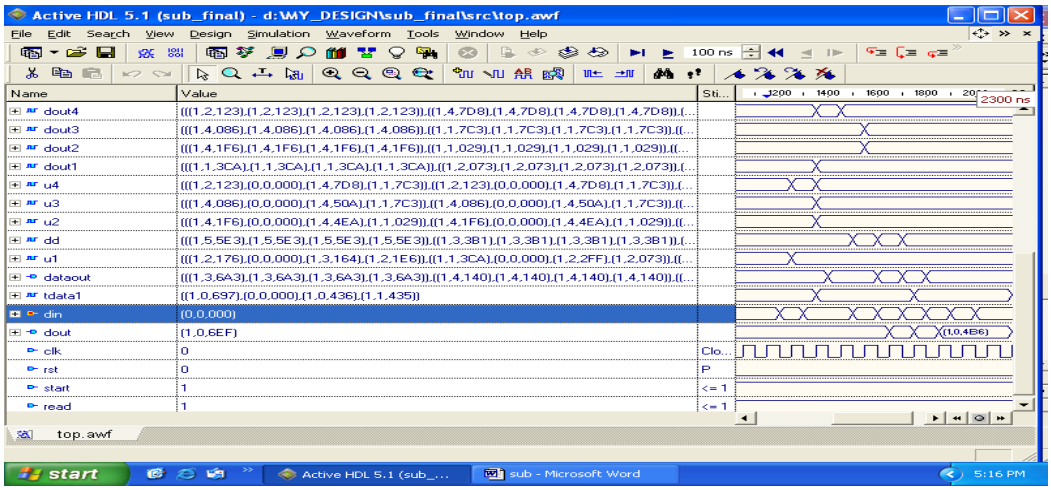
Fig.15 Simulation result for the implemented subband design (cont.fig14)

Figure 5.12 shows the overall scale factors, detail coefficients and approximate coefficients for every packet of the signal samples. The scale factors obtained show the maximum values of the sub-band samples obtained under each subband. The energy for the samples are calculated as E=square (Magnitude of each sample).

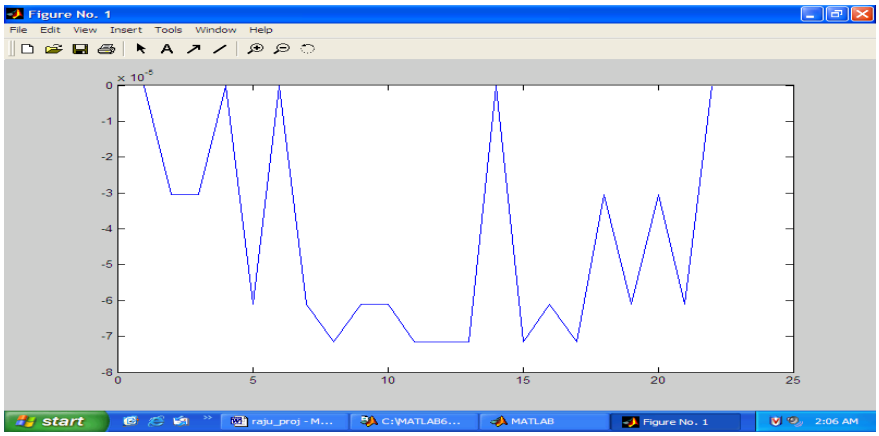
The elements of the detailed coefficient matrix (det1) show the samples lying in the higher frequency range from 8-4KHz. The second sub-band shown by (det2) gives the coefficients lying in the range of 4-2 KHz ranges. Det3 gives the sub-samples with a frequency of 2-1KHz and app3 matrix shows the approximate coefficients lying in the range of 1-0KHz ranges

Simulation Result For The Reconstructer Module





comparison for the input and output results obtained



detailed coefficients and approximate coefficient for the input signalsignal. Each sub-band constitute of 9 sub-samples for every packet of the data burst. The ebank's gives the energy values of each subband sample for each subband.

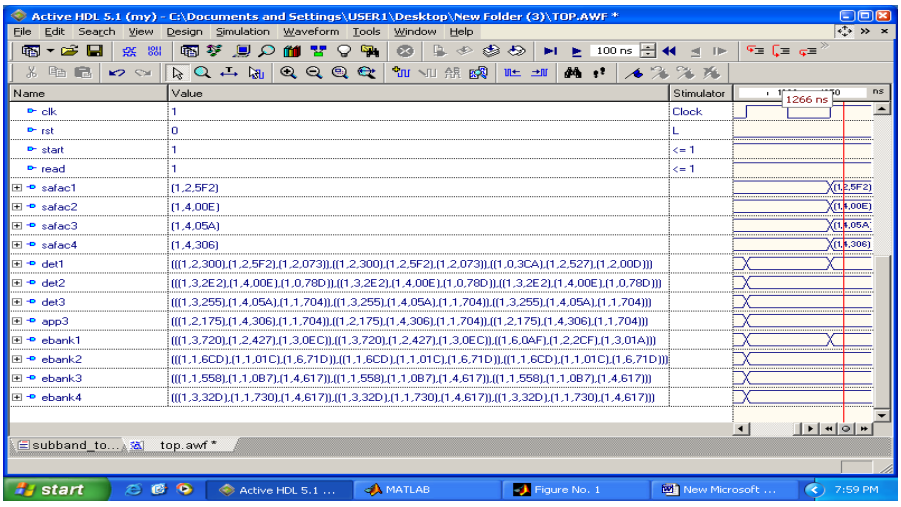
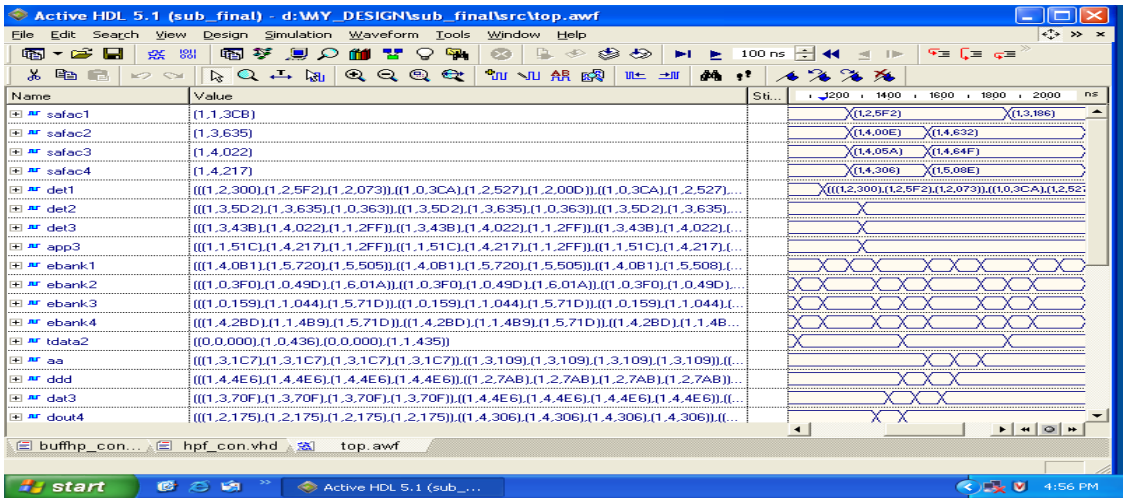
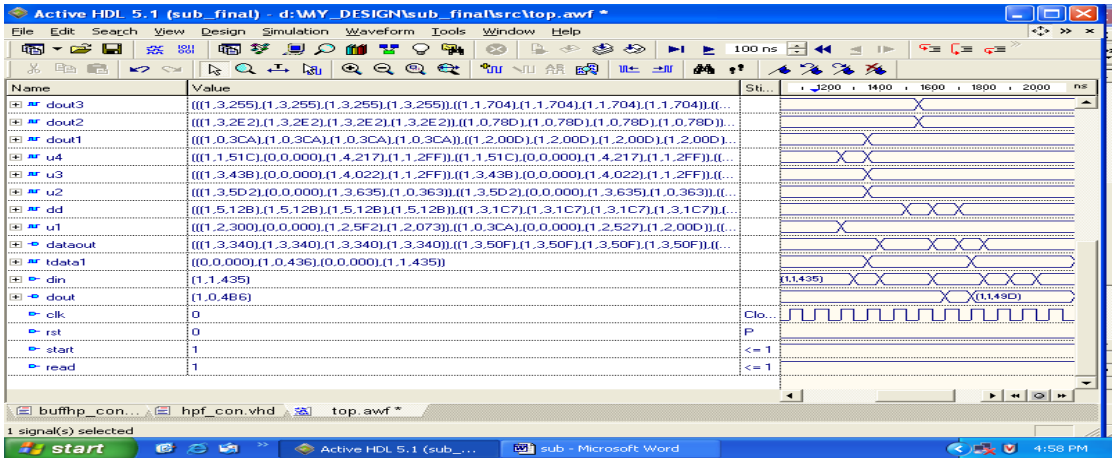


Fig.16 Simulation result for the implemented subband design

Fig. 16 shows the overall scale factors, detail coefficients and approximate coefficients for every packet of the signal samples. The scale factors obtained show the maximum values of the sub-band samples obtained under each subband. The energy for the samples are calculated as E=square (Magnitude of each sample).
The elements of the detailed coefficient matrix (det1) show the samples lying in the higher frequency range from 8-4KHz. The second sub-band shown by (det2) gives the coefficients lying in the range of 4-2KHz ranges.

Det3 gives the sub-samples with a frequency of 2-1KHz and app3 matrix shows the approximate coefficients lying in the range of 1-0KHz ranges.





logical routing of proposed design targeting to Xc2s50e-ft256-7

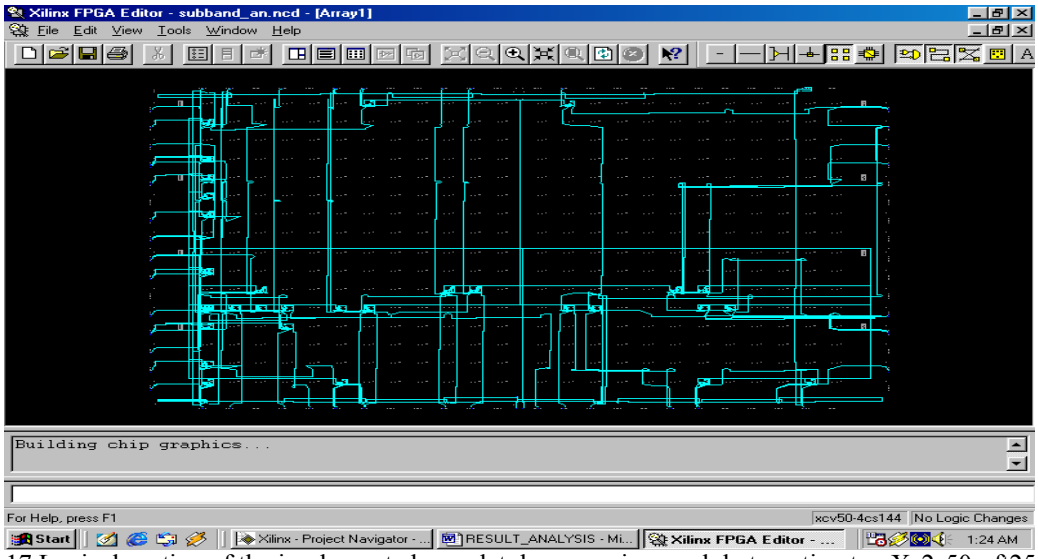


Fig.17 Logical routing of the implemented wavelet decomposing module targeting to Xc2s50e-ft256-7

Figure shows the logical routing of the implemented design targeting on to Xc2s50e-ft256-7 FPGA of virtex 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.

logical placement of proposed design targeting to Xc2s50e-ft256-7

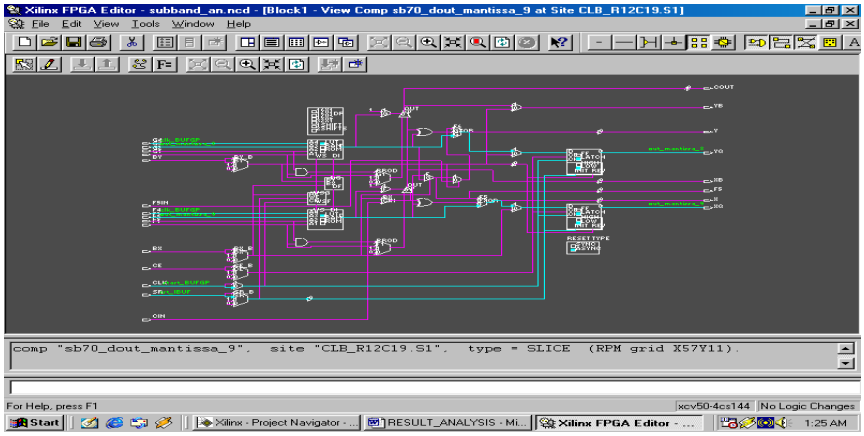


Fig.17 logical placement of the implemented design in the FPGA (xc2s50e-ft256-7)

Figure shows the logical placement of the implemented design on to the targeted FPGA (Xc2s50e-ft256-7). The figure shows the logical resources used by the Logic implemented for the implemented design for one CLB. The basic elements used for the implementation were LUT and Buffer elements as shown in figure.
floorplanning of proposed design targeting to Xc2s50e-ft256-7

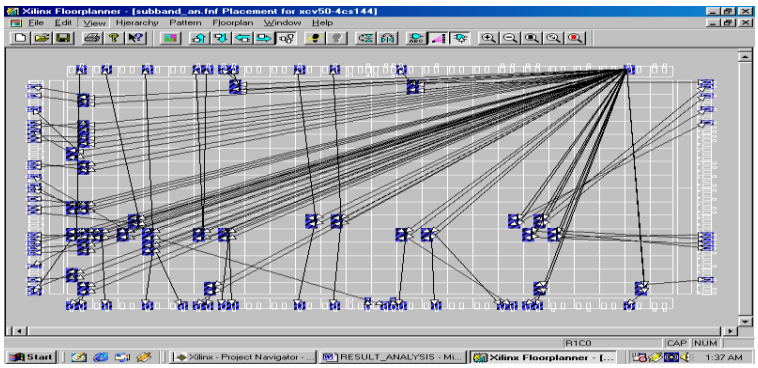


Fig.18 Floor planning of the implemented design in the targeted FPGA(Xc2s50e-ft256-7)

Figure shows the floor planning of the implemented design on to the targeted FPGA (Xc2s50e-ft256-7). The figure shows the logical net connection between the two CLB and the IO buffer used. The interconnects obtained shows the path covered for the logical mapping of the resources for data transfer for the operation.
packageveiw of proposed design targeting toXc2s50e-ft256-7

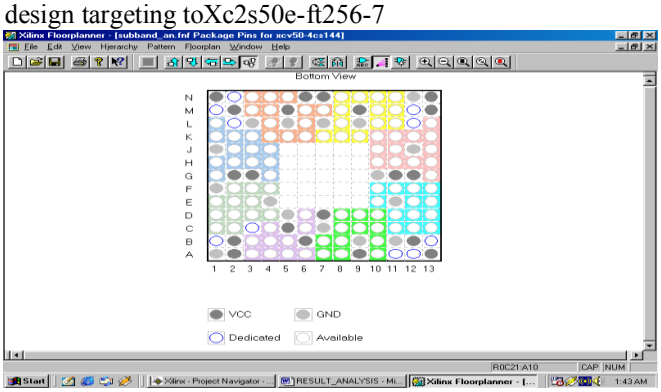


Fig 5.22 Bottom Package View for the Target FPGA(Xc2s50e-ft256-7)

Figure shows the bottom view of the targeted Virtex FPGA ((Xc2s50e-ft256-7) this package view is consisting of the power supply pins (VCC), the ground (GND) the dedicated lines used for data transfer and available pins for other applications.

Minimum period: 2.649ns

(Maximum Frequency: 377.501MHz)

Minimum input arrival time before clock: 9.656ns

Maximum output required time after clock: 6.366ns
Design Statistics
IOs : 105
Cell Usage :
BELS : 205

VI. Conclusion

Anew scalable Repetitive multiply accumulates (MACs) is proposed An efficient algorithm of MAC of multiple accumulations, where$N$ inputwords of L-bit size are transformed either into number of L-bitwords or number of $M = \lceil \log_{2(} N + 1) \rceil$ -bitwords, or M number of L -bitwordswhich could be shift-accumulated to generate the desired accumulated sum. The proposed algorithm, offer low power consumption and less time complexity when when compared to conventional methods. The proposed MAC structure are found to involve more area compared with conventional but provides low consumption and less computation time when compared to the conventional MAC in the filtration process of the recursive coding system.The proposed structurewould have potential advantages in many DSP applications forimplementation of FIR and IIR filters, adaptive filters and computationof sinusoidal transforms, etc. It would also be highlyuseful for implementation of matrix-vector multiplication anddecomposed-DA-based computation of inner-products whererepetitive multiple accumulation has an important role

Reference

[1] Kuroda, T.; Hamada, M. "Low-powerCMOS digital design with dual embedded adaptive power supplies " IEEE journal of solid-state circuits, vol. 35, no. 4, april 2000.
[2] Jan Rabaey,Jan M. Rabaey, AnanthaChandrakasan, BorivojeNikolic "Digital Integrated Circuits" (2nd Edition) Prentice Hall; 2 edition, 2003.
[3] T. Sato, M. Nagamatsu and H. Tago, "Orion: A Power-Performance Simulator for Interconnection Networks," IEEE Symposium on Low Power Electronics, San Diego, CA, USA, October 1994, pp. 46-47
[4] C. Piguet et al., "Low Power Design of 8-b Embedded CoolRISC Microcontroller Cores, "IEEE Journal of Solid-State Circuits, vol. 32, no. 7, July 1997, pp.1067-1078.
[5] Naehjuck Chang, Kuranho Kim and H. G. Lee, "Cycle accurate Energy consumption measurement and analysis: Case study ARM7 TDM1," International Symposium on Low Power Electronics and Design, Italy, July 2000, pp. 185-190.
[6] Yao Guo1 , SaurabhChheda , Israel Koren1 , C. Mani Krishna , and CsabaAndras Moritz, "Energy characterization of hardware-based data prefetching ,"IEEE international conference ,2004.
[7] Minana, G.; Hidalgo, J.I.; Lanchares, J.; Colmenar, J.M.; Garnica, O.; Lopez, S., "Reducing power of functional units in high-performance processors by checking instruction codes and resizing adders IET Comput. Digit. Tech., 2007, 1, (2), pp. 113–119
[8] Allen C. Cheng,Gary S. Tyson, "An energy efficient instruction set synthesis framework for low power embedded system designs ," IEEE transactions on computers, vol. 54, no. 6, june 2005
[9] M.W. Allamand M.I. Elmasry, ": Dynamic current mode logic (DyCML), a new low-power high-performance logic family," IEEE Custom Integrated Circuits Conference 2000**.**
[10] Laszlo Varga, Ferenc Kovacs, Gabor Hossni "An Efficient Adiabatic Charge-Recovery Logic," IEEE Transaction 2001..
[11] H. Mehta, R.M. Owens, M. J. Irvin, R. Chen and D. Ghosh, "Techniques for low energy software," International Symposium on Low Power Electronics and Design, Aug. 1997, pp. 72-75.
[12] Guangyi Zhang, Kaihua Liu, Zhu Wang, FangxiHou, JichangGuo "DSP-Based Signal Processing For Differential Frequency Hopping Communications System" IEEE Transaction for Electrical and computer Engneering 2003.
[13] Soontae Kim, N. Vijaykrishnan, M. J. Irwin and L. K. John, "On Load Latency in Low-Power Caches," ISLPED'03, August 25–27, 2003, Seoul, Korea.ACM 2003.
[14] Chanda, M.; Dandapat, A.; Rahaman, H., "Low-power sequentialcircuit using single phase Adiabatic Dynamic Logic," IEEE International Conference on Computers and Devices for Communication 2009.
[15] Duong Tran, Kyung Ki Kim, Yong-Bin Kim, "Power Estimation in Digital CMOS VLSI Chips" IEEE conference on Instrumentation and Measurement Technology Conference Ottawa, C anada, 2005.
[16] Ravi, S.; Raghunathan, A.; Chakradhar, S, "Efficient RTL power estimation for large designs," IEEE 16th International Conference , 2003 .
[17] Burnashev, M.; Dumer, I., "Error exponents for recursivedecoding of Reed-Muller codes ," IEEE Conference publication 2006.
[18] Liu, D.N.; Fitz, M.P., "Low complexity linear MMSE detector with recursive update algorithm for iterativedetection-decoding MIMO OFDM system IEEE Conference Publications 2006
[19] Chatzigeorgiou, I.; Wassell, I.J., "Revisiting the calculation of effectivefreedistance of turbo codes of recursive codes," Electronics Letters Volume: 44 , Issue: 1 2008.
[20] Tranter, W.; Taylor, D.; Ziemer, R.; Maxemchuk, N.; Mark, J., "Near Optimum Error Correcting Coding and Decoding: TurboCodes "Wiley-IEEE Press 2007
[21] Bee Leong Yeap, Tong HooiLiew, Hamorsky, J. Hanzo, L "Comparative study of turbo equalization schemes using convolutional, convolutional turbo, andblock-turbo codes" Wireless Communications, IEEE Transactions on Volume: 1 , Issue: 2 2002
[22] Valadon, C.G.F.; Tafazolli, R.; Evans, B.G. "Performance evaluation of concatenated codes with inner trellis codesandouterReed-Solomoncode" IEEE Transactions rolume: 49 , Issue: 4 2001.
[23] Qingchun Chen; Wai Ho Mow; Pingzhi Fan "Some New Results on Recursive Convolutional Codes and Their Applications " IEEE 2006.
[24] Liao, Xin; Ilow, Jacek; Al-Shaikhi, Al "Trellis Termination in Turbo Codes with Full Feedback RSCEncoders" 6th International ITG-Conference on Source and Channel Coding (TURBOCODING), IEEE 2006.
[25] RajendraPrasad "Analog Modeling of Recursive Estimator Design with Filter Design Model"
[26] Rajendraprasad" Design approach to analog signal processing in digital domain using HDL modeling"