

Architectural trade-offs in the design of low power FIR filtering cores

A.T. Erdogan, E. Zwyssig and T. Arslan

Abstract: There is a continuous drive for methodologies and approaches of low power design. This is mainly driven by the surge in portable computing. On the other hand, the design of low power systems for different portable applications is not a simple task. This is because of the number of constraints that influence the power consumption of a device. In addition to issues of performance and functionality, there is a need to satisfy strict test coverage constraints. The authors investigate the impact of DSP architectural realisation, multiplier type, and the choice of number representation on the overall power consumption of DSP devices. Work in the literature so far has concentrated on the effect of these on a part or a section of a DSP system. Furthermore the effect of DfT circuits on the overall performance is studied. A hearing aid device is considered as an example of a system with strict power/area constraints. It is shown that the choice of multiplier architecture and number representation should be carefully considered when specific DSP architectural choices are made. The results are demonstrated with a number of specially designed DSP architectures for the implementation of FIR filtering algorithms on hearing aid devices.

1 Introduction

FIR filtering algorithms such as subband decomposition, noise reduction and echo cancellation are executed repetitively in DSP systems such as hearing aids. Therefore an effective filtering architecture is a key part of a DSP system. The basic FIR filter is represented by the following equation:

$$y_n = \sum_{m=0}^{M-1} h_m \cdot x_{n-m} \quad (1)$$

From this equation a number of key components/operations can be identified. These are as follows:

- a multiply-accumulator (MAC),
- several forms of memory such as those required for the sample values x_{n-m} (RAM) and the coefficient values h_m (ROM),
- a storage cell for the filter output value y_n ,
- a controller which schedules the different components.

Figure 1 illustrates the principle data flow between the above components in a block diagram. Among these components the MAC is the most critical, which in turn accommodates the multiplier.

Different multipliers have been explored in the past and their architectures and layouts analysed with regard to power dissipation, area consumption and circuit delay [1].

Meier *et al.* [2] compared array multipliers with Wallace tree multipliers. Keane *et al.* [3] verified the impact of data characteristics and different multiplier architectures for low power dissipation. A power minimisation scheme for multipliers is presented by Fried [4] and a power-efficient multiply-accumulate design for filters can be found in Farag *et al.* [5]. A hearing aid specific low power multiply-accumulate scheme, particularly for FFT butterfly and filter structures, has been proposed by Moller *et al.* [6]. Nielsen *et al.* [7] have investigated a number of design issues relating to the implementation of low power asynchronous FIR filter circuits including those targeting hearing aid applications. In [8] the authors demonstrate that low power FIR filters can be constructed from the concurrent multiplier-accumulator circuits.

All of the above work consider performance issues regarding the MAC or multiplier alone without analysing the overall power consumption of a system. However, for a true performance evaluation the overall system architecture together with the design of the key components must be considered. For example, a linear phase FIR filter may be implemented utilising a direct form (DF) or a folded direct form (FDF) structure. The coefficients for such a filter are symmetric around the midpoint of the impulse response. A

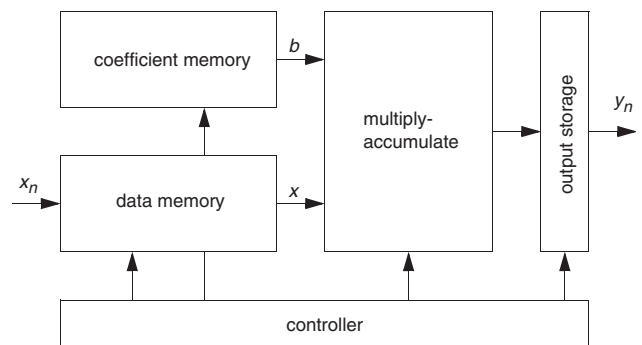


Fig. 1 Filter block diagram

© IEE, 2004

IEE Proceedings online no. 20040227

doi:10.1049/ip-cds:20040227

Paper first received 23rd April 2002 and in revised form 25th February 2003

A.T. Erdogan and T. Arslan are with the Department of Electronics & Engineering, The University of Edinburgh, The King's Buildings, Mayfield Road, Edinburgh EH9 3JL, Scotland, UK

E. Zwyssig is with Wolfson Microelectronics Ltd., 20 Bernard Terrace, Edinburgh EH8 9NX, Scotland, UK

linear-phase filter can thus be implemented efficiently by a FDF structure, where two data samples are added before being multiplied with the corresponding coefficient. Therefore, using a FDF structure will reduce the number of multiplications by half at the expense of additional hardware. A dual-port memory and an additional adder will be required to access two data samples at a time and to add them together prior to the multiplication stage. In addition, the use of different number representations will introduce some overheads. This is mainly due to the need for data converters before and after the multiplication process.

Furthermore, for an FIR filter core to become a viable product it must have high fault coverage and hence must incorporate adequate design-for-testability (DfT) circuitry. For example, a scan path can be used for small blocks such as counters and built-in self-test (BIST) for more complex ones such as memory and MAC. However, DfT circuits introduce an overhead in terms of the additional circuitry. Traditionally, high fault coverage and fast test time have been the main objectives of DfT designs. While these objectives still remain, a new design objective, namely low power dissipation, is becoming especially important. For these reasons, the test circuitry is crucial to the power and area performance of an FIR filter core.

In this paper, we investigate the impact of DSP architectural realisation, multiplier type, and the choice of number representation on the overall power consumption of DSP devices. We consider a hearing aid device as an example of a system with strict power/area constraints. Furthermore we study the effect of DfT circuits on the overall performance, using typical compact BIST circuitry used in area/power critical applications such as hearing aids. We show that the choice of multiplier architecture and number representation should be carefully considered when specific DSP architectural choices are made. This type of investigation is not presented in the literature so far. Our results are demonstrated with a number of specially designed DSP architectures for the implementation of FIR filtering algorithms in hearing aid devices.

2 Implementation

2.1 Multiplier cores

Although multiplication is now available from synthesis tools it is still worthwhile to design multipliers from scratch. There are several reasons for this. First of all, designing a multiplier with a technology independent HDL makes it available for all future projects. No dependence on a tool vendor exists and the resulting design is known in detail. Low power modifications and operand size changes can be performed easily, i.e. a maximum level of maintainability is achieved. Secondly, different multiplication schemes such as Dadda [9], Wallace [10], Booth [11], modified Booth [12], and pre-add Booth [6] can be implemented and compared which is not possible with a multiplier provided from an EDA vendor. Finally, DfT features can be included in the design.

Availability of a wide variety of multiplication schemes and limitations of automatic synthesis tools make it difficult to select a good multiplier for a design. It is therefore necessary to design and compare a series of different multipliers. The following parallel multipliers have been designed and evaluated in terms of area, power consumption, and fault coverage:

- Wallace–Dadda multiplier
- Booth multiplier

- reduced glitch Booth multiplier
- redundant binary Booth multiplier
- pre-add Booth multiplier

The rest of this Section briefly describes the multiplier cores to clearly distinguish the details of the architectures designed and utilised in this work.

Wallace–Dadda multiplier: Wallace [10] and Dadda [9] have shown that a parallel multiplication can be executed more efficiently than in an array multiplier. Dadda suggested the use of carry–save adders (csa) to reduce the partial products with a compression factor of 3 to 2. Once only two partial products are left the final sum is generated with a carry–look-ahead adder. Wallace suggested a modification of the Dadda algorithm so that the amount of carry–save adders and the delay of the partial product reduction tree can be reduced further.

Booth multiplier: Booth [11] suggested a higher radix multiplication algorithm to design faster multipliers. In his approach, multiple bits of the multiplier input are scanned to generate fewer partial products of the multiplicand. For example, three bits $\{y_{i+1}, y_i, y_{i-1}\}$ of the multiplicand input of the multiplier are scanned simultaneously if radix-4 Booth encoding is used. The positive or negative multiplicand (B , $2B$, 0 , $-2B$, or $-B$) is then added to the partial product reduction tree, depending on the scanned triple.

Reduced glitch Booth (RG_Booth) multiplier: Fried [4] suggested a two-gate-delay implementation of the Booth encoder and partial product generator in order to balance the gate delays in the Booth multiplier and therefore reduce the glitches and hence the power consumption.

Redundant binary Booth (RB_Booth) multiplier: Carry–save adders are usually used to reduce the partial products with a compression factor of 3 to 2. However, better results can be achieved using a compressor with a compression factor of 4 to 2 or more. Compressors with higher compression factors can be implemented using redundant signed-digit number representation. In this work, a radix-2 signed digit set, defined as $\{-1, 0, 1\}$, is used for the implementation of the redundant binary multiplier suggested in [12].

Pre-add Booth (P_Booth) multiplier: Moller *et al.* [6] suggested using a pre-add Booth multiplier, a Booth multiplier that has a pre-adder integrated in the Booth encoder, for filters with symmetric coefficients. The P_Booth encoding requires an adder and some logic gates for the carry generation, in addition to the Booth encoder. The P_Booth encoding is complex and has an increased delay compared to the normal Booth encoding due to the carry propagate logic. The P_Booth encoder is also liable to glitches due to the carry propagation delay through several Booth encoders. The authors in [6] have examined five different partial product addition structures in terms of number of nodes, gate delays, and the average number of node transitions per multiplication. It has been shown that the number of transitions could be minimised by building a mixed adder tree with 3 to 2 and 4 to 2 compressors.

2.2 Multiply-and-accumulate

The MAC is the core cell of a digital signal processor which calculates the sum of an accumulated intermediate result and a product. The block diagrams of the basic MAC and the pre-add MAC units that have been implemented are shown in Fig. 2. The basic MAC cell contains a multiplier, an adder/subtractor, a register (accumulator), and a multiplexer. The functionality of the MAC is defined

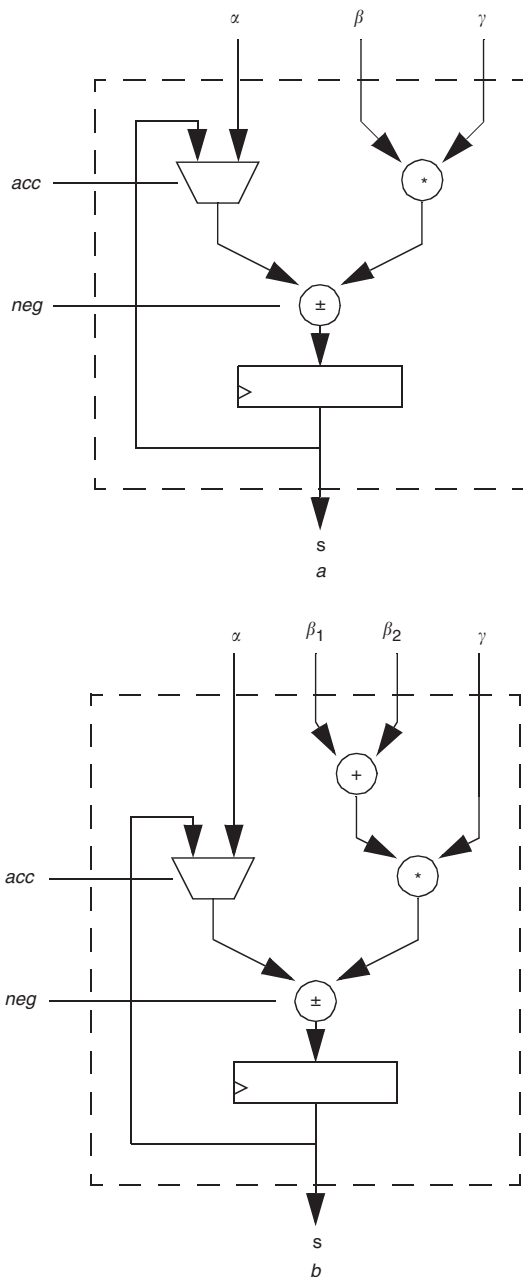


Fig. 2 Block diagram for the MAC units
a Basic MAC
b Pre-add MAC

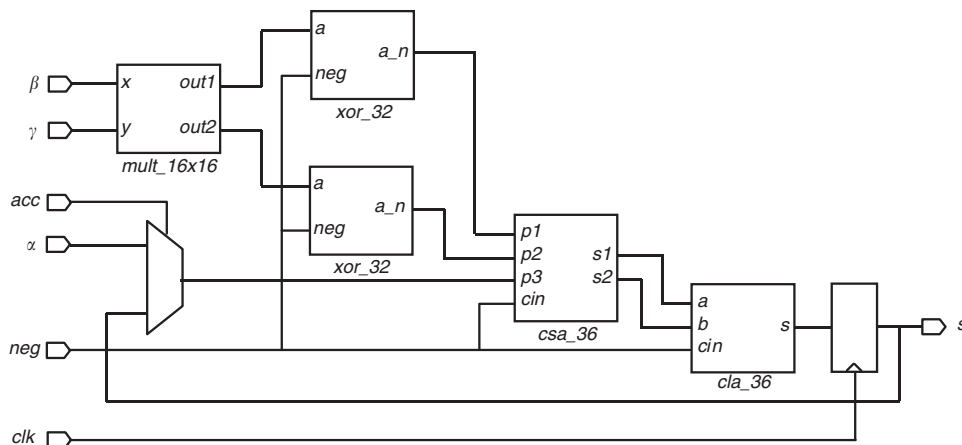


Fig. 3 Architecture for the MAC unit

as follows:

$$\begin{aligned} &\text{if } (acc) \\ &\quad s = s + (-1)^{neg} \cdot \beta \cdot \gamma \\ &\text{else} \\ &\quad s = \alpha + (-1)^{neg} \cdot \beta \cdot \gamma \end{aligned}$$

The implementation architecture of the MAC is depicted in Fig. 3. The $mult_{16 \times 16}$ determines the partial product of two 16-bit operands x and y . The actual product of x and y is the sum of the $mult_{16 \times 16}$ outputs $out1$ and $out2$ ($x \cdot y = out1 + out2$). The control signal, neg , determines whether the multiplier output is added to or subtracted from the multiplexer output which in turn is determined by the control signal acc . The multiplexer is used for switching between a new input value, α , or the stored accumulator value, s . The subtraction is realised by exclusive-oring the $mult_{16 \times 16}$ outputs with the neg signal and adding the neg to the resulting values through the carry-in inputs of the csa_{36} and cla_{36} circuits, effectively two's complementing the $mult_{16 \times 16}$ outputs $out1$ and $out2$. To reduce the likelihood of an overflow during the accumulation process the accumulator employs 4 guard bits, making it 36 bits wide. Therefore, the $mult_{16 \times 16}$ outputs are sign extended to 36 bits after they are exclusive-ored. A 36 bits carry-save adder (csa_{36}) is used to reduce the three inputs to two after which they are added through a 36 bit carry-look-ahead adder (cla_{36}). The final output is then stored into the accumulator.

This MAC architecture is also suitable for performing the multiplications using sign-magnitude (SM) number representations. It is well known that two's-complement (2'sC) number representation has a much higher switching activity than the SM representation [13]. However, SM addition and subtraction are complex operations to implement. For this reason SM is only used during the multiplication process in this work. Prior to a multiplication the data at both multiplier inputs are converted to SM. Then the multiplication is performed using the unsigned magnitudes, where the sign bits of the two operands are exclusive-ored to determine the control signal neg .

2.3 Digital filter implementation

This Section describes the FIR filter architectures implemented in this work. The generic architecture of a DF FIR filter is illustrated in Fig. 4. The coefficient memory consists of a ROM look-up table, an address counter and a data multiplexer. The data memory (RAM) is built around an array of latch banks (24×16 bits). In addition, it has a 1-to-24 address demultiplexer, a 24-to-1 data multiplexer, and BIST circuitry. The dual port RAM used for the FDF

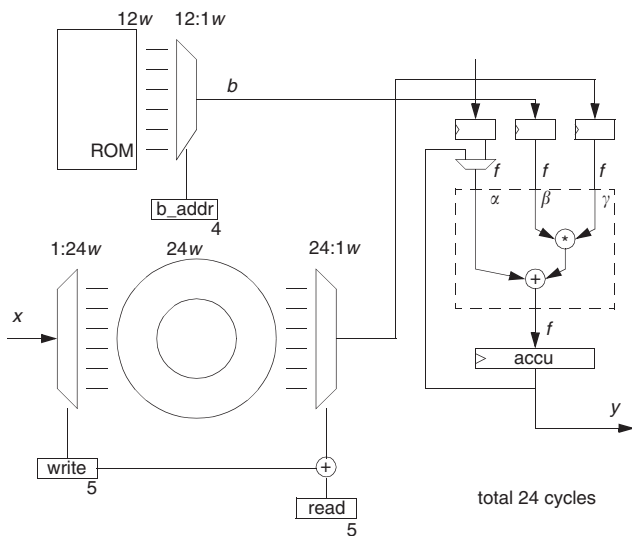


Fig. 4 DF FIR filter block diagram

implementations has an additional data multiplexer for the second output port. The MAC module contains a multiply-adder and an accumulator. The system controller is not shown here. Figure 5 shows the generic architecture of a FDF FIR filter. This looks very similar to the DF filter. However, two read ports for the x input sample memory and two read address counters and a pre-adder are required for the FDF FIR filter. Note that one input of the multiplier (γ in our example) requires a one bit increase in operand size to maintain the full precision.

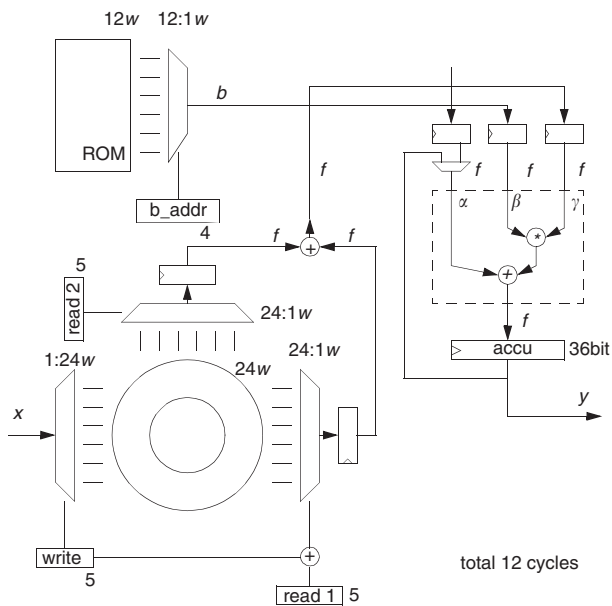


Fig. 5 FDF FIR filter block diagram

2.4 Design-for-testability

Different strategies could be used to achieve the required fault coverage for a given design. In this work, the required fault coverage ($>90\%$) of an FIR filter core dictated by the overall hearing aid DSP is achieved by using scan path for controller related cells and BIST for memory and MAC units. The BIST scheme employed in this work encloses the device under test (DUT) like a gauge and isolates it from the environment for the oncoming test, as shown in Fig. 6. A pattern generator applies specific test patterns to the DUT (e.g. MAC, RAM) in order to get maximum fault coverage with minimum test time. An output compressor

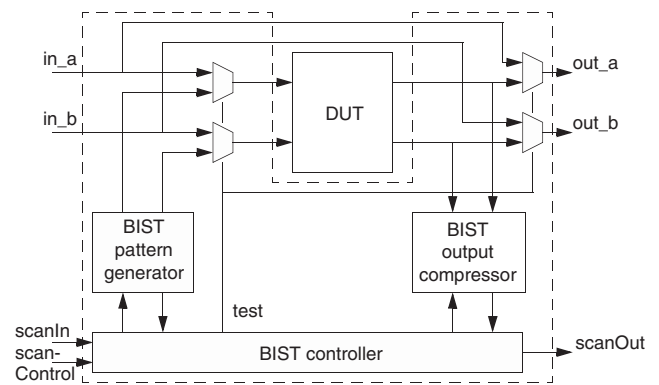


Fig. 6 Basic built-in self-test block diagram

takes the response from the DUT and minimises the amount of data to be analysed. Both the BIST pattern generator and the BIST output compressor are controlled by the BIST controller. Multiplexer circuits are used to isolate DUT while BIST is running. As an example, the BIST block diagram for the MAC is shown in Fig. 7. An effective BIST algorithm for fast multiplier cores has been presented in [14]. We have modified this scheme for a MAC circuit and achieved a fault coverage of $>97\%$ with only 1024 test patterns. A wide variety of memory test algorithms exist such as chess pattern, butterfly, MATS, MATS+, March C and 6-n algorithm [15]. In this work, we have implemented the 6-n algorithm, because of its small area consumption, short test time and excellent fault coverage compared to others. This has been demonstrated by previous research work in the literature [15]. These features are ideal for a device such as a hearing aid with strict performance constraints.

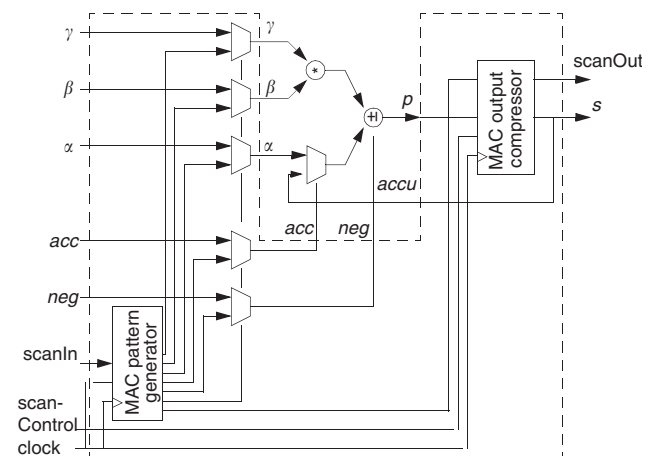


Fig. 7 Built-in self-test block diagram for MAC

3 Circuit synthesis and power analysis

A number of FIR filtering cores, each realising a 24-tap linear-phase low-pass filter, were developed. These filter cores vary in their realisation architecture, the type of multiplier circuits employed, and the number representation used, as shown in Table 1. Typical filter data (distorted sine wave) were used for the verification of the different filter cores. The data was generated by adding two sine wave signals, one representing the carrier and the other the distortion, with the following characteristics: $f_{\text{carrier}} = f_{\text{sample}}/9$, $f_{\text{distortion}} = f_{\text{sample}}/3$, and signal to noise ratio = 1.

The different FIR cores have been analysed with regard to fault coverage, area usage, and power consumption. The

Table 1: Designed FIR filters

Filter	Filter architecture	Multiplier type	Multiplier size	Partial product reduction tree	Number representation for multiplication
fir_df_booth	DF	Booth	16 × 16	3to2	2'sC
fir_df_rb_booth	DF	RB_Booth	16 × 16	4to2	2'sC
fir_df_rg_booth	DF	RG_Booth	16 × 16	3to2	2'sC
fir_df_wd	DF	Wallace-Dadda	16 × 16	3to2	2'sC
fir_df_booth_sm	DF	Booth	16 × 16	3to2	SM
fir_df_rb_booth_sm	DF	RB_Booth	16 × 16	4to2	SM
fir_df_rg_booth_sm	DF	RG_Booth	16 × 16	3to2	SM
fir_df_wd_sm	DF	Wallace-Dadda	16 × 16	3to2	SM
fir_fdf_booth	FDF	Booth	17 × 16	3to2	2'sC
fir_fdf_rb_booth	FDF	RB_Booth	17 × 16	4to2	2'sC
fir_fdf_rg_booth	FDF	RG_Booth	17 × 16	3to2	2'sC
fir_fdf_p_booth	FDF	P_Booth	17 × 16	3to2 & 4to2	2'sC
fir_fdf_wd	FDF	Wallace-Dadda	17 × 16	3to2	2'sC
fir_fdf_booth_sm	FDF	Booth	17 × 16	3to2	SM
fir_fdf_rb_booth_sm	FDF	RB_Booth	17 × 16	4to2	SM
fir_fdf_rg_booth_sm	FDF	RG_Booth	17x16	3to2	SM
fir_fdf_wd_sm	FDF	Wallace-Dadda	17x16	3to2	SM

cores were designed using verilog HDL and then synthesised using Ambit BuildGates™ targeting a 0.35μ standard cell CMOS library. The requirements for the synthesis were identical for all the cores. This was necessary to allow for a consistent power consumption and area usage comparisons. A maximum circuit delay of 35 ns has been defined for all the cores. A layout for each core was generated using the Envisia™ Silicon Ensemble™ place-and-route software. This was followed by extracting RC information and then performing RC back-annotated post-layout gate-level netlist simulations using Verilog-XL™ simulator. The resulting data including switching activity of the circuit nets and the capacitive load information extracted from the layouts was then used by the Synopsys DesignPower™ tool to compute power consumption figures for the different FIR cores. In all of the above stages a clock rate of 10 MHz and a supply voltage of 3 V were used. The results obtained are illustrated in Tables 2–4. The following can be concluded by analysing the tables:

- *Area consumption:* Area usage for different filter cores are illustrated in Tables 2 and 3. Clearly, *fir_df_wd* occupies the least area, followed by *fir_df_booth*. However, the variation in area among the different cores is less than 10%. Table 4 provides comparisons between different filter implementa-

tions using 2'sC and SM representations. An area increase of up to 3% is incurred when SM is used instead of 2'sC. On the other hand, a FDF implementation of a FIR filter consumes 4–9% more area compared to a DF implementation. This is mainly due to an area increase in the RAM (a dual port RAM is used instead of a single port RAM) and the additional adder circuitry used to add the two data samples obtained from the dual port RAM. The reduction in the number of multiplications (due to FDF) does not have much effect on the overall filter area, since both DF and FDF filters are single multiplier implementations.

- *Fault coverage:* Fault coverage of all the filters is above 90% (between 91 and 96%). This proves that the DfT features, which have been built into these filter cores, are very efficient. Only 1024 test vectors were needed to reach these results. It can be concluded that a fault coverage of 97 to 99% can be achieved using additional test vectors, especially since most of the undetected faults have been reported outside the DUT in the BIST circuitry.

- *Power consumption:* The results for DF and FDF filter architectures are illustrated in Tables 2 and 3. Comparing different DF filter core implementations using 2'sC number representation, *fir_df_rg_booth* achieves the best result with an overall reduction of 13% compared to *fir_df_booth*

Table 2: DF filter architecture analysis results

Filter	Fault coverage, %	Area, mm ²	Reduction, %	Power, nW/sample	Reduction, %
fir_df_booth	96	0.66	—	21.9	—
fir_df_rb_booth	96	0.68	–3	20.2	8
fir_df_rg_booth	96	0.71	–8	19.1	13
fir_df_wd	96	0.65	2	19.4	11
fir_df_booth_sm	94	0.67	—	22.4	—
fir_df_rb_booth_sm	94	0.70	–4	21.7	3
fir_df_rg_booth_sm	94	0.72	–7	21.0	6
fir_df_wd_sm	93	0.66	1	16.4	27

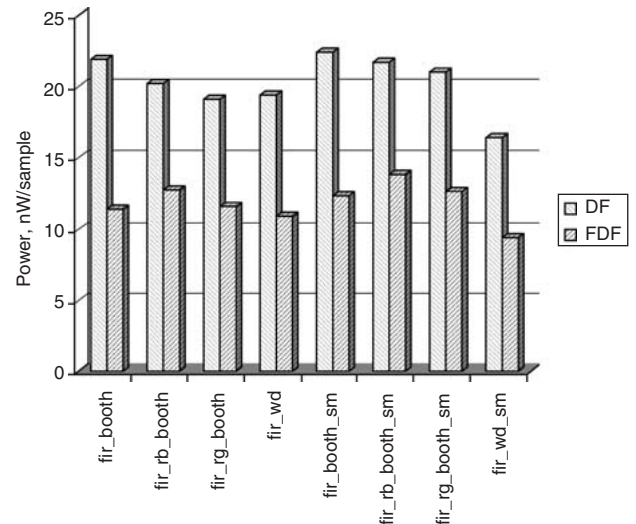
Table 3: FDF filter architecture analysis results

Filter	Fault coverage, %	Area, mm ²	Reduction, %	Power, nW/sample	Reduction, %
fir_fdf_booth	93	0.71	—	11.4	—
fir_fdf_rb_booth	94	0.71	0	12.7	-11
fir_fdf_rg_booth	95	0.76	-7	11.6	-2
fir_fdf_p_booth	94	0.74	-4	14.5	-27
fir_fdf_wd	95	0.71	0	10.9	4
fir_fdf_booth_sm	91	0.73	—	12.3	—
fir_fdf_rb_booth_sm	92	0.73	0	13.8	-12
fir_fdf_rg_booth_sm	94	0.78	-7	12.6	-2
fir_fdf_wd_sm	93	0.71	3	9.4	24

Table 4: Comparison between 2'sC and SM implementations

Filter	Area, mm ²			Power, nW/sample		
	2'sC	SM	%	2'sC	SM	%
fir_df_booth	0.66	0.67	-2	21.9	22.4	-2
fir_df_rb_booth	0.68	0.70	-3	20.2	21.7	-7
fir_df_rg_booth	0.71	0.72	-1	19.1	21.0	-10
fir_df_wd	0.65	0.66	-2	19.4	16.4	15
fir_fdf_booth	0.71	0.73	-3	11.4	12.3	-8
fir_fdf_rb_booth	0.71	0.73	-3	12.7	13.8	-9
fir_fdf_rg_booth	0.76	0.78	-3	11.6	12.6	-9
fir_fdf_p_booth	0.74	—	—	14.5	—	—
fir_fdf_wd	0.71	0.71	0	10.9	9.4	14

implementation. This is followed by *fir_df_wd* and *fir_df_rb_booth* achieving reductions of 11% and 8% respectively. However, when SM representation is used, *fir_df_wd_sm* provides the best result with a power reduction of 27%, followed by *fir_df_rg_booth_sm* (6%) and *fir_df_rb_booth_sm* (3%), compared to *fir_df_booth_sm*. On the other hand, when filter cores using SM representation are compared to their counterparts using 2'sC representation, the performance deteriorates for filter cores employing a Booth based multiplier. Although SM representation reduces the switching activity at data and coefficient inputs of the multiplier by 10% and 27% respectively, the overall power consumption increases by 2%, 7%, and 10% for *fir_df_booth_sm*, *fir_df_rb_booth_sm*, and *fir_df_rg_booth_sm* filter cores respectively, see Table 4. This is in sharp contrast to cases where a Wallace–Dadda multiplier is employed in the cores, resulting in a 15% improvement in the overall power reduction. The difference in this power profile can be explained by examination of the power reductions in the multiplier section of the filter cores. For example, in the case of *fir_df_wd*, a power reduction of 52% is achieved in the multiplier compared to only 11% reduction for *fir_df_rg_booth*. Therefore, the power reduction in the multiplier achieved with SM representation is not sufficient to compensate for the added overheads for Booth-based filter cores. When FDF filter cores are considered, *fir_fdf_p_booth* results in 27% more power consumption compared to *fir_fdf_booth*, see Table 3. This is mainly due to an increase in the number of glitches in the encoder section of the *P_Booth* multiplier. Similar to DF filter cores, the best result for FDF is obtained using *fir_fdf_wd_sm*, achieving a 24% reduction. Figure 8 shows that power

**Fig. 8** Power consumption comparison between DF and FDF implementations

savings between 36% and 48% can be achieved using FDF instead of DF architecture with an area increase of less than 10%. In general, our results indicate that SM representation outperforms 2'sC when used in Wallace–Dadda based filter core implementations. However, a Wallace–Dadda multiplier is slower compared to Booth-based multipliers. Therefore, if the required speed cannot be met by the Wallace–Dadda multiplier then either a faster multiplier (such as a Booth-based multiplier) or some speed-up techniques (such as pipelining and/or use of multiple multipliers; note that in this work these cases were not considered since they increase area usage significantly) has to be considered. If a Booth-based multiplier is chosen in a filter core our results indicate that 2'sC representation will lead to less overall power consumption compared to SM representation.

To analyse the performance due to the constituent components of the filtering cores examples of DF and FDF filter cores were considered using *fir_df_rg_booth* and *fir_fdf_rg_booth* as examples. Considering the power and the area performance of *fir_df_rg_booth* filter core, 62% of the area and 44% of the power in the MAC is used by the multiplier, as shown in Fig. 9. Although the carry-look-ahead (*cla*) adder consumes only 11% of the area, it is responsible for 28% of the MAC power. The *cla* is therefore another critical building block in addition to the multiplier. Therefore, a pipeline stage before the *cla* could be used to reduce its power consumption. The BIST in the MAC unit

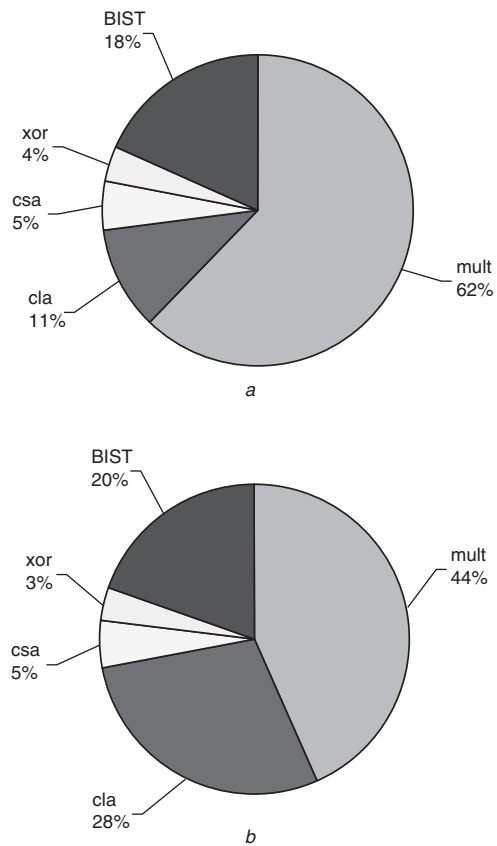


Fig. 9 MAC
a Area usage
b Power usage

consumes 18% of the area and 20% of the power. When the whole filter core is examined the MAC unit consumes 49% of the area and 81% of the power, as shown in Fig. 10. Although the RAM occupies 36% of the filter area, it consumes only 5% of the total power. This proves that the MAC is the most critical circuit and that the latch-based memory used in this work is very efficient in terms of power consumption.

Similarly, our results show that for the FDF filter core (*fir_fdf_rg_booth*), the multiplier consumes 75% of the MAC area and 55% of the MAC power. Approximately 30% of the MAC power is consumed in the *cla* circuit. A small increase in power consumption could be measured in the multiplier of the FDF compared to the DF filter core. This is due to an increase in the switching activity and the wordlength of the data, due to the addition stage before the multiplier. The BIST circuitry in the MAC consumes less than 20% of the total MAC area and power. The MAC unit consumes approximately 50% of the area and 70% of the power of the whole filter core. Although the RAM occupies 40% of the filter area, it consumes less than 10% of the total power.

4 Conclusions

The impact of different filter realisation structures, multiplier architectures and number representations on the overall power and area performance of a number of FIR filter cores has been studied within the context of a hearing aid application. The study includes the effect of DfT circuits on the overall performance, using typical compact BIST circuitry used in hearing aids applications. In general, power savings of up to 48% can be achieved using FDF filter cores

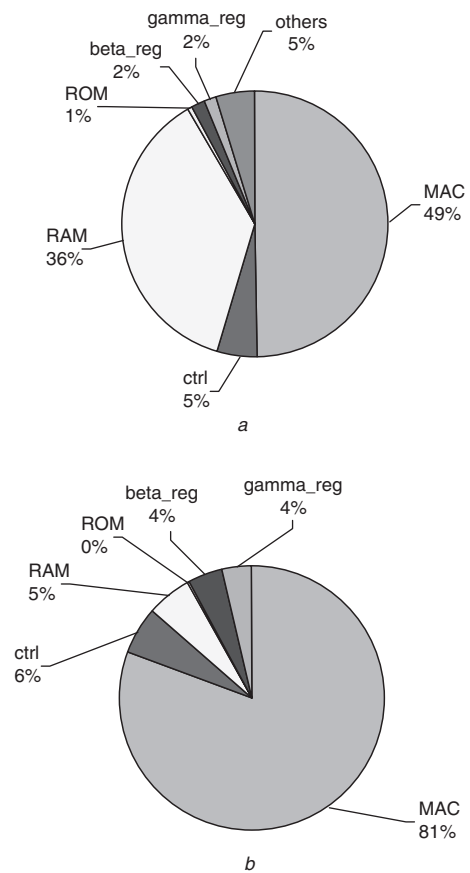


Fig. 10 FIR
a Area usage
b Power usage

over the DF ones at the expense of a less than 10% increase in area. The best power performance was obtained using a filter core with a Wallace–Dadda multiplier employing a signed-magnitude number representation. However, for high-speed applications where a booth multiplier will be required the best power performance can be obtained using a core with booth multiplier and two’s-complement data representation.

5 Acknowledgment

The authors would like to thank Bernafon Ltd., Switzerland, and the U.K. Engineering and Physical Sciences Research Council (grant no: GR/N08322) for their support.

6 References

- Lapsley, P.D., Bier, J.C., Shoham, A., and Lee, E.A.: ‘DSP processor fundamentals: architectures and features’ (IEEE Press, New York, 1997)
- Meier, P., Rutenbar, R.A., and Carley, L.R.: ‘Exploring multiplier architecture and layout for low power’. Proc. IEEE Custom Integrated Circuits Conf. CICC’96, pp. 513–516
- Keane, G., Spanier, J., and Woods, R.: ‘The impact of data characteristics and hardware topology on hardware selection for low power DSP’. Proc. IEEE Symp. on Low power electronics and design, ISLPED’98, Monterey, CA, USA, pp. 94–96
- Fried, R.: ‘Minimizing energy dissipation in high-speed multipliers’. Proc. IEEE Symp. on Low power electronics and design, ISLPED’97, pp. 214–219
- Farag, E., Yan, R.-H., and Elmasry, M.I.: ‘Power-efficient multiplier-accumulator design for FIR filters’. Proc. IEEE Canadian Conf. on Electrical and computer engineering, CCECE’97, pp. 27–30
- Moller, F., Bisgaard, N., and Melanson, J.: ‘Algorithm and architecture of a 1 V low power hearing aid instrument DSP’. Proc. IEEE Symp. on Low power electronics and design, ISLPED’99, San Diego, CA, USA, pp. 7–11

- 7 Nielsen, L.S., and Sparsø, J.: 'Designing asynchronous circuits for low power: An IFIR filter bank for a digital hearing aid', *Proc. IEEE*, 1999, **87**, (2), pp. 268–281
- 8 Bartlett, V.A., and Grass, E.: 'A low-power concurrent multiplier-accumulator using conditional evaluation'. Proc. 6th IEEE Int. Conf. on Electronics, circuits and systems, Pafos, Cyprus, 1999, pp. 629–633
- 9 Dadda, L.: 'Some schemes for parallel multipliers', *Alta Freq.*, 1965, **34**, pp. 349–356
- 10 Wallace, C.S.: 'A suggestion for a fast multiplier', *IEEE Trans. Comput.*, 1964, **EC-13**, pp. 14–17
- 11 Booth, A.D.: 'A signed binary multiplication technique', *Q. J. Mech. Appl. Math.*, 1951, **4**, pp. 236–240
- 12 Huang, X., Liu, W.-J., and Wei, B.W.Y.: 'A high-performance CMOS redundant binary multiplication-and-accumulation (MAC) unit', *IEEE Trans. Circuits Syst.*, 1994, **41**, (1), pp. 33–39
- 13 Chandrakasan, A.P., and Brodersen, R.W.: 'Minimizing power consumption in digital CMOS circuits', *Proc. IEEE*, 1995, **83**, pp. 498–523
- 14 Paschalis, A., Gizopoulos, D., Kranitis, N., Psarakis, M., and Zorian, Y.: 'An effective BIST architecture for fast multiplier cores'. Proc. IEEE Conf. Design, Automation and Test in Europe, DATE'99, pp. 117–121
- 15 Van De Goor, A.J.: 'Testing semiconductor memory' (John Wiley & Sons, 1991)