International Journal of Information Technology and Knowledge Management
July-December 2009, Volume 2, No. 2, pp. 329-331

PERFORMANCE IMPROVEMENT OF DIGITAL FIR FILTER USING HARDWARE / SOFTWARE CO-DESIGN METHODOLOGY

Meghana A. Hasamnis[*], Dr. Rajesh S. Pande[*] & Dr. S. S. Limaye[**]

Finite Impulse Response (FIR) filter is the key functional block in Digital Signal Processing. A number of implementations can be found in public literatures either by Hardware or Software solutions. Our aim is to design Digital FIR filter and apply the concept of Hardware/Software Co-Design. For implementing the concept of co-design, we have to partition the system into hardware and software and then check the result for co-design of Hardware and Software. Our main focus is to reduce the simulation time required that can be achieved using Hardware/Software Co-Design. Our Hardware part will be FPGA implementation of FIR filter using VHDL (VHSIC Hardware Description Language).We target our devices to Xilinx Virtex V. Our software part will be implementation of the same filter in C Language. In order to attain high performance Digital FIR Filter is implemented with Codesign Concept.

Keywords: Hardware/Software Co-Design, Digital FIR Filter, VHDL Simulation, C Coding.

I. Introduction

In recent years, considerable attention has been placed on the implementation of signal processing algorithms in VLSI, ranging from full custom VLSI to general purpose digital signal processors. A variety of approaches to high speed implementation of FIR filters have been pursued [1]. This Paper describes the performance improvement in Digital FIR Filter with Hardware/Software Codesign Concept.

FPGA's are being increasingly used for a variety of computationally intensive applications mainly in the realm of Digital Signal Processing (DSP) and communications. Due to rapid increases in technology, current generation of FPGA contain a very high number of configurable logic blocks(CLB), and are becoming more feasible for implementing wide range of applications such as communications and multimedia [7]. These functions are major determinants of performance and power consumption of the whole system. Therefore, it is important to have good tools for optimizing these functions.

First, we will design a 16-tap Digital FIR filter using VHDL and implement that filter on FPGA. Observe the simulation time. This will be simulation time for hardware Part.

Next, we will design 8-taps of Digital FIR filter using VHDL i.e. Hardware design part and remaining 8-taps using C Language. Add the simulation results for both (i.e. hardware and software).

[*] Department of Electronics Engineering, SRKNEC, Nagpur, Maharashtra, India. *E-mail: meghanahasamnis@rediffmail.com*[*] *E-mail: panderaj@yahoo.com*[*]

[**] Department of Electronics Engineering, JIT, Nagpur, Maharashtra, India. *E-mail: shyam_limaye@hotmail.com*

II. Filters

In signal processing, the function of a filter is to remove unwanted parts of the signal. There are two main kinds of filter, analog and digital. We prefer digital filter over analog because of the following :

1) A digital filter is programmable.
2) Digital filters are easily designed, tested and implemented.
3) The characteristics of analog filter circuits are subject to drift and are dependent on temperature. Digital filters do not suffer from these problems, and so are extremely stable with respect both to time and temperature.
4) Digital filters can handle low frequency signals.

There are a few terms used to describe the behavior and performance of Digital FIR filter including the following:

a) Filter Coefficients–The set of constants, also called tap weights, used to multiply against delayed sample values. For an FIR filter, the filter coefficients are, by definition, the impulse response of the filter.

b) Impulse Response–A filter's time domain output sequence when the input is an impulse. An impulse is a single unity-valued sample followed and preceded by zero-valued samples. For an FIR filter the impulse response of a FIR filter is the set of filter coefficients.

c) Tap–The number of FIR taps, typically *N*, tells us a couple things about the filter. Most importantly

it tells us the amount of memory needed, the number of calculations required, and the amount of "filtering" that it can do. Basically, the more taps in a filter results in better stopband attenuation (less of the part we want filtered out), less rippling (less variations in the passband), and steeper rolloff (a shorter transition between the passband and the stopband) [6].

d) Multiply-Accumulate (MAC)–In the context of FIR Filters, a "MAC" is the operation of multiplying a coefficient by the corresponding delayed data sample and accumulating the result. There is usually one MAC per tap.

III. ARCHITECTURE

Analysis with the FIR mathematic shows that minimal operations of "loading", "storing", "delay", "multiplication", "addition" and "iteration" are required for filter processing. The delay operation can be done with "loading" and "storing". Following Figure shows block diagram of FIR filter.

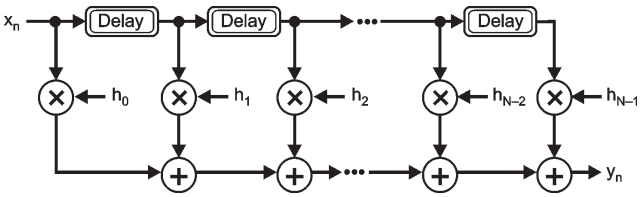

Figure 1: Block Diagram of FIR Filter

Equation (I) represents the output of L tap FIR filter which is convolution of latest L input samples. L is the number of coefficients $h(k)$ of the filter and $x(n)$ represents input time series.

$$y[n] = \Sigma h[k]x[n-k] \quad k = 0,1,....,L-1 \qquad \text{(I)}$$

The conventional tapped delay line realization of this is shown in Figure1. This implementation translates to L multiplications and L–1 additions to compute the result [3].

In signal processing, window function is a function that is zero valued outside some chosen interval. Applications of window functions include spectral analysis, filter design and beamforming. Window functions may be hanning, hamming, Bartlett, triangular etc. We are using Hamming window for calculating coefficient of FIR filter. Equation II defines the hamming window function.

$$w(n) = 0.54–0.46 \cos(2*3.14*n/(N–1)) \qquad \text{(II)}$$

n may have value from 0 to N–1,where N refers to number of taps. For example, for 16-tap filter, N = 16.

IV. HARDWARE-SOFTWARE CODESIGN

Hardware/Software Codesign can be defined as the cooperative design of hardware and software. Codesign research deals with the problem of designing heterogeneous systems. One of the goals of codesign is to shorten the time-to-market while reducing the design effort and costs of the designed products. Therefore, the designer has to exploit the advantages of the heterogeneity of the target architecture. The advantages of using processors are manifold, because software is more flexible and cheaper than hardware [4, 8]. This flexibility of software allows late design changes and simplified debugging opportunities. Furthermore, the possibility of reusing software by porting it to other processors, reduces the time-to-market and the design effort [9]. Finally, in most cases the use of processors is very cheap compared to the development costs of ASICs, because processors are often produced in high-volume, leading to a significant price reduction. However, hardware is always used by the designer, when processors are not able to meet the required performance. This trade-off between hardware and software illustrates the optimization aspect of the codesign problem. codesign is an interdisciplinary activity, bringing concepts and ideas from different disciplines together, e.g. system-level modelling, hardware design and software design [10].

Figure 2: Flow Chart of Co-Design Process

VI. RELATED WORK

All our work related to hardware will be done on Xilinx Virtex III devices and software part will be implementation of FIR filter in C Code.

Program of C to find out the filter coefficients for a cut-off frequency of 0.2" is as follows:

```
# include<stdio.h>
# include<math.h>
# define PI 3.14159265636
void main( )
{
 int n;
 double hrec[16],hham[16],arg;
 double norm1,norm2;
 for(n=0;n<=7;n++)  // LPF filter design
{
 arg = (.5+n)*0.2*PI;
 hrec[8+n] = .2*sin(arg)/arg;
 hrec[7-n] = hrec[8+n];
}
for(n+0;n<=15;n++)
{
 hham=hrec[n]*(0.54-0.46*cos(2*PI*n/15));
}
norm1=norm2=0.00;
for(n=0;n<=15;n++)
{
 norm1=norm1+hrec[n];
 norm2=norm2+hham[n];
}
for(n=0;n<=15;n++)
{
 hham[n]=hham[n]/norm2;
 printf("coeff%d =%f\t",n,hham[n]);
}
```

VII. RESULTS AND CONCLUSIONS

We will get the value of coefficients as :

coeff0=-0.003471	coeff1=-0.004851
coeff2=-0.004246	coefff3=0.008891
coeff4=0.044237	coeff5=0.100233
coeff6=0.160100	coeff7=0.0.199106
coeff8=0.0.199106	coeff9=0.160100
coeff10=0.100233	coeff11= 0.044237
coeff12=0.008891	coeff13=-0.004246
coeff14=-0.004851	coeff15=-0.003471

After designing the 16 tap Digital FIR Filter with the concept of Hardware/ Software Our conclusion is that the simulation time required for hardware implementation will be more as compared to implementation using hardware-software co-design concept used for Digital FIR filter Design. Thus, co-design concept helps in improving time.

VIII. REFERENCES

[1] Monson H. Hayes, Statiscal Digital Signal Processing and Modelling, *Wiley,* (1996) ISBN 0-471-59431-8.

[2] Simon Haykin Adaptive Filter Theory, *Prentice Hall,* (2002) ISBN 0-13-048434-2.

[3] I. Bolsens, H. De Man, B. Lin, K. Van Rompaey, S. Vercauteren, and D. Verkest, "Hardware-Software Codesign of Digital Telecommunication Systems", Proceedings of *IEEE,* **85** (1997) 391-418.

[4] Y. Li, T. Callahan, E. Darnell, R. Harr, U. Kurkure, and J. Stockwood, "Hardware–Software Codesign of Embedded Reconfigurable Architectures", in Proc. *Design Automation Conference,* 2000.

[5] R. Niemann, HW/SW Co-design for Data Flow Dominated Embedded Systems. Kluwer Academic Publishers, 1998.

[6] Ben Ismail, T. Jerraya, A. A. Synthesis Steps and Design Models for Codesign. *Transactions on Computers*, (1995) 44-52.

[7] Kalavade, A. Lee,. A Hardware-Software Codesign Methodology for DSP Applications. *IEEE Design & Test of Computers,* **10**(3) (1993) 16-2.

[8] D. Sharp, W. B. Gardner and M. Serra, Gizgate: An Object-Oriented Gateway for Hardware/Software Codesin on the CMC Rapid Prototyping Board. Proc. FDP '98, *Montreal,* (1998).

[9] W. Gardner and M. Serra. "An Object-Oriented Layered Approach to Interfaces for Hardware/Software Codesign of Embedded Systems". *Proc. Hawaii Int. Conf. on System Sciences*, (1998).

[10] J. Rozenblit and K. Buchenrieder, Ed. *Codesign.* IEEE Press, 1995.