A Delay-Insensitive FIR Filter for DSP Applications

W. Kuang[1], J.S. Yuan[1], R. Demara[1], D. Ferguson[2], and M. Hagedorn[2]
[1]Chip Design and Reliability Laboratory, University of Central Florida, Orlando, FL 32816
tel: (407)823-5719, fax: (407)823-5835, email: yuanj@mail.ucf.edu
[2]Theseus Logic, Inc., 3501 Quadrangle Blvd., Suite 100, Orlando, FL 32817

Abstract: **This paper explores delay-insensitive design techniques for digital signal processing applications. Direct and interleaved FIR filter architectures using this technique are designed. Synopsys simulation shows the interleaved FIR filter is better than direct architecture in throughput and chip area.**

1 INTRODUCTION

In synchronous logic, clock skew becomes a bottleneck for system design when the chip density is larger and larger due to continued scaling of CMOS technology. Although some methods to overcome this problem were proposed [1], avoiding clock skew becomes very costly in chip area. In addition, synchronous systems tend to consume more energy for unnecessary switching events. In contrast to synchronous systems, asynchronous circuits require no global synchronization. Subsystems are handshaked locally by the communication protocols between them. The absence of a global clock circumvents the clock skew problem. Moreover, switchings in a particular subsystem only occur when data are passing through it. Therefore, there is no dynamic power consumption when the subsystem is idle. This makes asynchronous design very attractive for system-on-a-chip application.

There are two main categories for asynchronous systems: bundled-data protocol and dual-rail protocol. Bundled-data protocols, however, are not delay-insensitive [2]. This leads to extensive timing analysis of worse-case simulation to ensure correct circuit operation. Thus timing design margins are required to compensate for wire delays in bundled-data protocol. Dual-rail protocols are traditionally constructed with C-elements and Boolean gates [3]. Dual rail can be used in delay insensitive circuits because the dual rail signal can be used to distinguish between data0 and data1, and no-data. Here we use the NULL Convention Logic (NCL) threshold gates [4] and NCL design paradigms to insure delay-insensitivity. The paradigms include monotonic transitions, mutual exclusive assertion groups (MEAGs), completeness of Data, and completeness of Null. The delay-insensitive NCL design allows the system builder to plug together NCL gates, circuits, and systems without having to worry about breaking the functionality because of timing issues. NCL, patented by Theseus Logic, Inc. is implemented in the design of digital filters using dual-rail protocol [5]. NCL is a clockless logic, where the control information is integrated in the data paths. NCL also provides low power, high performance, and low EMI advantages for wireless communication systems using digital signal processing functions. In fact, the unique structure of NCL is especially suitable for DSP applications, which are data-flow dominated. Computation and control can be easily integrated in one path.

In this paper, finite impulse response (FIR) filters for DSP applications [6] are designed and implemented in NCL. Direct architecture using multipliers and adders and interleave architecture using only logic gates and adders are evaluated.

2 NULL CONVENTION LOGIC

To perform a computation operation, one must differentiate between logic "1" and logic "0". In NCL they are represented by the dual-rail encoding shown in table 1. Null is also defined when both rails are low. In Table 1 X is represented by a two-wire mutually exclusive assertion group. An assertion of high on one-rail (X1) or zero-rail (X0) represents a logical Data_1 or Data_0 respectively. An assertion of high on both rails is invalid. A data wave front is defined as the presence of logical Data_1 or Data_0, while a Null wave front is similarly defined as the absence of a valid data value.

In NCL the circuit processes successive sets of data as its inputs. It is necessary to know when the processing of one set is completed, so that the following one can be inputted and processed. To do this, NCL inputs Null state to reset the circuit to all-Null state after one set has been processed. By detecting the completeness of output, NCL registers determine whether and when the circuit can accept the Data or Null input from the previous one.

Table 1 Dual-rail encoding

Logical value	Encoding	
	X0	X1
Data 1	L	H
Data 0	H	L
Null	L	L
Invalid	H	H

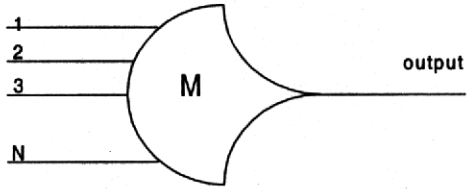


Fig 1. The symbol of M-of-N threshold gate

The basic building elements of NCL circuits are M-of-N threshold gates with hysteresis [5] as shown in Fig. 1. The threshold gate has N input signals and one output signal. Beginning with all inputs Null and asserting a Null output value, the gate will not assert a Data output until the number of the inputs asserted Data is equal to or more than M; After asserting a Data output, the gate will not assert a Null output until all of its inputs are Null. A C-element is a 2-of -2 threshold gate.

An asynchronous register consists of NCL threshold gates for data-path and another single gate to detect the output completeness and send readiness to accept the following input set (Data or Null). A 2-bit register is shown in Fig. 2. The bit a0 is encoded to a0_0 and a0_1 and the bit a1 is encoded to a1_0 and a1_1. The output bits z0 and z1 are encoded similarly. ki is the request signal from the next register and ko is the request signal sent to the previous register. It will manage the inputs and outputs among combinational circuits so as to implement the Data-Null alternative operation mode, shown in Fig. 3. Assume that all the data-paths are in a Null state and that ko_1 and ko_2 are requesting a Data wave front and that a complete Data set is presented to Register 1. When the complete data set passes through Register 1 and is recognized by the detecting gate in Register 1, the detecting gate transitions its control line to the previous register

to Null to indicate that it has received and stored the data wave front and the previous register can pass a Null wave front. The requested Null wave front from the previous register can arrive at Register 1. But, as long as ko_2 is Data, the Null wave front will be blocked and Register 1 will maintain the presentation of the set of Data values to the combinational circuit. ko_2 will remain Data until the Data wave front has propagated through the combinational circuit and has been received and detected by Register 2. The detecting gate in Register 2 detects the complete Data set and transitions its acknowledge line to Null and allow a Null wave front to pass through Register 1. This structure is inherently pipelined.

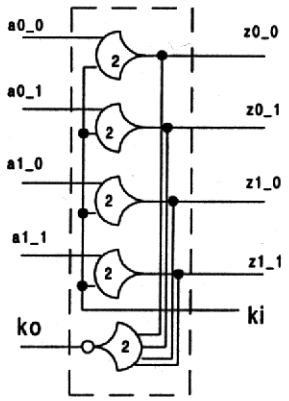


Fig. 2 2-bit register

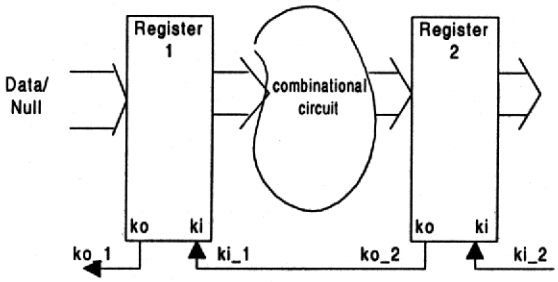


Fig. 3. NCL pipeline architecture

3 CIRCUIT DESIGN FOR FIR FILTER

A N-tap FIR filter performs the following convolution [6]

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

where h(k), k=0,1,2... N-1 are the coefficients of the filter and x(n) and y(n) are input and output sequences, respectively. The general structure of an FIR filter is shown in Fig. 4. The circuit consists of multipliers, adders and delay elements. Although realization of FIR filters can vary widely from one using dedicated hardware to another using code executed by a general purpose processor (e.g., TI TMS320C50) and its ancillary units, we focus on the implementation of specific design since NCL is still relatively new in the design community. For simplicity a 2-tap FIR filter is addresses in this paper.

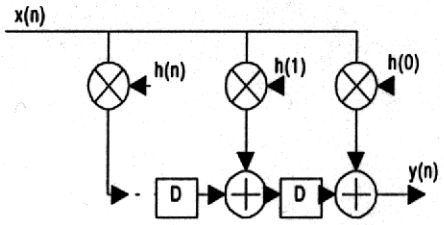


Fig. 4 General FIR structure

3.1 Basic Components

An adder is a basic unit in arithmetic circuits. In fact, multiplier is often constructed by adders and gates. Therefore, the performance of adders plays an important role in a computational circuit. An optimized NCL full adder is shown in Fig. 5 [2], where (Ci_0, Ci_1) denotes the dual-rail encoded carry input, (x_0, x_1) and (y_0, y_1) denote the dual-rail encoded input addends. (C0_0, C0_1) and (S_0, S_1) denote the dual-rail encoded carry and sum outputs respectively. The circuit consists of two 2-of-3 threshold gates and two 3-of-4 threshold gates.

A delay element is one of the fundamental building blocks in DSP circuits. Suppose that we are given an input sequence {x(n), n = 0,1,2,3...}, and that computation is based on x(I) and x(I-1), then a delay element is needed. The input of the delay element is X(I), and the output is x(I-1). Three registers are used to design a delay element, shown in Fig. 6. At the initial state, for register 1 and register 3, the outputs are set to Null while output request signal is set to Data; for register 2, the output is set to Data_0 or Data_1 while output request signal is set to Null.

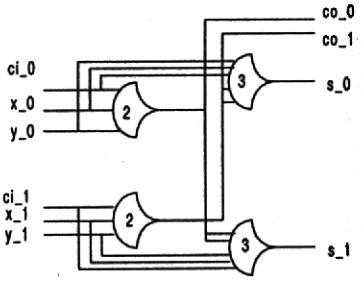


Fig.5. An optimized NCL full adder

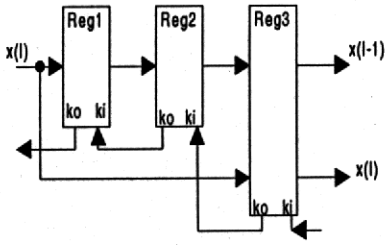


Fig. 6. Delay element

3.2 Direct FIR Architecture

Assume that multipliers, adders, and delay elements are given, one can directly construct an FIR filter shown in Fig 4. The multipliers are implemented as a 3-stage pipelined Baugh-Wooley carry-save array in NCL [7]. Synopsys simulation shows that the multiplier's data cycle is 9.488 ns, and that the FIR data cycle is 10.850 ns.

3.3 Interleaved FIR structure

Consider a 2-tap FIR filter operation

$$y(n) = f \cdot x(n) + g \cdot x(n-1)$$

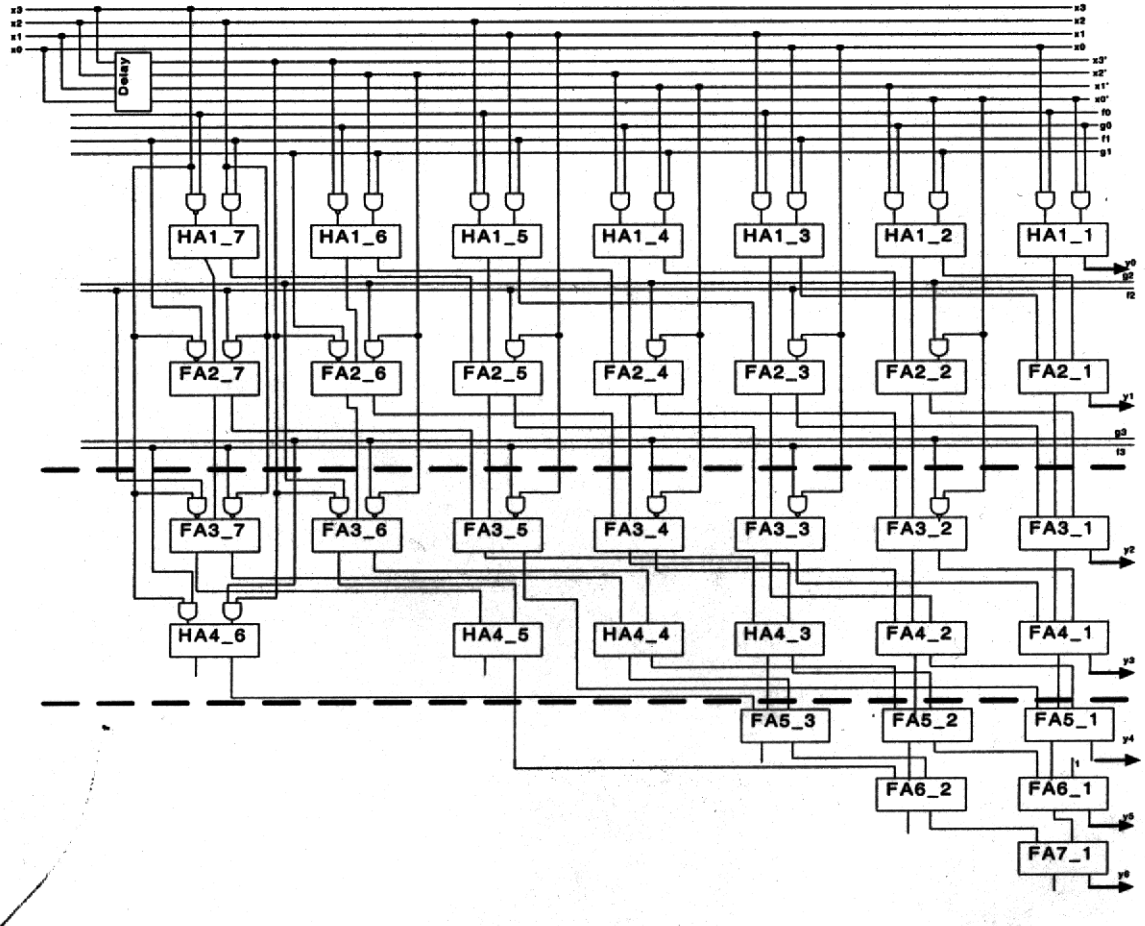
where f and g are the coefficients, x(n) is the input, and y(n) is the output. We can perform the computation and accumulation of partial products associated with f and g simultaneously, which leads to dramatic reduction in the routing complexity. Furthermore, since the truncation is performed only at the final step, it also leads to better accuracy. The multiplication chart for computing the output y(n) using this interleaved approach is shown in Fig. 7., which is the result of interleaving of two parallel Baugh-Wooley multiplications. The interleaved FIR architecture is shown in Fig. 8. An effective way to improve data throughput is to place registers on a feed-forward cutset of the data path to split the combinational circuit into several stages. For example,

if two registers are placed on the positions by dash lines, it becomes a 3-stage pipelining architecture. Similarly, an 8-stage pipelining architecture can be implemented in the cost of more hardware.

			/(f0 x3)	f0 x2	f0 x1	f0 x0
			/(g0 x3')	g0 x2'	g0 x1'	g0 x0'
		/(f1 x3)	f1 x2	f1 x1	f1 x0	
		/(g1 x3')	g1 x2'	g1 x1'	g1 x0'	
	/(f2 x3)	f2 x2	f2 x1	f2 x0		
	/(g2 x3')	g2 x2'	g2 x1'	g2 x0'		
f3 x3	/(f3 x2)	/(f3 x1)	/(f3 x0)			
g3 x3'	/(g3 x2')	/(g3 x1')	/(g3 x0')			
+)	1					
y6 .	y5	y4	y3	y2	y1	y0

Fig. 7. Multiplication chart for the interleaved FIR. f = f3.f2f1f0, g = g3.g2g1g0, xi represents the ith bit of x(n), xi' represents the ith bit of x(n-1), and /( ) represents one's complement operation.

Fig. 8 An interleaved FIR architecture



3.4 Simulation Results

We use the Synopsys simulation to validate the functions of several FIR architectures, and use the Synopsys Design Compiler to synthesize the architecture in order to obtain the speed and area performances. The data cycle is measured from the waveform shown in Fig. 10. (All data lines are not displayed for the sake of brevity). Table 2 shows a comparison of several FIR architectures addressed in this paper. The throughput is the reciprocal of data cycle. It is clear from Table 2 that the 3-stage pipelining interleaved architecture is the most optimized tradeoff in throughput and number of transistors used. Although the 8-stage pipelining interleaved architecture is little bit faster that the 3-stage pipelining interleaved one, it requires much more hardware. Note that the 3-stage pipelining interleaved architecture has the smallest latency.

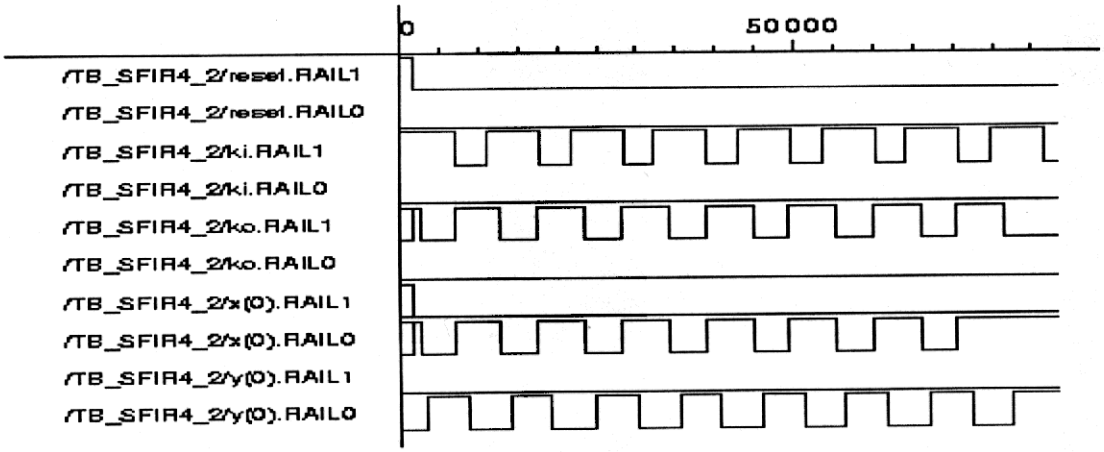


Fig. 9 Synopsys simulation waveform for measurement of data cycle

Table 2 Comparison of architectures for 2-tap 4-bit input 7-bit output FIR

	Data cycle (ns)	Throughput (MHz)	Latency (ns)	# of transistors
Direct	10.850	92.2	11.318	9680
Interleaved (non-pipelined)	10.550	94.8	11.170	5490
3-stage pipelining interleaved	**7.790**	**128.4**	**8.690**	**6834**
8-stage pipelini interleaved	7.720	129.5	10.100	9938

4 CONCLUSION

Direct and interleaved FIR filters using NCL techniques have been designed. Pipelining interleaved FIR architectures offer reduced data cycle time and high throughput compared to direct architecture. The three-state pipelining interleave FIR filter gives the most optimized result in throughput, latency, and number of transistors used. Our future work includes developing NCL design automation for DSP applications such as discrete cosine transform (DCT) and discrete Fourier transform (DFT) suitable for intellectual property (IP) design reuse.

References

[1] E.G. Friedman and W. Powell, "Design and analysis of a hierarchical clock distribution system for synchronous standard cell/macrocell VLSI", IEEE J. Solid-State Circuit, vol. SC-21, No. 2, pp. 240-246, April 1986.

[2] S. Hauck, "Asynchronous Design Methodologies: An Overview", Proceedings of the IEEE, vol. 83, No. 1, pp. 69-93, 1995.

[3] Ilana David, Ran Ginosar, and Michael Yoeli, " An Efficient Implementation of Boolean Functions as Self-timed Circuits", IEEE Transactions on Computers, vol. 41, No. 1, pp. 2-10, 1992.

[4] Karl M. Fant and Scott A. Brandt, "Null Convention Logic Systems", US patent 5,305,463, April 19,1994.

[5] Gerald E. Sobelman, and Karl M. Fant, "CMOS Circuit Design of Threshold Gates with Hysteresis", IEEE International Symposium on Circuits and Systems (II), pp. 61-65, 1998.

[6] J.G. Proakis and D.G. Manolakis, " Digital Signal Processing: principles, algorithm and applications", 3rd edition, Prentice Hall.

[7] Scott C. Smith, "Design Methodology and Optimization Techniques for Null Convention Delay-Insensitive Digital Circuits", Ph.D. Candidacy Proposal, UCF, 2000