Energy-Efficient Low Latency IIR Using Enhanced Boost Logic

[1]R.Gomathi & [2]Kalpalatha Reddy

[1]SKR Engineering College, Poonamallee , Chennai
[2]ECE dept. SKR Engineering College, Poonamallee , Chennai
E-mail : go22mathi@gmail.com

***Abstract* – In this work, a 6-tap 8-bit IIR filter which uses a novel charge recovery logic, called Enhanced Boost Logic, which achieves high efficiency and high performance operation through the use of aggressive voltage scaling, gate overdrive, and charge-recovery techniques. A 6-tap 8-bit IIR filter implemented using Enhanced Boost Logic with only 1.5 cycles of latency overhead compared to a static CMOS implementation. The IIR using EBL logic dissipates only 24mW which achieves 36.07% improvement compared to previously reported FIR Filter (14 tap 8-bit FIR) .**

Index Terms—Digital signal processing (DSP), low-power VLSI

I. INTRODUCTION

Voltage scaling has diminished with each advancement in process technologies, making power dissipation one of the primary design considerations for modern digital systems. This design describes a novel charge-recovery logic family and that utilizes energy-recovery techniques to recycle the charge from the system, effectively reducing dynamic power dissipation[1].

Charge recovery circuitry has the potential to reduce dynamic power consumption in digital systems with significant switching activity. To keep energy consumption to a minimum, charge-recovery circuitry is typically designed so that it maintains low voltage drops across device channels, while recovering the charge supplied to it every clock cycle[1]. The overall energy-efficiency of charge-recovery circuitry therefore depends on the rate at which transitions occur, yielding an inverse relationship between energy consumption and clock period. Relying on this energy/latency tradeoff, charge-recovery circuitry can operate with energy consumption below, the fundamental limit of static CMOS.

In recent years, a charge-recovery family that uses multiple power supply level is called as Boost Logic was demonstrated in silicon at clock speeds exceeding 1GHz [5], [6]. Boost Logic improves upon the energy/latency tradeoff of previous charge-recovery circuit families, as it relies on gate overdrive to evaluate logic functions with significantly decreased delay and with minimal short-circuit current. It thus has the potential to achieve high-speed and low-power operation with pipeline latencies that are comparable to those of static CMOS designs. Enhanced Boost Logic (EBL) is an improved version of the basic Boost Logic that achieves shorter pipeline latencies while retaining its energy advantages over static CMOS. Similar to Boost Logic, EBL is capable of operation at high clock frequencies by developing a near-threshold voltage before the onset of the power clock.

Evaluation devices in EBL have twice the gate overdrive compared to first-generation Boost Logic[5],[6] however, enabling the design of complex logic gates and thus decreasing total gate counts. Consequently, EBL used in IIR further improves upon the energy/latency tradeoff of Boost Logic, yielding lower latency while maintaining good energy efficiency. It improves upon Boost Logic also with respect to implementation complexity, as it requires a smaller number of power supplies and less area[12],[13].

II. METHODOLOGY

A. Enhanced Boost logic

Enhanced Boost Logic (EBL) is presented in this chapter which is an improved version of the basic Boost Logic that achieves shorter pipeline latencies while retaining its energy advantages over static CMOS. Similar to Boost Logic, EBL is capable of operation at high clock frequencies by developing a near-threshold voltage before the onset of the power-clock. Evaluation

devices in EBL have twice the gate overdrive compared to first generation Boost Logic, however, enabling the design of complex logic gates and thus decreasing total gate counts. Consequently, EBL further improves upon the energy/latency tradeoff of Boost Logic, yielding lower latency while maintaining good energy efficiency. EBL improves upon Boost Logic also with respect to implementation complexity, as it requires a smaller number of power supplies.

The Enhanced Boost Logic presented in this paper is another variant of Boost Logic that is aimed at pushing the iso-energy frequency point higher than Subthreshold Boost Logic, while at the same time decreasing latency overhead. Fig. 1 shows a cascade of three EBL buffers. Each EBL gate has two stages: Evaluation and Boost. Similar to SBL, the Boost stage consists of a cross-coupled inverter with the source of the PMOS connected to a charge-recovering clock phase pc, enabling high performance through enhanced gate overdrive. Unlike SBL, however, the Evaluation stage relies on a NMOS precharge device for pull-up, instead of a complementary pull-up network, thus increasing performance by avoiding the series-connected devices in the pull-down network (PDN). The bulk of all NMOS transistors are connected to ground, and the bulk of PMOS transistors in the cross-coupled inverters are connected to the corresponding power-clock phases. From a functional point of view, each EBL gate is equivalent to a combinational logic block (Evaluation stage) that is powered by a near-threshold supply V CC and drives a transparent latch synchronized by clock phase pc (Boost stage). Cascades of EBL gates are clocked by alternating clock phases pc and pc b.

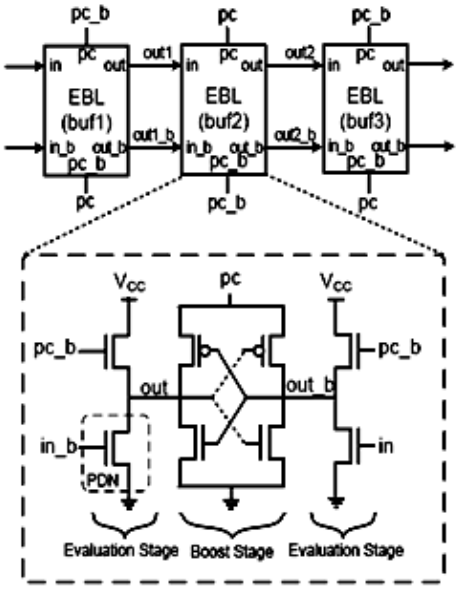


Figure 1.Cascade of three EBL buffers

III. ENHANCED BOOST LOGIC USED IN IIR FILTER

IIR filters are digital filters with infinite impulse response. Unlike FIR filters, they have the feedback (a recursive part of a filter) and are known as recursive digital filters.

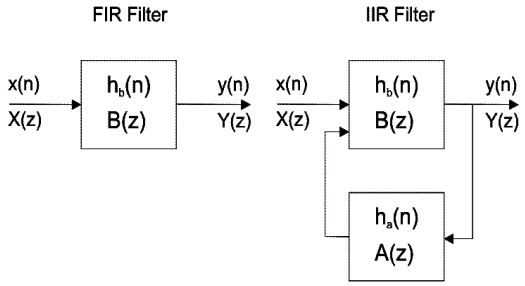


Figure 2 Block diagrams of FIR and IIR filters

For this reason IIR filters have much better frequency response than FIR filters of the same order. Unlike FIR filters, their phase characteristic is not linear which can cause a problem to the systems which need phase linearity. For this reason, it is not preferable to use IIR filters in digital signal processing when the phase is of the essence.

Otherwise, when the linear phase characteristic is not important, the use of IIR filters is an excellent solution. There is one problem known as a potential instability that is typical of IIR filters only. FIR filters do not have such a problem as they do not have the feedback. For this reason, it is always necessary to check after the design process whether the resulting IIR filter is stable or not.

IIR filters can be designed using different methods. One of the most commonly used is via the reference analog prototype filter. This method is the best for designing all standard types of filters such as low-pass, high-pass, band-pass and band-stop filters.

The design method using reference analog prototype filter. Figure.3 illustrates the block diagram of this method.

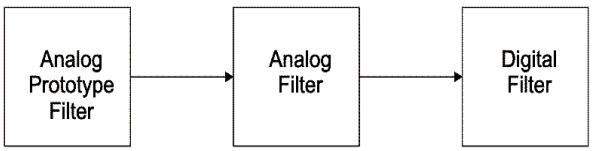


Figure 3. Block diagram of design method using reference analog prototype filter

FIR filters can have linear phase characteristic, which is not typical of IIR filters. When it is necessary to have linear phase characteristic, FIR filters are the only available solution. In other cases when linear phase characteristic is not necessary, such as speech signal processing, FIR filters are not good solution. IIR filters should be used instead. The resulting filter order is considerably lower for the same frequency response.

The filter order determines the number of filter delay lines, i.e. number of input and output samples that should be saved in order that the next output sample can be computed. For instance, if the filter order is 10, it means that it is necessary to save 10 input samples plus 10 output samples preceding the current sample. All these 21 samples will affect the next output sample. The IIR filter transfer function is a ratio of two polynomials of complex variable z^{-1}. The numerator defines location of zeros, whereas the denominator defines location of poles of the resulting IIR filter transfer function.

A. Infinite impulse response (IIR) filter design

The impulse response or the frequency response classify digital filters. The impulse response is the response of a filter to an input impulse: x[0]=1 and x[i]=0 for all i≠0. The Fourier transformation of the impulse response is the filter frequency response which describes the gain of the filter for different frequencies.

If the impulse response of the filter falls to zero after a finite period of time, it is an FIR (Finite Impulse Response) filter. However, if the impulse response exists indefinitely, it is an IIR (Infinite Impulse Response) filter. How the output values are calculated determines whether the impulse response of a digital filter falls to zero after a finite period of time. For FIR filters the output values depend on the current and the previous input values, whereas for IIR filters the output values also depend on the previous output values. The output values of IIR filters are calculated by adding the weighted sum of previous and current input values to the weighted sum of previous output values. If the input values are x_i and the output values y_i, the difference equation defines the IIR filter:

$$y_i = \frac{1}{a_0}\left[-\sum_{j=1}^{N_y-1} a_j\, y_{i-j} + \sum_{k=0}^{N_x-1} b_k\, x_{i-k}\right]$$

The number of forward coefficients N_x and the number of reverse coefficients N_y is usually equal and is the filter order. The higher the filter order, the more the filter resembles an ideal filter. This is illustrated in the following figure 4 of a frequency response of lowpass Butterworth filters with different orders. The steeper the filter gain falls, the higher the filter order is.

The most commonly used IIR filter design method uses reference analog prototype filter. It is the best method to use when designing standard filters such as low-pass, high-pass, bandpass and band-stop filters. The filter design process starts with specifications and requirements of the desirable IIR filter. A type of reference analog prototype filter to be used is specified according to the specifications and after that everything is ready for analog prototype filter design. The next step in the design process is scaling of the frequency range of analog prototype filter into desirable frequency range. This is how an analog prototype filter is converted into an analog filter.

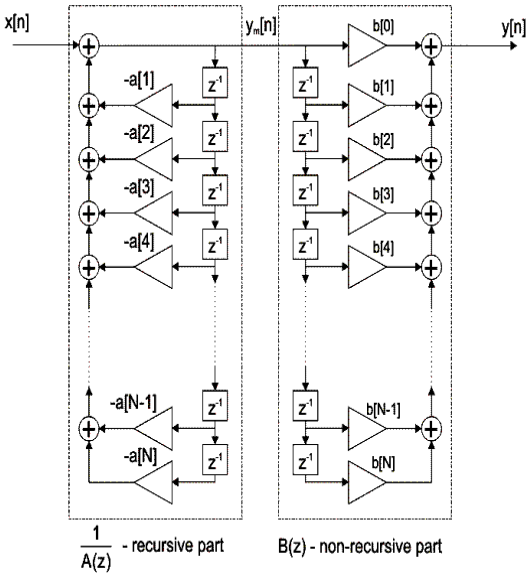


Figure.4 IIR filter direct realization

This structure is also known as a direct form I structure. As seen from Figures 4, direct realization requires in total of 2N delay lines, (2N+1) multiplications and 2N additions.

Direct realization is very convenient for software implementation and this is where it is most commonly used.

Biquad filter is a second order recursive linear filter, containing two poles and two zeros. Biquad" is an abbreviation of "biquadratic", which refers to the fact that in the Z domain, its transfer function is the ratio of two quadratic functions:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

High-order recursive filters can be highly sensitive to quantization of their coefficients, and can easily become unstable. This is much less of a problem with first and second-order filters; therefore, higher-order filters are typically implemented as serially-cascaded biquad sections (and a first-order filter if necessary)[10]. The two poles of the biquad filter must be inside the unit circle for it to be stable. In general, this is true for all filters i.e. all poles must be inside the unit circle for the filter to be stable. The figure5 shows the block diagram of an IIR using EBL logic.

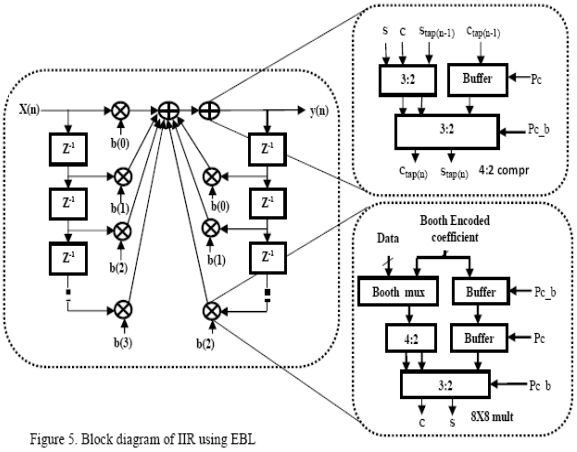


Figure 5. Block diagram of IIR using EBL

B. 4:2 Compressors

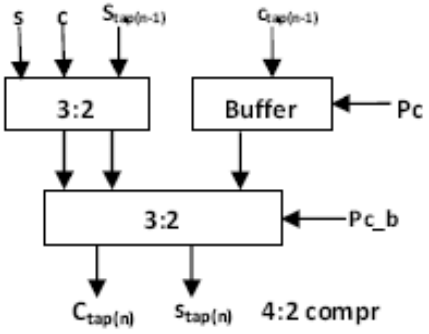


Figure 6. 4:2 Compressor with buffer

A full adder is a 3:2 lossy compressor, it sums three one-bit inputs, and returns the result as a single two-bit number; that is, it maps 8 input values to 4 output values. Thus, for example, a binary input of 101 results in an output of 1+0+1=10 (decimal number '2'). The carry-out represents bit one of the result, while the sum represents bit zero. Likewise, a half adder can be used as a 2:2 lossy compressor, compressing four possible inputs into three possible outputs.

Such compressors can be used to speed up the summation of three or more addends. If the addends are exactly three, the layout is known as the carry-save adder. If the addends are four or more, more than one layer of compressors is necessary and there is various possible designs for the circuit: the most common are Dadda andWallace trees. This kind of circuit is most notably used in multipliers, which is why these circuits are also known as Dadda and Wallace multipliers.

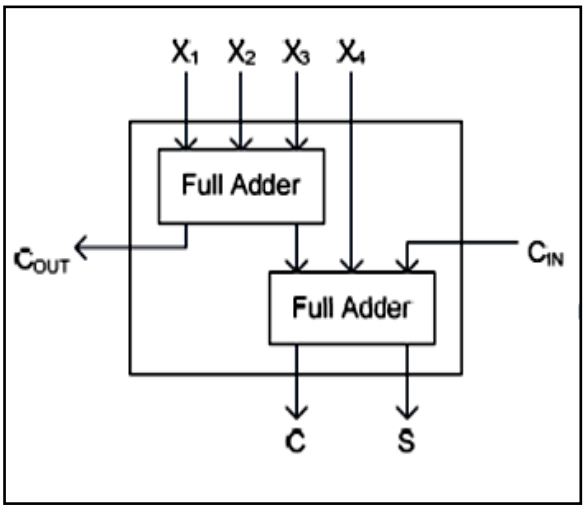


Figure 7. 4:2 Compressor

The 4:2 compressor takes five equally weighted inputs (CIN, X1, X2, X3, X4) and generate a sum bit (S), a carry-bit (C) and a carry-propagate-bit (COUT). The 4:2 compressor array is formed by a series of 4:2 compressors cascaded together it together, is used to perform column wise compression of the partial product.

IV. RESULTS AND DISCUSSION

Results are simulated using modelsim6.2c and synthesized using Xilinx ISE 9.9 . The simulation result is shown in figure 8 in which the random inputs are generated from the test bench wave form and the input is operated in IIR using EBL which tends to reduce the power consumption using recovery logic. Figure 9 shows the power calculation which shows 24mw of power consumption, whereas EBL based FIR dissipates 39.1mW and achieves 36% improvement.

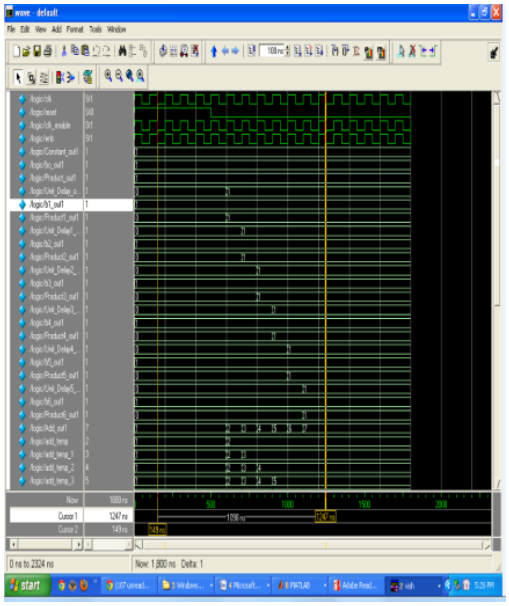
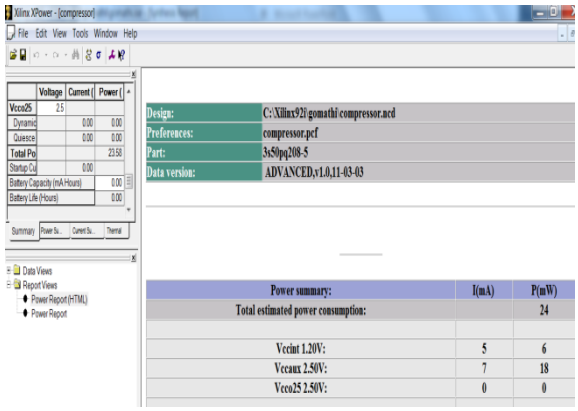


Figure 8. Simulation result of the 6tap 8bit IIR



Design:	C:\Xilinx92i\gomathi\compressor.ncd
Preferences:	compressor.pcf
Part:	3s50pq208-5
Data version:	ADVANCED,v1.0,11-03-03

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		24
Vccint 1.20V:	5	6
Vccaux 2.50V:	7	18
Vcco25 2.50V:	0	0

Figure 9. Power calculation

V. CONCLUSION

Sixth order IIR filter with Enhanced boost logic (EBL) has been designed which consumes less power. EBL uses an aggressively scaled near threshold supply to perform logic evaluation. The EBL in IIR presented in this project is aimed at pushing the iso-energy frequency point higher than EBL in FIR while at same time decreasing latency over head and consumes less area. The design of Sixth order IIR filter with EBL was verified using Modelsim 6.2c and synthesized using Xilinx ISE 9.9 and the hardware results are obtained using FPGA Xilinx Spartan 3 kit. It dissipates only 25mW of power which achieves 36.07% improvement compared to previously reported FIR Filter (14 tap 8-bit FIR)

As a future enhancement power dissipation can be further decreased by using different charge recovery logic or by pipelining concept in IIR.

VI. REFERENCES

[1] Jerry C. Kao, Wei-Hsiang Ma, Visvesh S. Sathe,and Marios Papaefthymiou, "Low-Latency Energy-Recovery" *IEEE transactions on very large scale integration (vlsi) systems*, vol. 20, no. 6, june

[2] S. Kim, C. Ziesler, and M. Papaefthymiou, "Charge-recovery computing on silicon," *IEEE Trans. Comput.*, vol. 54, no. 6, pp. 651–659,Jun. 2005.

[3] S. Kim and M. Papaefthymiou, "Single-phase source-coupled adiabatic logic," in *Proc. Int. Symp. Low Power Electron. Des.*, 1999, pp. 97–99.

[4] S. Kim, C. Ziesler, and M. Papaefthymiou, "A true single-phase 8-bit adiabatic multiplier," in *Proc. Des. Autom. Conf.*, 2001, pp. 758–763.

[5] V. Sathe, J.-Y. Chueh, and M. Papaefthymiou, "A 1.1 GHz chargerecovery logic," in *Dig. Techn. Papers IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2006, pp. 1540–1549.

[6] V. S. Sathe, J.-Y. Chueh, and M. C. Papaefthymiou, "Energy-efficient GHz-class charge-recovery logic," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 38–47, Jan. 2007.

[7] Y. Moon and D.-K. Jeong, "An efficient charge recovery logic circuit," *IEEE J. Solid-State Circuits*, vol. 31, no. 4, pp. 514–522, Apr. 1996.

[8] L. B. Jackson, "An improved Martinez/Parks algorithm for IIR design with unequal numbers of poles and zeros," *IEEE Trans. Signal Processing*, vol. 42, pp. 1234–1238, May 1994.

[9] A. Betser and E. Zeheb,"Reduced Order IIR Approximation of FIR Digital Filters," *IEEE Transactions on Signal Processing*, Vol. 39, No. 11, pp.2240-2544, Nov. 1991.

[10] Yutaka Yamamoto, Brian D. O. Anderson, Masaaki Nagahara, and Yoko Koyanagi, "Optimizing FIR Approximation for Discrete-Time IIR Filters",*IEEE Transactions on Signal Processing*, VOL. 10, NO. 9,pp 273-276 , Sep., 2003

[11] V. Sreeram and P. Agathoklis, Design of linear-phase IIR filters via impulse-response gramians,

IEEE Transactions on Signal Processing, vol. 40, pp. 389-394, Feb. 1992.

[12] B. Beliczynski, J. Kale, and G. D. Cain, Approximation of FIR by IIR digital filters: An algorithm based on balanced model reduction,*IEEE Transactions on Signal Processing*, vol. 40, pp. 532-542, Mar. 1992.

[13] M. F. Fahmy, Y. M. Yassin, G. Abdel-Raheem, and N. El-Gayed, Design of linear-phase IIR filters from FIR specifications, *IEEE Transactions on Signal Processing*, vol. 42, pp. 437-440, Feb. 1994.

◈◈◈