RISC IMPLEMENTATION OF OPTIMAL PROGRAMMABLE DIGITAL IIR FILTER

Miss. Sushma kumari
IES COLLEGE OF ENGINEERING, BHOPAL
MADHYA PRADESH

Mr. Ashish Raghuwanshi(Assist. Prof.)
IES COLLEGE OF ENGINEERING, BHOPAL
MADHYA PRADESH

Abstract:
This paper is based on the implementation of Reduced Instruction set computer with the application of Discrete Cosine transform (DCT), Inverse DCT, Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT), Digital filter are performed by DSP system. Digital filter is one of the important contents of digital signal process. The performance of the processor design is improved by using the pipeline approach. It allows the processor to work on different steps of the instruction at the same time, thus more instruction can be execute in a shorter period of time. The analysis of this processor will provide various features including arithmetic operations. The speed of operation is mainly affected by the computational complexity due to multipliers and adder modules of the digital systems. Our work will targets the computer architecture courses and presents an FPGA (Field Programmable Gate Array) implementation of a MIPS (Microprocessor without Interlocked Pipeline Stages) via VHDL (Very high speed integrated circuit Hardware Description Language) design. The latency and computational time is utmost important in microprocessor. Thus we design the multiplier and adder module with improve latency and computational time.

Key Words: RISC, IIR, FPGA, DSP Processor, Direct Form I, Direct Form II, VHDL, Parallel, Cascade Structure

Introduction:
Conventionally, the digital filter, Discrete Fourier and Z transform applications are implemented either using general purpose DSP processors (low speed, less expensive, flexible) or using Application Specific Integrated Circuits (ASIC) which offer high speed but are expensive and less flexible. An alternate approach is to use reduce instruction set computer (RISC) as they provide solutions that maintain both the advantages of the approach based on DSP processors and the approach based on microprocessor. The RISC decodes the instruction format and execute one instruction per cycle. By pipeline approach it fetches, decoded, and execute two or three instructions at the time. Instructions are of fixed number of bytes and take fixed amount of time for execution with lesser amount of circuitry. Minimisation of hardware resources by utilisation of symmetrical coefficient FIR filter properties will reduce combinational logic signal path with balanced adder trees and pipelining. Pipelining balance the delay balancing needed in high data rate applications. It also reuses multiplication results in Direct cascade and parallel form IIR structure.

Principles of IIR digital filter:
Digital filter is actually a linear time-invariant discrete system using finite precision algorithms, and its function is actually achieved by a large number of addition and multiplication operations [4]. IIR digital

ISSN: 2278 – 909X

filters are recursive systems that involve fewer design parameters, less memory requirements, and lower computational complexity than finite impulse response (FIR) digital filters. It is characterized by the general linear constant-coefficient difference equation as follows:

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

Transforming this difference equation into the z-domain by means of the z-transform, such a class of linear time-invariant discrete-time systems is also characterized by the transfer function as follows:

$$H_k(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

Different structures of IIR filters are described by the difference equation in above. These structures are referred to as direct-form realizations. It should be noted that although these structures are different from one another by design, they are all functionally equivalent. Three prominent direct-form realizations are the Direct-Form I, the Direct-Form II, and the Transposed Direct-Form II structures. In terms of hardware implementation, the Direct-Form I structure requires M+ N+ 1 multiplication, M+N additions, and M+N+1 memory locations.

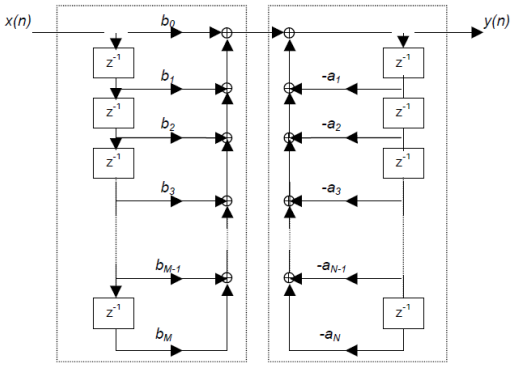


Fig 1 Direct form I realization

The Direct-Form II structures require M+N+1 multiplications, M+N additions, and the maximum of {M,N} memory locations. Because the Direct-Form II structure requires less memory locations than the Direct- Form I structure, it is referred to as being canonical. Figure above shows an IIR digital filter in Direct-Form II format [4].

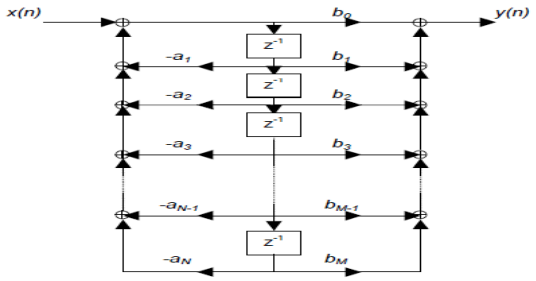


Fig 2 Direct form II realization

Implementation of Digital filters:

The aim of digital filter design is to removing noise, shaping spectrum, and minimizing inter-symbol interference in communication architectures and is use to find a suitable structure which passes requirements and minimizes numerical and quantization errors. There are few types of realizations and their combinations which can be used for implementation:

1. **Universal processor**: There operation is usually processed in a few clock cycles in universal processors so the computation performance is low.
2. **Digital signal processor** (DSP processor) with instruction set optimized for signal processing.
3. Special hardware such as **Field programmable gate array (FPGA)** or Application-specific integrated circuit (ASIC).
4. **Reduce Instruction Set Computer (RISC)**

These realizations are described below and the advantages and disadvantages are summarized in Table below.

ISSN: 2278 – 909X

Table I: Comparative analysis of our work with base papers.

Type of Implementation	Advantages	Disadvantages
Universal processor	low cost	Low Performance
DSP processor	Speed increases as the number of calculation cycle decreases, simple implementation	but architecture limitations Area depends on the number of calculation cycles selected
FPGA	highest performance	more complicated implementation, higher power consumption
RISC	Reduce Instructions Computation of one output sample in one clock cycle. The fastest architecture	Though the RISC has less instruction set, as its the bit processing size increases then the test pattern becomes complicated and the structural faults are maintained high.

RISC:

Reduce instruction set computer (RISC) is a microprocessor with few instruction set with simple addressing modes that takes same amount of time to execute. Thus it has the faster architecture. It has superscalar and pipelining architecture. The module design of RISC includes the Register, Instruction Memory, Data Memory, instruction fetch unit, instruction decode unit, the control unit, and execution unit. Instruction unit fetches the instruction code as per the address pointed by program counter register. It includes the incrementer logic to increment the content of program counter after every instruction byte execution. The instruction decoder decodes the instruction provided from the instruction fetch unit. The control unit generates the control signals for the operation of the arithmetic and logical operations. The memory read and write signals are asserted for the control unit is use to access the data or write the data in memory.

Implementation tool Xilinx and VHDL:

VHDL (Very High Speed Integrated Hardware Description Language) is a very powerful, high-level, concurrent programming language. At the implementation level we can build structural models using component instantiation statements that connect and invoke subcomponents. The VHDL generate statement provides ease of block replication and control. A dataflow level of description offers a combination of the behavioral and structural levels of description.

Methodology for IIR implementation in RISC

First of all, we convert the model design files into corresponding VHDL design files using Signal Compiler, compile the .vhd files under Xilinx integrated environment, and conduct timing simulation to simulate and verify the design files.

(1) Convert the digital filter specifications into an Binary representations.

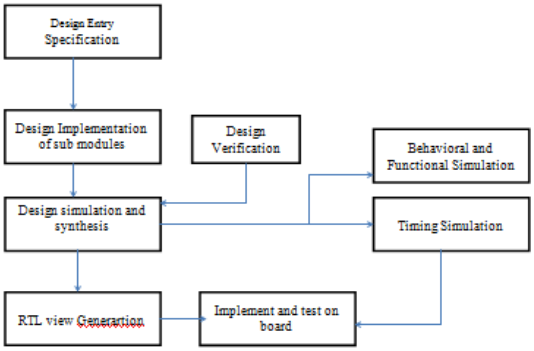
(2) Extract the filter coefficients.

(3) We use a parallel-processing filter implementation, in which parallel IIR paths are formed,

ISSN: 2278 – 909X

4) We use multiplier and/or adder, in parallel approach which is faster.
(5) Pipelining, a standard feature in RISC processors, is a technique used to improve both clock speed and overall performance.
(6) Pipelining allows a processor to work on different steps of the instruction at the same time, thus more instruction can be executed in a shorter period of time.
(7) When our design processor is pipelined, during a single clock cycle each one of design modules or stages is in use at exactly the same time executing on different instructions in parallel.

Design Flow diagram:



Modules Design and its simulation:

The processor implementation consists of two main types of logic elements: combinational and sequential elements. Combinational elements are elements that operate on data values, meaning that their outputs depend on the current status of inputs. Such elements in the processor implementation include the arithmetic logic unit (ALU) and memory. Sequential elements are elements that depend on current status of input and previous status of output i.e. it acts as latch. Each state element has at least two inputs and one output. The two inputs are the data value to be written and a clock signal. The output signal provides the data values that were written in an earlier clock cycle. Logic units are design and implement in VHDL.

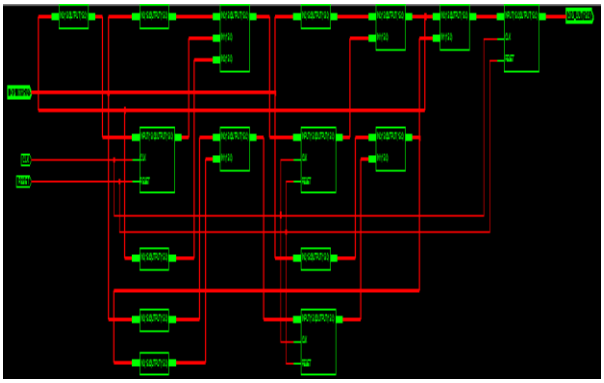


Fig 3 RTL view of Parallel structure IIR filter

The fig 2 shows the RTL view of the parallel structure of IIR filter. Parallel-form realization of an IIR digital filter can be obtained by performing a partial-fraction expansion on the transfer function H(z). Performing this mathematical function produces the resulting transfer function in the form as follows:

$$\mathbf{H(z) = C + \sum_{k=1}^{K} H_k(z)}$$

The function Hk(z) is in 2nd order form as follows:

$$\boldsymbol{H_k(z) = \frac{b_{k0} + b_{k1}z^{-1}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}}}$$

Transfer function H(z) is generally composed of poles and coefficients (residues) of the partial-fraction expansion. A more direct result of the partial-fraction expansion of H(z) yields the functional equivalent as follows

$$\mathbf{H(z) = C + \sum_{k=1}^{N} \frac{A_k}{1 - p_k z^{-1}}}$$

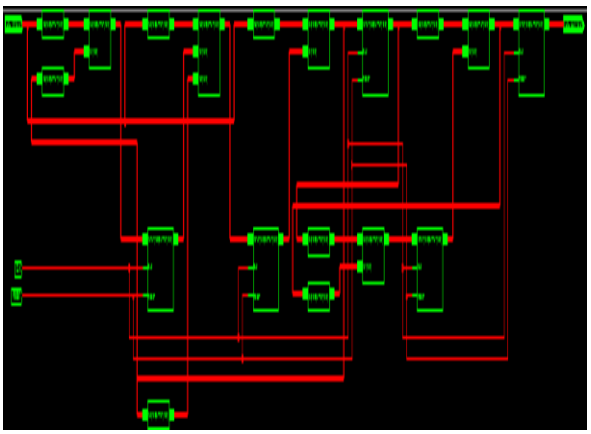


Fig 4 RTL view of Cascade structure IIR filter

ISSN: 2278 – 909X

Cascade-form realization of an IIR digital filter can be obtained by performing mathematical factoring on the transfer function H(z) into a cascade of 2nd order sub-blocks. The resulting transfer function can then be expressed as:

$$\mathbf{H(z)} = \prod_{k=1}^{K} H_k(z)$$

where K is the integer part of (N+1)/2 and Hk(z) has the 2nd order form as follows:

$$H_k(z) = \frac{b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}}$$

Since this is a cascade system, the matter of grouping together a pair of complex-conjugate poles and a pair of complex-conjugate zeros becomes extremely critical. The output of a quantized digital filter is strongly dependent on the sequential ordering of the K sub-blocks as well as the exact way in which numerator and denominator sections are paired together. Arbitrarily grouping these terms can be performed on the part of the system designer but at the cost of a potentially high output noise variance.

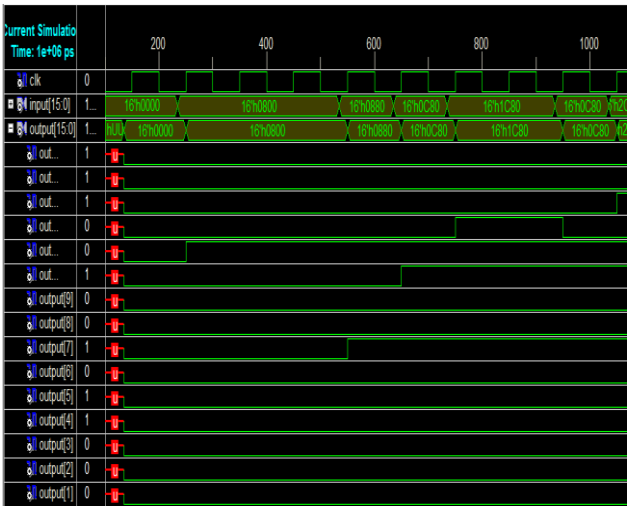


Fig 5 Timing simulation of Z^{-1} Delay module

By the time shifting property the time delay of signal by one sample is represented by Z^{-1} in Z domain. The fig 4 shows timing simulation of the delay module.

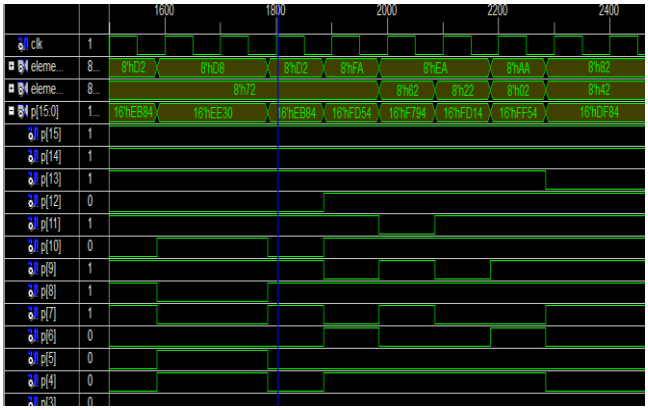


Fig 6 Timing simulation of coefficient multiplier unit.

The multiplier module is design for coefficients multiplication in the signal flow graph of direct form structures in IIR filter design. The RTL view of the multiplier module requires 818 gate counts.

The comparative analysis in tabulation format is shown below in table I, II and III.

Table II: Utilization of Cells, Registers, Blocks and Slices in Module design.

Module design	**Cell uses**	**Registers**	**Multiplications blocks**	**Slices**
Multiplier Module	940	0	1	7
Parallel Structure	34	0	7	1
Cascade structure	34	0	8	1
Controller logic	468	49	0	3
Program counter	158	16	0	3
SRAM	2106	1170	0	9

ISSN: 2278 – 909X

Table III: Latency computational time comparative analysis with related work

	Latency computational time		
Module design	**Our work**	**Reference Paper [1]**	**Reference paper [2]**
Multiplier Module	0.1us	3.0us	1.62us
Parallel Structure	0.1us	5.1us	1.62us
Cascade structure	0.1us	5.1us	1.62us
Controller logic	0.1us	3.0us	1.62us

Table IV: Comparative analysis of synthesis results with related work

Utilization	**Our work**	**Reference paper[3]**
Number of slice registers	1235	2191
Number used as logic cells	3740	700
Number of slice LUTs	700	997

Conclusion:

The goal of this thesis is to enhance the simulator based approach by integrating RISC hardware implementation to help the digital IIR filter with the improved latency, computation time 0.1us and throughput. Efficiency is measured in terms of the number of multiplications required for each output sample point, for this we use RISC implementation because it have reduce Instructions Computation of one output sample in one clock cycle, thus it is fastest architecture. Utilization of number of Slice registers is 1235 and number of slice LUTs is 700.

References:

[1] Sheikh Md. Rabiul Islam, Robin Sarker, Shumit Saha, A. F. M. Nokib Uddin " Design of a programmable digital IIR filter based on FPGA" IEEE International Conference on Infonnatics, Electronics & Vision ICIEV year 2012.

[2]Asma Hemdani,M.Wissem Naouar,Ilhem Slama Belkhodja,Eric Monmasson,"Design of a Digital IIR Filter for Active Filtering Applications"IEEE international conference in year 2012.

[3]Sufeng,Semih and Jafar Saniie"FPGA based Architectures for High Performance Adaptive FIR Filter Systems"IEEE International Conference 2013.

[4] ZHANG Jian-ping, PAN Ling-ling, DING Quan-fei "IIR Digital Filter Design Based on DSP and SOPC" International Symposium on Intelligence Information Processing and Trusted Computing year 2010.

[5] Amit Kumar Singh Tomar, Rita Jain "20-Bit RISC & DSP System Design in an FPGA" IEEE international conference in year 2013.

[6] Chaohua Dai, Weirong Chen, and Yunfang Zhu "Seeker Optimization Algorithm for Digital IIR Filter Design" IEEE Transactions On Industrial Electronics, Vol. 57, No. 5, May 2010 pp no. 1710.

[7] Shing-Tai Pan, Member, IEEE" Evolutionary Computation on Programmable Robust IIR Filter Pole-Placement design"ieee transactions on instrumentation and measurement, vol.60, no.4,april2011.

[8] Masayoshi Nakamoto,and Shuichi Ohno,"Design of Multi-Band Digital Filters and Full-Band Digital Differentiators Without Frequency Sampling and Iterative Optimization" ieee transactions on industrial electronics, vol. 61, no. 9, september 2014.

[9] Joaquín,J. Carlos, Daniel, Luis, Pedro, "Effects of Intermittent Faults on the Reliability of a Reduced Instruction Set Computing (RISC) Microprocessor" ieee transactions on reliability.

[10] S. Salivahanan, A. Vallavaraj And C. Gnanapriya, Digital Signal Processing,Tata McGraw - Hill, New Delhi 2009.

ISSN: 2278 – 909X