Available online at www.elixirpublishers.com (Elixir International Journal)

Electrical Engineering

Elixir Elec. Engg. 97 (2016) 42402-42404

ISSN: 2229-712X

Design of Digital Low Power FIR Filter with Serial Architecture.

Kunal P. Raval

RKDF Institute of Science and Technology, Bhopal.

ARTICLE INFO

Article history:

Received: 11 September 2015;

Received in revised form:

21 August 2016;

Accepted: 31 August 2016;

Keywords

Assignment 3,

FIR Filter,

Delay balancing.

ABSTRACT

This paper is about designing a 15 tap 8-bit FIR filter using Direct form II. Serial Architecture is used for Multiplication and Accumulation. Entire Design is created in structural manner using verilog models from fsa0m_a a library. Delay balancing technique is used for reducing glitches in Multiplier.

Introduction

FIR filter of N th order has N-1 taps and has N filter coefficient. Here Design of 14th order FIR Filter is presented. The Design is rigid for given coefficients. Serial Architecture is used for implementation.

Given Filter specification is as follows,

15 tap FIR filter with following coefficients,

-0.04557, 0, 0.06366, 0, -0.1061, 0, 0.3183, 0.5, 0.3183, 0, -0.1061, 0, 0.06366, 0, -0.04557.

These coefficients will give below time domain equation,

$Y[n] = x[n] * (-0.04557) + x[n-2] * (0.06366) + x[n-4] * (-0.1061) + x[n-6] * (0.3183) + x[n-7] * (0.5) + x[n-8] * (0.3183) + x[n-10] * (-0.1061) + x[n-12] * (0.06366) + x[n-14] * (-0.04557)$

Z transform of this will be,

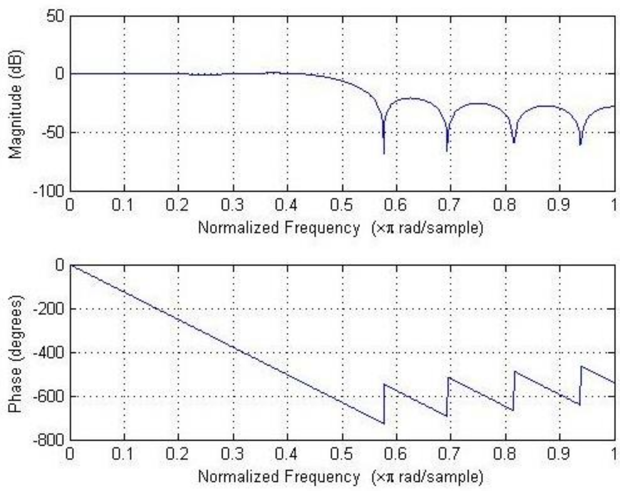


Figure 1. Frequency Response of Desired Filter

Arithmetic right shift. So total No. of multiplication operation in 250μs is 8.

So after multiplication operation we have Nine numbers to add. Here Carry save adder & Ripple carry in cascade pair is used instead of using only carry adder. As carry ripple adder consumes more power than carry save because of carry propagation.

$X(z) = \frac{Y(z)}{}0.04557 + 0.06366 * z^{-4} + -0.1061 * z^{-4} +$

So within 250μs 8 multiplication operations and 9 addition

$.3183 * z{-}6 + 0.5 * z{-}7 + 0.3183 * z{-}8 + -0.1061 * z{-}10 + 0.06366 * z{-}12 + -0.04557 * z{-}14$

This will give -6db Bandwidth at 1 kHz and greater than 20dB Stop-band attenuation at 1.1 kHz.The bode plot of given transfer function is shown in figure 1.

Design Strategy

Sampling Frequency and Architecture Frequency

As frequency specification is given from 0 to 2 kHz, Sampling frequency is chosen of 4kHz. So at every 250μs new sample arrives.

Design has Six zero coefficients and Nine nonzero coefficients. But out of those Nine one coefficient is 0.5. Multiplication of any number with 0.5 can be done by simply giving Operations should be complete. As we are implementing serial Architecture, the frequency of Architecture will be 10 times of sampling frequency i.e. 40 kHz.

Number representation in FIR

Here required resolution in input sample is 8bit. As given coefficients are signed so signed number system must required. So 8 bit signed number-system is opted here. Samples are quantized form -128 to 127.

As all coefficients are less than 0.5 in magnitude except that one 0.5, we can represent them in signed numbers by multiplying them by 256 and than rounding and then converting them to signed binary. Moreover here the coefficients are in replica style. Coefficient representation is as follows.

− 0.04557: −11.6659: −12: 80 b11110100

+ 0.06366: 16.297: 16: 80 b00010000

− 0.1061: −27.1616: −27: 80 b11100101

+ 0.3183: 81.4848: 81: 80 b01010001

Quantification Error

To analyze quantization error, two filters are modeled in Matlab. One is Ideal FIR with real coefficients and other is Quantized FIR with quantized coefficients with resolution of ±1/128 = 0.0078125.

Quantized FIR takes round off values of input signals and Ideal FIR takes actual real values of input signal.Input signal is discrete sine wave of magnitude 128 and frequency 400Hz (with 1600Hz noise) and is given to both FIR. The error due to quantization is shown in figure 2.

Tele:

E-mail address:kunal_raval2006@yahoo.co.in

Frequency response of coefficient error vector is shown in figure 3.

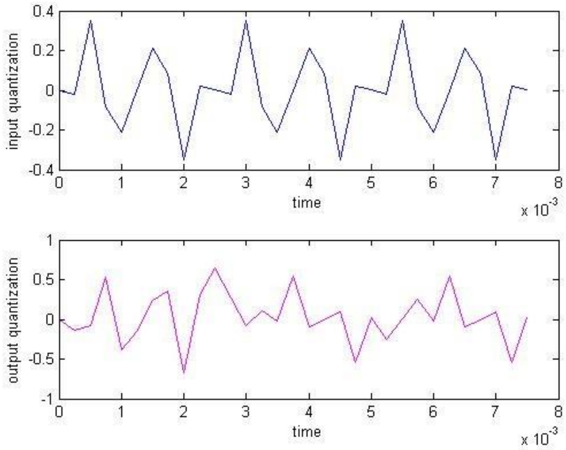


Figure 2. Quantization Error in Time Domain

Choice of Adders and Multiplier

As we multiply coefficients by 256 before they get multiplied by given sample, we need to divide the multiplier output after 256 before it goes into adder. Multiplier output is of 16-bit. So only higher 8 bits goes into adder. Adder need to add 9 such 8 bit numbers so adder should be of 12-bit.

So we need 8 bit signed multiplier and 12 bit adder. As number system is 2's complement signed addition can be treated as unsigned addition.

As we have 250µs, to multiply 8 numbers, even after dividing total time in 10 segments, each multiplication will be given 25µs to get complete, which is more than enough. So in this design speed is not a constraint. So Simple Signed Array multiplier is used. Addition and Multiplication cycles are overlapped. For addition Carry Save Adder and Carry Ripple adder are used.

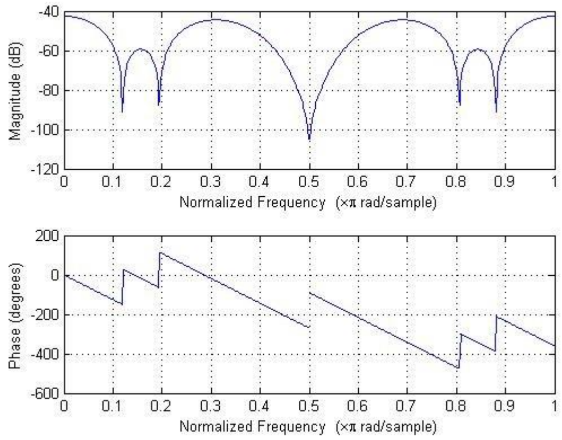


Figure 3. Quantization Error in Frequency Domain

Architecture Design

Architecture is made with structural methodology. Everything is made with cells from fsa0m_a stranded cell library. Top level representation is shown in figure 4.

FIR Pipeline is a chain of 8-bit register. Its diagram is shown in fig 5. It runs at frequency of 4 kHz. At every 250µs new sample arrives at x [0] and all values in pipeline shifts to next register.

FIR MUX block contains 2:1 muxes and 4:1 muxes. Select lines of MUX are driven by control unit. FIR MUX passes one by one each coefficient and its sample to"m1" and"m2" pin. In single FIR pipeline cycle i.e. 250µs S3, S1, S0 runs for 8 cycles.

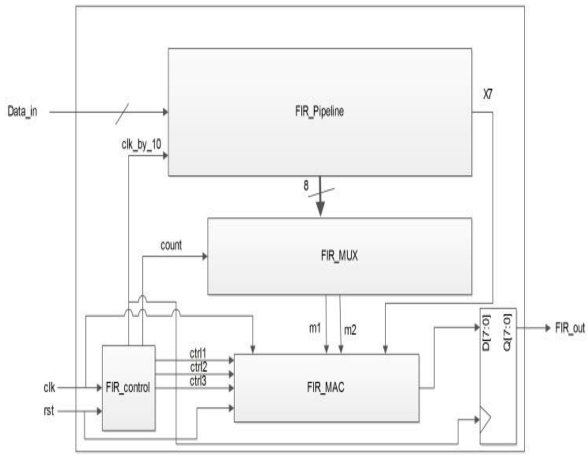


Figure 4. Fir Top Module

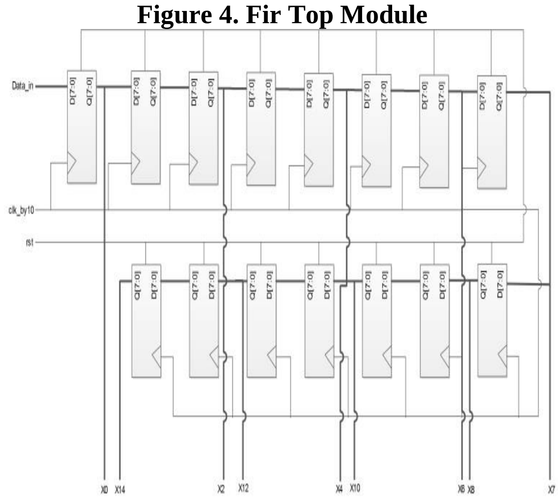


Figure 5. Fir Pipelining For Data Buffering

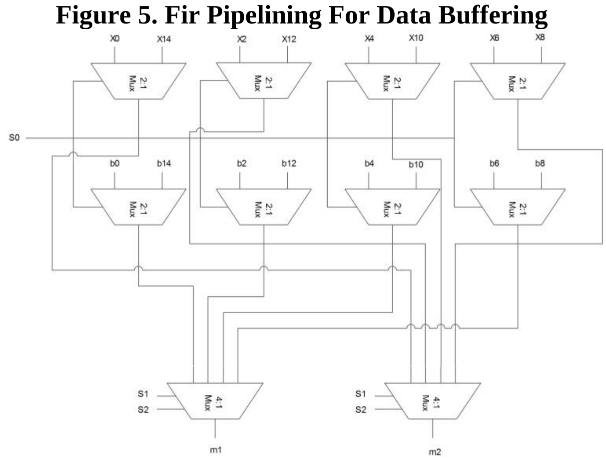


Figure 6. Fir Mux Block for Passing Selected Values from Pipeline to Mac Unit

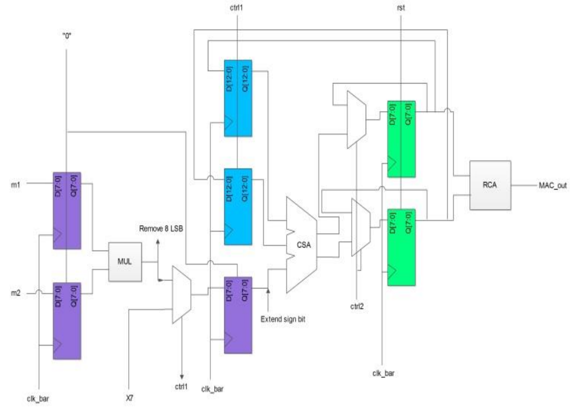


Figure 7. Fir Mac Unit to Compute Output Response.

Reducing Glitches in Multiplier

Array based 8-bit signed Multiplier is used in Design.8-bit Array multiplier consists of total 56 instances. Some of them are full-adder blocks and some of them are full- subtracters. There are some dependencies between them. Like, sum of instance i goes into Ain of instance j. So all spurious transitions at sum of instance i will propagate through next stages.

Our scheme of reducing glitches disables such unwanted spurious transitions to propagate. It propagates sum of current stage to next stage only after its output is stabilized to correct value. So each adder and subtracter block sees Ain, Bin, and C in arriving at the same time.

The schematic is shown in fig. 8. It uses tristate buffers and chain of delay cells. Intermediate output from delay cell is given to control of tristate buffers. The delay of delay cell is little higher than the delay of full adder or full subtracter

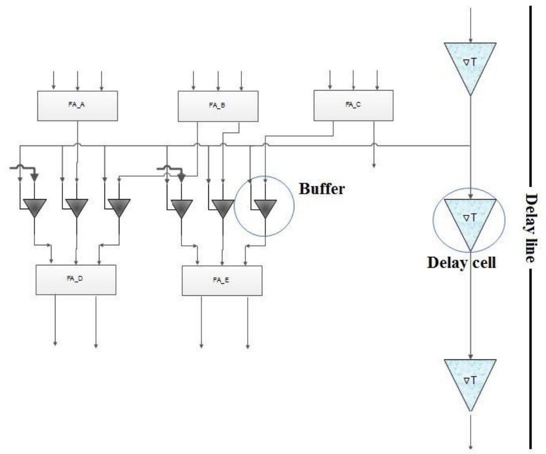


Figure 8. Mechanism to Avoid Spurious Transition to Propogate

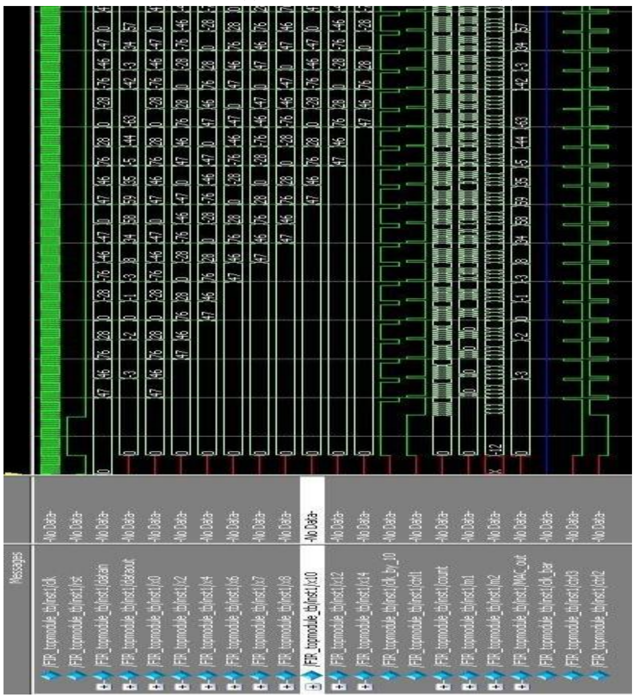
Simulation Results

A. FIR top module wave forms along with different input and outputs of other modules are also shown in figure 9.

Power Results achieved from Cadence RTL compiler is as follows:

Leakage Power 0.000 mW

Dynamic Power 0.343 mW Total Power 0.343 mW



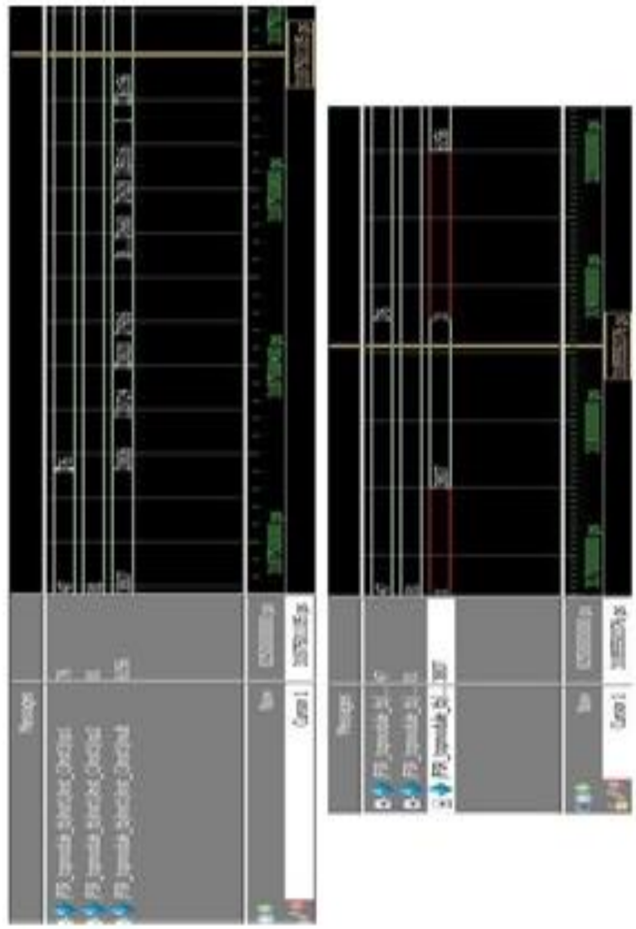


Figure 9. Fir Top Module Waveforms

Conclusions

Serial Architecture for given FIR filter specification is optimized at algorithm level because it avoids some of mul-tiplication operations that can be performed by shifting. So it has less hardware compare to Parallel Architecture, hence less leakage power.

Delay balancing technique uses tri-state buffers instead of simple buffers, so it reduces glitches, and hence reduces dynamic power.

This Design with the proposed delay balancing technique is advisable for both low as well as high signal activity due to less glitch in arithmetic circuits.

References

[1] H E weste, D. M. Harris, Principles of CMOS VLSI Design, Addiision- Wesley, MA, May 2004.

[2] A. V. Oppenheim, A. S. Willsky, Second Edition Signals and Systems, Pearson education, Inc, publishing as Prentice Hall.

[3] K. K. Parhi, “A systematic approach for design of digit-serial signal processing architectures,”IEEE ’A-ans. Circuits and Systems, Vol. 38, pp.358-375, April 1991.

[4] Delay Balanced Multipliers for Low PowerLow Voltage DSP Core, Toshiyuki Sakuta*, Wai Lee and Poras T. Balsara Low Power Electronics, 1995, IEEE Symposium on

[5] Xilinx, The Programmable Logic Data Book, Xilinx, Inc.,1996.