A low-power asynchronous VLSI FIR filter.

Viv Bartlett
Cavendish School of Computer Science

Eckhard Grass
IHP-GmbH, Frankfurt (Oder), Germany

A Low-Power Asynchronous VLSI FIR Filter

V.A. Bartlett
Department of Electronic Systems
University of Westminster,
London, England
v.bartlett@westminster.ac.uk

E. Grass
IHP-GmbH
Frankfurt (Oder), Germany
grass@IHP-microelectronics.com

Abstract

An asynchronous FIR filter, based on a Single Bit-Plane architecture with a data-dependent, dynamic-logic implementation, is presented. Its energy consumption and sample computation delay are shown to correlate approximately linearly with the total number of ones in its coefficient-set. The proposed architecture has the property that coefficients in a Sign-Magnitude representation can be handled at negligible overhead which, for typical filter coefficient-sets, is shown to offer significant benefits to both energy consumption and throughput.

Transistor level simulations show energy consumption to be lower than in previously reported designs.

1. Introduction

Achieving low power dissipation in VLSI systems has become a focus of current research offering not only extended battery life in mobile applications, but also improved system reliability through reduced operating temperatures. Approaches to power minimization span the design hierarchy from the algorithmic level down to the process technology [1]. This paper addresses power minimization in dedicated FIR filters at the architectural and circuit levels through reductions in circuit activity and switched capacitance. Much work on low-power design has been based on retaining high throughput whilst reducing power. In such contexts an approach similar to high-speed design can be employed, speed-gains being traded for a reduction in supply voltage with an attendant quadratic reduction in power [1]. Many such approaches incur an additional hardware overhead (e.g. pipeline registers) whose energy cost is, hopefully, more than compensated for by the benefits of the supply reduction made possible.

For moderate to low throughput applications, and in those in which supply voltage reductions are not possible, the emphasis of designing for low power falls less on the power required to deliver a given computational speed, but rather on the absolute energy-consumption associated with a given computation. In such cases, speed might often be considered a secondary design goal. Asynchronous circuits have the inherent potential to achieve lower energy consumption than synchronous implementations partly due to the absence of a global clock signal. They also have other advantages such as low electromagnetic emissions – an attractive property in radio-based applications. Moreover, because an asynchronous framework potentially allows data-

This work was supported by a grant from the Engineering and Physical Sciences Research Council (EPSRC), UK.

1522-869X/01 $10.00 © 2001 IEEE

dependent delays to be exploited, worst-case performance becomes a less important design constraint.

In many DSP algorithms including filters, multiplication is often not only the computational bottleneck but also an energy-expensive operation. Combinational multipliers, being high-depth circuits are often subject to much wasted energy due to spurious transitions (glitches). Several techniques have been shown to reduce glitches including latch insertion [2], delay balancing [3][4], the use of tree structures [1][4] and 4:2 counters [5]. Alternatively, the use of dynamic-logic - an inherently glitch-free logic style - has been shown to be particularly effective when under asynchronous control [6][7].

With static CMOS logic, necessary (as opposed to spurious) output switching activity occurs whenever the results of the previous and present computations differ. Therefore the energy consumption of a circuit such as a multiplier, is a function not only of the current operand values but also their previous values. For this reason, an effective method of reducing power consumption of DSP circuits can be to increase the similarity between successive computations. Such approaches include coefficient reordering [8] and permuted-differences [9].

Choice of number representation can also have a significant impact on the similarity between successive computations [1]. With a two's complement (2'sC) data-representation, a signal perturbation from +1 to -1, requires all bits but the LSB to change state. By contrast, such a signal swing in sign-magnitude (SM) representation requires only the sign-bit to change. Consequently, signals with a high proportion of low amplitude zero-crossings can produce less switching activity in SM rather than 2'sC. Such signals are commonly found in many DSP environments and several authors have reported energy benefits from SM [10][11]. In practice these benefits can be difficult to realize since the addition of signed numbers is considerably more complex in SM than 2'sC.

With dynamic logic implementations, the energy dissipated within the circuit (as distinct from the energy required to switch its operands themselves) is largely independent of the previous operand values since the circuit is precharged before each operation. This would appear to suggest that optimizing computational similarity has little to offer in dynamic implementations. In [12] the authors reported a strategy for designing very low-power VLSI FIR filters combining a Single Bit-Plane architecture with the energy-saving Conditional Evaluation dynamic-logic technique. In this paper we show that despite its dynamic-logic circuitry, this filter is able to exploit, not the similarity between successive computations but rather, a 'static' property of sign-magnitude numbers. Specifically, a set of filter coefficients can be represented in SM with, on average, a lower *'1-density'* (here defined as the proportion of non-zero-bits in the representation) than in the case of a 2'sC representation. Since, by virtue of the Conditional Evaluation implementation, the energy consumption of the filter is highly sensitive to the coefficient-set 1-density, a significant energy saving is obtained.

Furthermore, we show that by using the Single Bit-Plane architecture advocated in [12], the combination of a bit-serial coefficient format together with a sign-magnitude coefficient representation enables the sign-bits to be retained between successive computations. This has significant implications for improving average filter throughput as well as reducing activity in the coefficient storage circuitry with attendant energy benefits. (Note: the filter's input data-stream is retained in two's complement format thereby avoiding the complexities of sign-magnitude addition).

These filters exhibit an interesting property in that the delays associated with 'slow' coefficients (i.e. those with a high 1-density) can be offset by 'fast' coefficients of which, in a typical FIR filter, there will normally be a majority. For a given coefficient-set, therefore, the total sample computation delay is largely constant, being proportional to the coefficient-set 1-density. As a result, the low-power benefits of this fundamentally asynchronous strategy can

be obtained in synchronous environments provided that the coefficient-set 1-density is constrained below some predetermined maximum.

The contribution of this paper is to show that the proposed techniques result in highly regular FIR filter structures which, to the best knowledge of the authors, can deliver significantly lower energy consumption than other comparable reported designs.

Section 2 reviews the design of the dynamic logic, Single Bit-Plane filter. Section 3 investigates the correlation between the 1-density of the coefficients and the energy consumption of the filter. Section 4 assesses the theoretical benefits of sign-magnitude coefficient-representations and how these can be translated into real energy savings. Finally, conclusions are presented in Section 5.

2. Data dependent computation for FIR filtering

2.1 Architecture

Recently, circuits for very low power asynchronous multiplication and accumulation have been reported in which a significant proportion of the energy savings derive from the use of a data-dependent carry-save array [13], implemented using Conditional-Evaluation (CE) [7][14]. For application to digital filtering, this strategy is appropriate in situations where both operands to the carry-save array (CSA) are variables; these include programmable and adaptive filters as well as those using one or more multiplexed multipliers. The use of CE in dedicated FIR filters has been proposed by the authors in [12].

In the data-dependent CSA each cell has no AND gate for bit-product generation but is instead fitted with a pair of multiplexers (MUXs) which, when a row's bit-product is zero, pass on unaltered, the sum and carry inputs from the row above to the row below. For rows whose bit-product is non-zero, the MUXs select the adder outputs to feed to the row below. Since the propagation delay through the bypass path is less than that through the adders, values of MR which are heavily populated with zeros produce less delay through the array than those heavily populated with ones. A comparison of the standard and data-dependent architectures is shown in Figure 1.

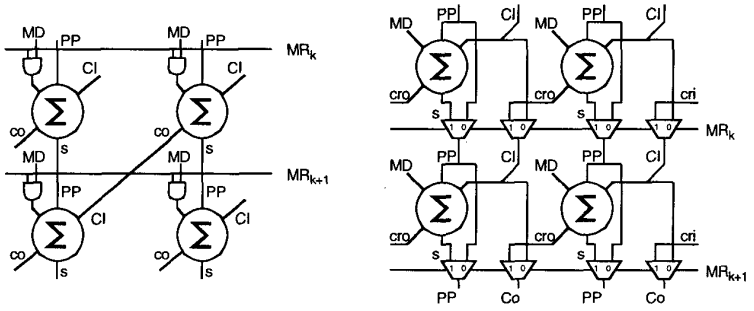

Figure 1. Carry-save array architectures: standard (left) and data-dependent (right)

The data-dependent architecture has the additional property that a significant amount of the logic becomes redundant in proportion to the number of zeros in the multiplier operand 'MR'. In a low-power application, this property can be exploited by inhibiting circuit activity within the redundant logic, as a function of input data. In this way, energy-consuming computations can be confined to those rows in which addition is required i.e. on average half the rows, assuming equiprobable ones and zeros in MR.

A horizontal slice taken through such a carry-save array can be used as a building block for many low-power asynchronous DSP structures. For example, a multiplier is composed of a parallelogram shaped stack (assuming equally weighted bits are arranged vertically) of slices. By registering the multiplier's outputs and feeding back into the stack, a concurrent multiplier-accumulator (CMAC) is obtained [14]. With appropriate peripheral circuitry for handling coefficient and data storage, multiplier and multiplier-accumulator based FIR filters can be constructed.

As an alternative to multiplier and multiplier-accumulator based architectures, FIR filters can be composed of carry-save slices as proposed in [15]. Here, the summation of bit-products is reordered such that all those of equal weight are added in adjacent rows of carry-save adders forming a 'bit-plane'. Appropriate weighting of coefficient bits is achieved by a hard-wired shift at the connections between bit-planes resulting in a structure using one row of carry-save adders per bit, per coefficient. Such filter architectures have been successfully implemented in very high throughput applications [16] by extensive pipelining of the array.

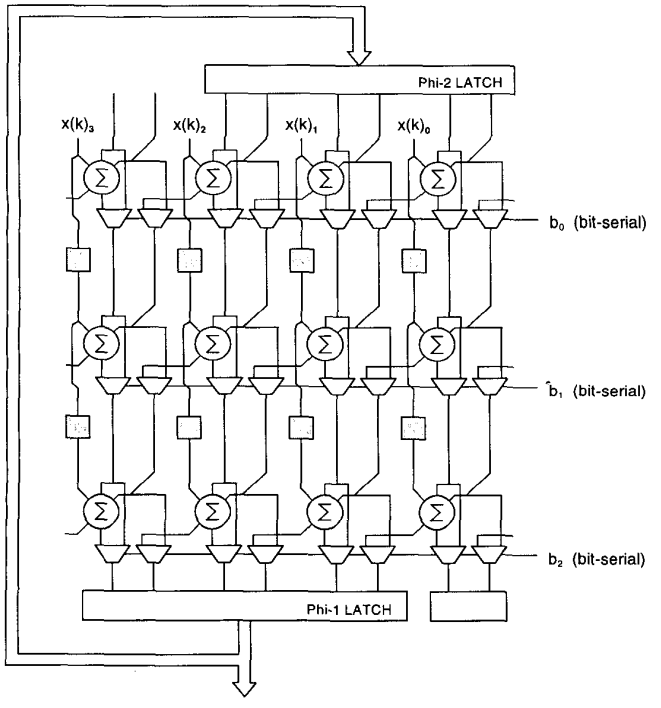

Figure 2. Architecture of data-dependent single bit-plane filter. The latches comprising the sample-data-store are shown shaded.

For moderate throughput filters an architecture is possible using a single combinational bit-plane which is multiplexed M-times (assuming M-bit coefficients). This Single Bit-Plane (SBP) requires one carry-save slice per coefficient, the coefficients being supplied concurrently in a bit-serial fashion. Use of the data-dependent carry-save slice yields the architecture shown in Figure 2 (illustrated here for 3 coefficients and 4-bit data samples). As with the CMAC design

the data emerging from the bottom of the array is registered and fed back to the top, albeit with a 1-bit shift to allow for the increasing significance of successive coefficient bits.

It is interesting to note that with this structure the data-sample store, shown (shaded) in Figure 2, can be interleaved into the array itself with the benefit that its interconnect becomes local. These data-store latches are configured as an asynchronous tapped shift register for providing the delayed data-samples required in direct-form filters. This approach offers good energy efficiency since a single shift operation delivers M data samples. By contrast, a register-file-based data-store would require M discrete read operations to deliver the same amount of data and would consume significantly more energy.

Each slice is associated with a single coefficient supplied from a (1-bit) register file. Consequently, a filter of arbitrary order can be constructed using a regular stack of filter-slices - a feature which recommends the SBP as a structure appropriate for design automation.

2.2 Implementation

In a static-logic carry-save implementation, inhibiting circuit activity as suggested above generally requires the insertion of additional latches to prevent input transitions propagating through the redundant logic [17]. These latches not only introduce additional delays but also represent a significant energy cost. On the other hand, by using a dynamic-logic data path, computation can be inhibited by suspending transitions on the precharge/evaluate lines. This strategy, termed Conditional Evaluation in [7], is an almost overhead-free mechanism for 'shutting down' the redundant logic and has been shown to contribute heavily to energy savings; an 8-bit row in bypass mode using approximately 1/4 of the energy of a row in which addition is required (in 0.35μm technology).

Use of dynamic logic also offers the known benefits to speed and area as well as other potential advantages for low-power. Specifically, its reduced input-capacitance means that circuit activity draws less energy from the supply. Furthermore, spurious transitions, which in array multipliers can account for as much as 50% of the energy consumption [3] are eliminated, without the need for techniques such as delay balancing [3][4]. In general however, two factors tend to undermine the energy benefits of dynamic logic. The first is the need to charge and discharge the precharge/evaluate lines. The second is the increased probability of output activity, which, despite the reduced load capacitances, puts dynamic logic at a disadvantage for some logic functions. For other functions the increase in activity is small [1]. In this application, the benefits outweigh the drawbacks, ease of inhibiting activity giving particular weight to the choice of dynamic logic for the full-adders. In contrast, the MUXs are never required to be disabled and a static-logic MUX implementation was adopted.

To avoid the race problem, dynamic-logic carry-save arrays usually employ differential circuitry [13] despite its relatively poor energy efficiency [1]. Alternatively, the power benefits of single-ended logic can be obtained by using self-timing to avoid the race [6]. This was the approach adopted here. In our CSA cell, the full-adder comprises a cascade of two dynamic n-blocks, one which evaluates the carry-out signal and one which uses carry-out to evaluate the sum output. The MUXs, on the other hand, are used whether addition is required or not and therefore a static logic implementation was adopted for these.

Merging of the sum and carry vectors involves an addition operation requiring complete carry-propagation. This is carried out in a vector-merging or Carry-Resolution Adder (CRA). Being a slower and more energy consuming operation than carry-save addition, vector merging is performed relatively infrequently - as much as possible of the filter's computation is carried out in carry-save arithmetic, resorting to vector merging only at the final filter output. In performance-critical applications the CRA will either be pipelined, if the additional latency can

be tolerated, or use a high-speed technique such as carry look-ahead. In our application, low-energy is the primary design goal and vector merging is carried out in a ripple-carry adder chosen for its low switching activity. In order to offset the long worst-case settling time of this circuit, completion-detection circuitry is added, introducing some further data-dependency in sample computation delay. The CRA uses a similar full-adder to that used in the array. This lends itself to the adoption of the dynamic-logic AMCD method of completion detection [18] a technique offering good energy efficiency.

With filters constructed using this approach both energy consumption and sample-computation delay are heavily dependent on the *total* number of ones in the filter coefficient-set. A coefficient-set with a reduced 1-density will require the rows of the carry-save array to perform additions less frequently and will therefore have an average-power advantage. Similarly, from a speed perspective, a reduced 1-density translates into an improvement in performance.

The filter designer can therefore search for a coefficient-set with a 1-density meeting the multiple design objectives of filter characteristic, throughput and energy budget.

2.3 Area

In comparison with a standard, static-logic carry-save array, the layout of the data-dependent dynamic circuit benefits from the elimination of the bit-product generation circuitry and from the significant reduction in transistor-count and good packing density associated with the domino logic. On the other hand, the bypass-MUXs used in each cell require additional area. The overall CSA cell area of the two approaches is therefore similar.

The proposed array does however, incur a silicon overhead for the conditional-evaluation controller circuit required for each of its rows. The function of this controller is to perform gating of the row's precharge/evaluate transitions and to provide the matched delays necessary for self-timed race elimination in the dynamic logic. This cell is dominated by the delay circuit, which is equivalent to a worst-case hard-wired full adder. The area overhead is therefore equivalent to approximately one full-adder per row – some 10% for 8-bit data and 2 guard-bits.

3. Energy consumption, throughput and 1-density

3.1 Energy Consumption

A number of transistor level simulations of a 7^{th}-order SBP filter with 8-bit coefficient and data wordlengths and two guardbits were carried out in order to ascertain the relationship between the 1-density of the coefficient-set and the energy consumption. These filters are based on a 0.35 μm technology with a 3.3V supply. Layout parasitics were not included. A spread of coefficient-set 1-densities was used ranging from 0.5 (half ones) to 0.125. 1-densities below this range cause significantly reduced activity in the circuits peripheral to the CSA and result in very low energy consumption. As discussed in Section 4 such situations are not typical and are therefore not included in this analysis; the selected range covers the region of most interest, representing the 1-densities likely to occur in a real filter. The same uniformly distributed random input data stream was used in each simulation and the resulting sample computation energy was plotted against 1-density as shown in Figure 3.

Each computation incurs energy overheads which are largely data-independent. Detailed simulations of the filter's components enable these to be estimated at 36pJ, breaking down as 6pJ for shifting the data-store, 4pJ for accessing the coefficient register-file, 4pJ for merging the final carry and sum vectors and 22pJ for registering. The remainder of the energy consumption

is dissipated in the CSA and can be expected to correlate with the 1-density of the coefficient-set. This expected approximately linear correlation is confirmed in Figure 3.

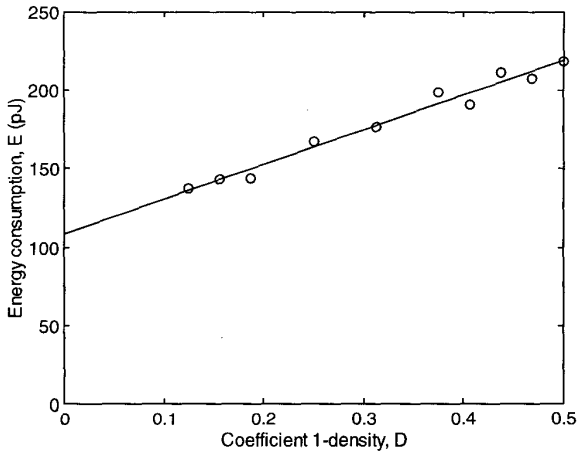

Figure 3. Correlation of energy and coefficient 1-density for 8x8x8 SBP filter

Each point represents a single case of the given 1-density. Energy consumption is not solely dependent on 1-density and some measure of the variance can be seen from the data points. The long simulation times of the circuit (≈3h per point) preclude the generation of a statistically precise data-set. However, all the measured points lie within ±5% of the least-squares best-fit straight line shown in Figure 3. By inspection this line can be approximated by the equation:

$$E=(3.06D+1.5)*72pJ$$

in which E and D are energy and 1-density respectively. The gradient of 3.06 arises from the ratio between the evaluate and bypass row-energies (approximately 4:1). The intersection on the energy axis represents the row bypass-mode average energies (≈72pJ) plus the 36pJ overhead.

For typical filters with more than 8 coefficients, the energy overhead will become less significant as the contribution of the carry-save array becomes increasingly dominant. The normalized line equation therefore approaches:

$$E=(3.06D+1).E_{min}$$

where E_{min} is a constant of proportionality representing the energy cost of all rows in bypass. Furthermore, since the number of additions performed by a row is proportional to the number of ones in its coefficient, it is reasonable to infer that the energy consumption will scale approximately linearly with the number of coefficient bits. These inferences suggest that the linear relationship shown in Figure 3 is representative of larger filters albeit with a less significant overhead and the energy axis scaled appropriately.

3.2 Throughput

The impact of 1-density on throughput can be estimated from the performance of the filter's sub-circuits. Detailed simulations show the propagation delays of the CSA rows to be 0.25ns and 0.9ns in bypass and addition modes respectively. Register and precharge delays add ~1.7ns for each pass through the array and the final adder incurs a fixed delay overhead of ~1.3ns plus a data-dependent component of ~0.5ns per carry-ripple pair. In this filter, a sample computation

comprises 8 passes through the array plus a final addition. Assuming worst case delays in the adder, the sample computaion delay, τ_{scd} is a linear function of the 1-density ranging from 33.4ns to 75.0ns and is given by the equation:

$$\tau_{scd} = (2.6D+2.09)*16.0ns$$

As with the energy equation, the gradient of 2.6 derives from the ratio of row-bypass and row-evaluate performance. For a filter with a large number of coefficients, delays are dominated by the CSA the normalized equation approaches:

$$\tau_{scd} = (2.6D+1).T_{min}$$

where T_{min} is a constant of proportionality representing the delay of all rows in bypass mode. The above correlations establish that improvements in energy consumption and throughput can be achieved by using a coefficient-set with a reduced 1-density.

DSP systems typically rely on a fixed sample-rate and for this reason, DSP has often been considered an unpromising area for the application of asynchronous techniques. With our filter, however, the time required to process any single coefficient is a linear function of its 1-density. Therefore, in processing a *coefficient-set,* fast coefficients compensate for slow coefficients and the filter's worst-case sample computation delay can be determined a priori from knowledge of the coefficient-set. The filter can therefore take advantage of such data-dependencies whilst operating within a fixed-sample-rate environment.

4. Comparison of coefficient representations

An inherent feature of the Single Bit Plane is its facility to perform a subtraction in every row. This is necessary because each row must be able to handle its coefficient's sign-bit (unlike multiplier based filters in which all coefficient sign-bits are dealt with in the last row of the array). The cost of this feature is small, requiring principally that a MUX be included in the data-store to select inverted data at the appropriate time. However, the potential advantages of such a facility are significant since the use of other numerical coefficient representations is made possible, including Canonic Signed Digit (CSD) and Sign-Magnitude (SM).

Although a CSD coefficient-representation is known to be optimal in terms of minimizing the number of non-zero bits, two particular considerations argue against its adoption here. Firstly, since each CSD bit has an associated sign-bit, coefficient storage is doubled with an attendant energy cost. (Alternatively, a real-time CSD encoder is required with probably higher cost). Secondly, the add/subtract control on each row of the array will incur additional activity and hence energy consumption.

By contrast, SM coefficient-representations have a single sign-bit common to all other bits. Since each CSA row handles one complete coefficient, its sign-bit can be latched, remaining in place until the coefficient-set changes - a relatively infrequent occurrence. In this respect, architectures using bit-serial-coefficients, such as the Single Bit-Plane, are at a distinct advantage over single-multiplier based filters with bit-parallel coefficients since, in the latter, a relatively energy-expensive sign-bit transition potentially occurs on every multiplication cycle.

In any case, a SM coefficient representation potentially offers some advantage over 2'sC. This stems from the fact that processing an M-bit coefficient requires the summation of M, bit-products in 2'sC but only M-1 in SM. (By contrast, the data-stream, and hence bit-products and partial-products, are arguably best represented in 2'sC thereby avoiding the complexities of SM addition).

The architectural modifications necessary to operate the SBP filter with SM rather than 2'sC coefficients are small. With 2'sC, all sign bits are accessed from the coefficient-store simultaneously and therefore all CSA rows are set to select the data-store complement, globally.

For SM coefficients, such control is not global but row-specific according to the sign-bit of the row's coefficient. The sign-bits are therefore stored external to the coefficient-store (which can now be made one register smaller) but other modifications are minor.

For typical FIR filter applications, SM offers yet further benefits because realistic coefficient-sets usually contain a significant proportion of small negative numbers particularly at the extremities of the impulse response. Since small negative numbers have a large number of ones in 2'sC but considerably fewer in SM the latter representation is likely to incur a smaller 1-density.

In order to assess this particular advantage, the 1-density of various typical filter coefficient-sets was compared. Filters using 63, 12-bit coefficients were designed using the FIR1 function in MATLAB. Coefficients were then scaled such that the largest positive coefficient is +2047 (i.e. the maximum positive value that can be represented in 12 bits) and rounded to the nearest integer. The integers were then converted to both 2'sC and SM binary representations and their respective 1-densities measured. Results for some 50 different filters including low-pass (LPF), high-pass (HPF), band-pass (BPF) and band-stop (BSF) characteristics, show that for 2'sC coefficient-sets, 1-densities of between 0.41 and 0.49 are typical. By contrast, typical 1-densities for SM representations lie in the range 0.22 to 0.31.

Some exceptions are observed; filters in which a high proportion of the coefficients are positive show similar 1-densities in both representations (e.g. extremely narrow band low-pass filters). Similarly, low/high-pass filters with cutoffs at 0.5 and band-pass/stop filters centered at 0.5, all of which have a high proportion of zero-valued coefficients, show less benefit from SM representation. Nevertheless in every case, SM results in a smaller 1-density than 2'sC.

Table 1. Coefficient 1-densities for particular filters

	Cutoff		2'sC	SM	Reduction
	lower	upper			
LPF	0.2	-	0.459	0.295	0.164
	0.5	-	0.271	0.144	0.127
HPF	0.2	-	0.388	0.205	0.183
	0.7	-	0.499	0.290	0.209
BPF	0.1	0.3	0.472	0.303	0.169
	0.5	0.9	0.438	0.255	0.183
BSF	0.3	0.4	0.480	0.205	0.275
	0.3	0.7	0.226	0.142	0.084

A few examples of filter comparisons, including some of the special cases mentioned above, are shown in Table 1. Also shown is the reduction in 1-density afforded by a SM representation. Cutoff frequencies are normalized to half the sample frequency. Overall, for the majority of the filters, use of SM causes the 1-density typically to reduce by approximately 0.16 from 0.43 to 0.27.

The equations relating energy and 1-density given above can be used to translate this typical reduction into an energy saving. In the case of the simulated filter, the sample computation energy drops by 18%. For a larger filter following the E=3.06D+1 line, these density reductions translate to a typical energy saving of 21%. When allowance is made for the one less bit-product produced in SM the energy savings become approximately 29% for both filters (assuming 12-bit coefficients in the larger filter). Therefore, for the simulated filter with random data the average energy to perform the equivalent of eight, 8x8-bit multiplications (i.e. to compute a sample) reduces to 147pJ (0.35μm CMOS process, Vdd=3.3V).

Since long run-times have precluded simulations with real data, it should be noted that these energy figures are based on a random data stream. With more realistic data streams, it can be argued that expected correlations between successive samples will give rise to less activity at the inputs to the CSA and within the data-store latches. Furthermore realistic data streams are likely to be subject to substantially reduced activity on several of the MSBs [1]. These considerations suggest that the figure of 147pJ may well reduce significantly in a real application.

5. Comparisons with other filters

A recent paper describing a low-power (synchronous) FIR filter [19] claims more than a fivefold improvement in energy consumption over previous work. For comparison with our filter, the energy figure reported in [19] can be normalized for technology, wordlength and supply voltage differences, using the algorithm advocated therein. Such normalization shows a dissipation of 61pJ per multiplier in an 8x8-bit implementation. With our filter, after an allowance of 100% for parasitics (layout extracted simulations of our circuits indicate this figure to be a realistic estimate), 37pJ per equivalent multiplication is dissipated - 39% less energy than [19].

The sample computation delay is also heavily dependent on the number of required row additions and will also benefit from the lower 1-density of a SM coefficient representation. In our filter, 1-densities of 0.43 and 0.27 give *worst-case* delays of 51.3ns and 44.6ns respectively. The use of SM coefficients will therefore produce a throughput increase from 9.75MHz to 11.2MHz after a 100% allowance for parasitics – an improvement of 15%. In a large filter the CSA is again the dominant component but for delays, the ratio between evaluate and bypass modes is 3.6:1 making the use of SM slightly less beneficial to throughput than to energy. For such a filter the throughput improvement approaches 24%. Since the filter of [19] uses several multipliers to produce an output sample, throughput comparisons are not meaningful.

Comparisons with other asynchronous filter implementations are made difficult by the scarcity of published examples. The recent asynchronous design of [20], has a fixed filter-response using coefficients selected from those which contain a maximum of 3 ones (and averaging less than 2 ones per coefficient). Its function is therefore not strictly comparable with our full-resolution programmable filter. Nevertheless, after normalization using inferred wordlength-approximations, (actual wordlengths are withheld) the energy consumption is likely to be in the region of 40pJ per multiply-accumulate equivalent - broadly similar to that of our filter but without the programmability or regularity.

6. Conclusions

The data-dependent properties of the Single Bit-Plane FIR filter implemented using the Conditional Evaluation technique have been examined and quantified in relation to their correlation with the binary properties of the coefficient-set. An approximately linear relationship between 1-density and both throughput and energy consumption has been identified.

The facility of this structure to handle Sign-Magnitude coefficients at very little overhead, combined with its bit-serial coefficient-format, eliminates activity on the coefficient sign-bits. This property and the reduced 1-density associated with the SM representation, produces improvements in energy consumption and throughput of typically 29% and 15% respectively in the simulated filter. Larger filters with a more dominant carry-save array are predicted to benefit more from the use of SM coefficients.

The proposed synergy of filter-structure, data-dependent architecture, conditional-evaluation based domino implementation and coefficient representation yields a filter with, to the best knowledge of the authors, lower energy consumption than other published designs.

Whilst the data-dependency in terms of propagation delay is best exploited within an asynchronous framework, worst case delays are largely constant for a given 1-density and fairly easily determined from the coefficient-set. This makes the design applicable to both synchronous and asynchronous environments.

7. References

[1] A. Chandrakasan, S. Sheng and R.W. Brodersen, "Low-Power CMOS Digital Design" *IEEE Journal of Solid-State Circuits*, Vol. SC-27, pp. 473-484, 1992

[2] C. Lemonds and S.S. Mahant-Shetti, "A low power 16 by 16 multiplier using transition reduction circuitry," *Proc. International Workshop on Low Power Design*, pp.139-142, 1994

[3] S.S. Mahant-Shetti, P.T. Balsara and C. Lemonds "High performance low power array multiplier using temporal tiling," *IEEE Trans. VLSI Syst.* Vol. 7, (1), pp. 121-124, Mar. 1999.

[4] T. Sakuta, W. Lee and P.T. Balsara, "Delay Balanced Multipliers for Low Power/Low Voltage DSP Core," *Proc. Int. Symp. Low Power Electron. and Design*, pp. 36-37, 1995

[5] V.G. Moshnyaga and K. Tamaru, "A comparative study of switching activity reduction techniques for design of low-power multipliers," *Proc. Int. Symp. Circuits and Systems (ISCAS'95)*, pp.1560-1563, 1995.

[6] G.E. Sobelman and D. Raatz, "Low-power multiplier design using delayed evaluation," *Proc. Int. Symp. Circuits and Systems (ISCAS'95)*, pp. 1564-1567, 1995

[7] V.A. Bartlett and E. Grass, "A Self-Timed Multiplier using Conditional Evaluation," *Proc. PATMOS'98, 8th Int. Workshop on Power, Timing, Modeling, Optimization and Simulation*, Lyngby, Denmark, pp. 429-438, 1998

[8] K. Masselos, P. Merakos, T. Stouraitis, C.E. Goutis, "A Novel Methodology for Power Consumption Reduction," *Proc. Int. Symp. Circuits and Systems (ISCAS'98)*, vol 6, pp. 199-202, 1998

[9] N. Sankarayya et. al "Algorithms for Low-Power and High Speed FIR Filter Realization Using Differential Coefficients," *IEEE Trans. Circuits Syst. II* vol.44, no. 6, pp. 488-497, 1997

[10] A.P. Chandrakasan, and R.W. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits" *Proceedings of the IEEE*, Vol. 83, No.4, pp. 498-523, 1995

[11] M. Zheng and A. Albicki "Low Power and High Speed Multiplication Design through Mixed Number Representations," *Proc. Int. Conf. Computer Design ICCD'95* pp. 566-570, 1995

[12] V.A. Bartlett and E. Grass, "A Low-Power FIR Filter Exploiting Data-Dependent Operation," *Proc. PATMOS'99, 9th International Workshop on Power, Timing, Modeling, Optimization and Simulation*, Kos Island, Greece, pp. 231-240, 1999

[13] D. Kearney and N.W. Bergmann, "Bundled Data Asynchronous Multipliers with Data Dependent Computation Times", *Proc. ASYNC'97, 2nd Int. Symp. on Advanced Research in Asynchronous Circuits and Systems*, pp. 186-197, 1997

[14] V.A. Bartlett and E. Grass, "A Low-Power Concurrent Multiplier-Accumulator using Conditional Evaluation", *Proc. 6th IEEE Int. Conf. Electronics, Circuits and Systems*, Pafos, Cyprus, 1999

[15] P.B. Denyer and D.J. Myers, "Carry-Save Arrays for VLSI Processing," *Proc. VLSI'81 First Int. Conf. On VLSI*, pp. 151-160, Edinburgh, 1981

[16] T.G. Noll, "Carry-Save Architectures for High-Speed Digital Signal Processing" *Journal of VLSI Signal Processing*, 3, pp. 121-140, 1991

[17] M. Alidina, J. Monteiro and S. Devadas, "Precomputation-Based Sequential Logic Optimization for Low Power," *IEEE Transactions on VLSI Systems*, Vol. 2, (4) pp. 426-436 Dec. 1994

[18] V. A. Bartlett and E. Grass, "Completion-Detection Technique for Dynamic Logic," *Electronics Letters*, vol. 33, no. 22, pp. 1850-1852, Oct. 1997.

[19] C.J. Nicol and P. Larsson, "Low Power Multiplication for FIR Filters," *Proc. Int. Symp. Low Power Electronics and Design*, pp. 76-79, 1997

[20] L.S. Nielsen and J. Sparso, "Designing asynchronous circuits for low-power: An IFIR FIR filter bank for a digital hearing aid," *Proc. of the IEEE*, Vol. 87 (2), pp. 268-281, 1999