components ultimately limited f_s. The number of instructions required to sample, calculate the four coefficients and write to the DACs was 58, which gives f_s = 690kHz. The processor has two instructions which are of particular benefit to this application; these are *RECIPS* (reciprocal) and *RSQRT* (reciprocal square root). These generate approximate seeds which can be used by Newton- Raphson algorithms to generate a numerical solution. The architecture of the processor also allows multiple instructions to be executed in a single machine cycle, hence reducing execution time. Signal conversion is performed by two Linear Technology 14bit ADCs (LTC1419SWC) and two Burr-Brown 12bit dual DACs (DAC7802KP). The ADC input and DAC output ranges were chosen to be ±1.5V and ±10V, respectively. The sampling frequency dictated that an eighth-order Butterworth lowpass active filter was necessary to reconstruct the sampled output data; close tolerance, well-matched components were used in their construction. A Butterworth filter was chosen as a compromise between sharp cutoff and linear phase response.

The analogue and RF blocks are summarised as follows:

(i) *VCOs*: Z-Communications D930ME01 units were used; a divide-by-four prescaler lowers the RF output frequency to the 230MHz range. This also reduces the VCO tuning sensitivity from 8MHz/V to 2MHz/V, giving a loop gain element of 2×10^6. Buffering was added to prevent loading effects such as frequency pulling. The output power of the VCO blocks was –6dBm. The control signals were DC-coupled to the modulation ports; the VCO centre frequency, f_{vco} was set by summing a DC-level, using an opamp.

(ii) *Power amplifiers*: Mitsubishi FM power modules (M68707L) having a gain of ~30dB were used. Maximum output power is 7W at 45% efficiency.

(iii) *Power combiner*: In the prototype, a Wilkinson 3 dB combiner was used. This method of power combining is an inefficient process and limits the overall power efficiency of the CALLUM system. It should be possible to use a summing-switching amplifier to overcome this problem.

(iv) *Multiplier*: Four Analog Devices (AD-835) wideband multipliers were used to perform the multiplications and additions shown in eqn. 1.

(v) *Downconverter*: The LO is split into quadrature components by a Mini-Circuits PSCQ-2-250; the mixers are Mini-Circuits TUF-1s. The baseband I and Q channels are amplified using wideband opamps and provide both DC-offset and gain adjustments.

(vi) *Error amplifiers*: Analog Devices differential amplifiers (AD-830) were used.

A directional coupler is used to tap-off a fraction of the RF output power to the down-converter. The feedback path has suitable attenuation to achieve unity gain. The harmonic content of the fed-back signal is high due to the nonlinear amplifiers; a harmonic filter was used to reduce their level.

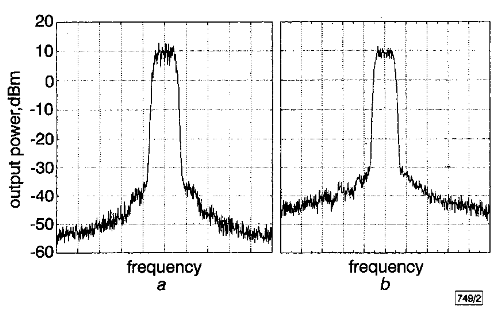

Fig. 2 *Output spectra of hardware CALLUM1 transmitter with π/4-DQPSK modulation* (α = *0.35, k = 500,000, R = 0.75*)

a 5kbaud; centre: 233MHz, 5kHz/div

b 18 kbaud (TETRA specification); centre: 233MHz, 20kHz/div

Results: The system was first set up to minimise demodulator gain and phase errors. The frequency of the LO defines the carrier frequency; the centre frequencies of the two VCOs were adjusted so that they are approximately the same as that of the LO. The magnitude of each VCO vector was set so that R = 0.75. Fig. 2*a* shows the synthesised output spectrum for a π/4-DQPSK baseband input signal with a data rate of 5kbaud and filter roll-off factor, a = 0.35. Fig. 2*b* shows the output with the same modulating signal, but with an increased data rate of 18kbaud. This is the specification for the emerging TETRA (terrestrial trunked radio) *standard.* The magnitude of the input vector r varies between ~0.15 and 1.0V. With R = 0.75, the maximum allowable value of r is 1.5V. In both cases, loop gain is set to approximately 5×10^5. This is the limiting stable gain, since a doubling of loop gain causes oscillation.

Discussion: For the first time, results are presented from a hardware implementation of a CALLUM1 system. In response to the 5kbaud π/4-DQPSK modulating signal, the adjacent channel power (ACP) is in the range –50 to –55dBc, whereas the 18kbaud signal's response is slightly worse at approximately –45 to –50dBc. This is to be expected because for a particular loop gain, tracking performance decreases with increasing frequency. If more loop gain is used then tracking performance will increase, but loop dynamics must be modified in order to prevent instability. Careful loop design is required to prevent low frequency poles from reducing stability margins. The overall power efficiency is less than the theoretical 100% due to the power combining method and the particular power modules used in the prototype. Results show that CALLUM1 is capable of excellent linearity with narrowband modulating signals and shows particular promise for use with future TETRA technology.

21 July 1998

Electronics Letters Online No: 19981315

D.J. Jennings and J.P. McGeehan (*Centre for Communications Research, University of Bristol, University Walk, Bristol BS8 1TR, United Kingdom*)

E-mail: David.Jennings@bristol.ac.uk

References

1 BATEMAN, A.: 'The combined analogue locked loop universal modulator (CALLUM)'. 42nd IEEE Veh. Technol. Conf., May 1992, pp. 759–763

2 COX, D.C.: 'Linear amplification with nonlinear components', *IEEE Trans.*, 1974, **COM-22**, pp. 1942–1945

3 CHAN, K.Y., and BATEMAN, A.: 'Linear modulators based on RF synthesis: Realization and analysis', *IEEE Trans.*, 1995, **CAS-42**, (6), pp. 321–333

Low-power coefficient segmentation algorithm for FIR filter implementation

A.T. Erdogan and T. Arslan

A new multiplication algorithm is introduced for the low-power implementation of digital filters on CMOS based digital signal processing systems. The algorithm decomposes individual coefficients into two less complex subcomponents. The decomposition, performed using a heuristic approach, divides a given coefficient such that a part is produced which can be implemented using a single shift operation, leaving another part with a reduced wordlength to be applied to the coefficient input of the hardware multiplier. This results in a significant reduction in the amount of switched capacitance and consequently power consumption. The authors describe the algorithm and present associated results, including the effects of overheads due to shift operations, showing up to 63% saving in power.

Introduction: Owing to the surge in the portable computing industry, there is a continuous demand for effective methods of implementing commonly used computationally intensive digital signal processing (DSP) tasks such as digital filtering. One way of reducing the power consumption of digital filters is to reduce the amount of switched capacitance during their operation. Examples of approaches in the literature for reducing the switched capacitance of digital filters are coefficient ordering [1], choice of appropriate coding techniques [2, 3], and the application of high level transformations [4, 5]. This Letter presents a new multiplication algorithm for the low-power implementation of digital filters on CMOS based DSP systems. The algorithm reduces the switched capacitance by decomposing individual coefficients into two less complex subcomponents. The decomposition, performed using a heuristic approach, separates a given coefficient such that a part is produced which can be implemented using a single shift operation, leaving another part with reduced wordlength to be applied to the inputs of the hardware multiplier. This results in a significant reduction in the amount of switched capacitance and consequently power consumption.

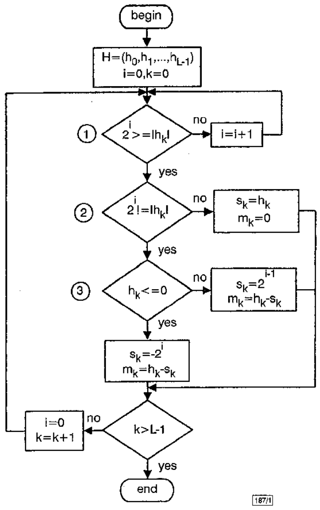

Fig. 1 *Flowchart of algorithm*

The two's complement representation is most commonly used in DSP applications. This is due to the ease of performing arithmetic operations such as additions and subtractions. However, a major drawback of the two's complement representation is the sign extension, which causes the MSB sign-bits to switch when signals change from positive to negative or vice-versa [2], hence increasing its power overhead. For this reason, the proposed algorithm is especially tailored for filters using two's complement multipliers, although the algorithm can be generalised to filters with multipliers using other data representations such as sign magnitude etc.

Implementation: The flow chart in Fig. 1 illustrates the main stages of the algorithm (indicated in circles). Given the coefficient set $H = (h_0, h_1, ..., h_{L-1})$, where L is the filter order, the algorithm proceeds through the coefficients sequentially. For a given coefficient h_k, the algorithm targets dividing it such that $h_k = s_k + m_k$, where s_k is the component to be implemented using a shift operation and m_k is the number to be applied to the hardware multiplier. To reduce the switched capacitance of the hardware multiplier, consecutive values of m_k applied to the multiplier input must be of the same polarity, to minimise switching, and have the smallest value possible, to minimise the effective wordlength. In our case m_k is chosen to be the smallest positive number. This criterion can be met by careful selection of s_k and consequently m_k. This selection procedure is the pivot of the stages shown in Fig. 1. For a small positive m_k, s_k must be the largest power of two number closest to h_k. For this reason, stage 1 is an iterative procedure which aims to find the largest power of two number greater than or equal to $|h_k|$.

Stage 2 deals with coefficients which are already power of two numbers, in which case the complete coefficient is realised using a single shift operation (i.e. $s_k = h_k$ and $m_k = 0$). In stage 3, the polarity of h_k is monitored. If h_k is a positive number then s_k is chosen as the largest power of two number smaller than h_k (i.e. $s_k = 2^{i-1}$). On the other hand, if h_k is negative, s_k is chosen to be the smallest power of two number larger than $|h_k|$ (i.e. $s_k = -2^i$). In both cases m_k is $h_k - s_k$. As an example, consider the filter $H = (-5, 8, -23, 75, 112, -37, -7, 65, -46, 49)$. Applying the algorithm will yield the s_k values in $S = (-8, 8, -32, 64, 64, -64, -8, 64, -64, 32)$ and the m_k values in $M = (3, 0, 9, 11, 48, 27, 1, 1, 18, 17)$.

To show the reduction in switching activity at the coefficient inputs of the multiplier, h_k and m_k values are shown in 8bit two's complement binary format in Table 1. Clearly, the total switching activity *TS* is reduced from 42 to 24 (by 42.87%). In addition, the effective coefficient wordlength is reduced to six bits since the most significant two bits of M are always zero. Given the data samples $X = (21, -58, -10, 111, -22, -6, -60, 122, 78, 77, ...)$ then the shift operations and the hardware multiplications produce the following output sets: $Y_s = (-168, 632, -1056, 2232, -984, ...)$, and $Y_m = (63, -174, 159, 42, 214, ...)$, respectively. The final filter output is obtained by summing the two sets, i.e. $Y = Y_s + Y_m = (-105, 458, -897, 2274, -770, ...)$.

Table 1: Total switching activity of H and M coefficient sets

h_0	11111011	m_0	00000011
h_1	00001000	m_1	00000000
h_2	11101001	m_2	00001001
h_3	01001011	m_3	00001011
h_4	01110000	m_4	00110000
h_5	11011011	m_5	00011011
h_6	11111001	m_6	00000001
h_7	01000001	m_7	00000001
h_8	11010010	m_8	00010010
h_9	00110001	m_9	00010001
TS	42	TS	24

Simulations and results: 8×8, 16×16, and 24×24bit two's complement array multipliers were implemented using Cadence VLSI suite with ES2 0.7μm CMOS technology. Coefficient sets were obtained by designing 10 practical FIR filters, five lowpass and five bandpass, for different filter specifications with filter lengths varying from 32 to 89. The coefficient sets were quantised to 8, 16, and 24bits. This was followed by the generation of zero mean uniformly distributed data samples for each filter. Next the coefficient sets were processed by the algorithm, which produces s_k and m_k values for each coefficient. This was followed by the automatic generation of input simulation files, in which the generated input data samples are associated with the corresponding m_k values for a given filter, for the Verilog-XL simulator [6]. Verilog-XL uses a hardware description language form of the multiplier circuit for the simulation procedure. For each simulation, the number of signal transitions for each gate was monitored. Capacitive information (wiring and loading capacitances) was extracted by performing a layout of the multiplier circuit. Both capacitive information and the switching activity figures were used to obtain the switched capacitance of each gate. This was then accumulated to give an overall figure for the switched capacitance of the multiplier.

Table 2: Average power reductions

Multiplier size	Algorithm	Without overheads		With overheads	
		SC	Reduction	SC	Reduction
		pF	%	pF	%
8×8	Conventional	14.88		14.88	
	New	5.23	64.88	5.57	62.56
16×16	Conventional	113.00		113.00	
	New	50.09	55.67	51.52	54.41
24×24	Conventional	413.81		413.81	
	New	256.32	38.06	260.08	37.15

The results obtained with the algorithm are summarised in Table 2 with different multiplier sizes. For each case, our results are compared to those obtained for conventional filtering, where data and coefficient values were directly applied to the multiplier. In each case, the results illustrate the amount of switched capacitance per multiplication, *SC*, and the percentage reduction in power when the new algorithm is compared with the conventional approach. Clearly, power saving is achieved in all cases with a maximum of 64.88% with an 8×8bit multiplier.

To estimate the overheads caused by shift operations, high-level models of the multiplier and the shifter circuits were used [5]. Analysis based on these models revealed that an 8bit shifter consumes 97.67% less capacitance than an 8×8bit multiplier. Similarly, 16 and 24bit shifters consume 98.73 and 99.09% less capacitance than 16×16 and 24×24bit multipliers, respectively.

These were used with our simulation results for estimating overheads. For instance, our simulations with an 8 × 8bit multiplier revealed that, using the conventional and the new multiplication algorithms, the average switched capacitance per multiplication is 14.88 and 5.23pF, respectively. Hence, the switched capacitance of the shifter was calculated to be 14.88 – 14.88 × 97.67/100 = 0.34pF. This is added to 5.23 to account for the shifter overhead. For this reason the net reduction for the above multiplier with our algorithm is 62.56%. The above procedure was repeated for both 16 × 16 and 24 × 24bit multipliers. The results show that overheads are between 1 and 3%, as can be deduced from Table 2.

Conclusions: This Letter presents a power saving algorithm for the implementation of digital filters on CMOS DSP systems. The algorithm aims to reduce the amount of switched capacitance within the multiplier section of the filter, this being the most computationally intensive part of the DSP, through fragmentation of the coefficients into two primitive parts, which can be processed with a significant reduction in the amount of switched capacitance. Results have been presented which indicate significant power savings. Although the algorithm has been demonstrated with FIR filter examples and two's complement representation, it can be generalised to most types of digital filter implementation using different data representations.

7 July 1998
Electronics Letters Online No: 19981290

A.T. Erdogan and T. Arslan (*Cardiff University of Wales, Cardiff School of Engineering, Cardiff CF2 3TF, United Kingdom*)

References

1 ERDOGAN, A.T., and ARSLAN, T.: 'Low power multiplication scheme for FIR filter implementation on single multiplier CMOS DSP processors', *Electron. Lett.*, 1996, **32**, (21), pp. 1959–1960
2 CHANDRAKASAN, A.P., and BRODERSEN, R.W.: 'Minimising power consumption in digital CMOS circuits', *Proc. IEEE*, 1995, **83**, (4), pp. 498–523
3 SU, C.-L., TSUI, C.-Y., and DESPAIN, A.M.: 'Low power architecture design and compilation techniques for high performance processors'. IEEE COMPCON 94, 1994, pp. 489–498
4 BRIGHT, M.S., and ARSLAN, T.: 'A genetic framework for the high-level optimisation of low power VLSI DSP systems', *Electron. Lett.*, 1996, **32**, (13), pp. 1150–1151
5 CHANDRAKASAN, A.P., POTKONJAK, M., MEHRA, R., RABAEY, J., and BRODERSEN, R.W.: 'Optimizing power using transformations', *IEEE Trans.*, 1995, **CAD-14**, (1), pp. 12–31
6 THOMAS, D.E., and MORBY, P.R.: 'The Verilog hardware description language' (Kluwer Academic Publishers, 1995)

Modified Monte Carlo scheme for high-efficiency simulation of the impulse response on diffuse IR wireless indoor channels

F.J. López-Hernández, R. Pérez-Jiménez and A. Santamaría

A modified Monte Carlo method to calculate multipath dispersion due to wall reflections on indoor wireless diffuse optical channels is presented. As with other Monte Carlo methods, it allows evaluation of not only the Lambertian but also specular reflections and can be used to validate previous simulation schemes. The main difference is that for each ray traced and for each rebound, a contribution to the receiver is calculated. This leads to a faster and more accurate simulation.

Introduction: A modified Monte Carlo algorithm for the calculation of the impulse response on infrared wireless indoor channels is presented. This work follows guidelines for studies of infrared (IR) wireless diffuse data communications systems. As is well-known, the characteristics of the room where the IR diffuse channel is implemented determine some communication problems, such as multipath penalty over the maximum baud rate or 'hidden station' situations. Classical algorithms [1, 2] require a high computational effort to calculate the impulse response in a regular sized room. The Monte Carlo method [3] makes it possible to validate the assumptions made for these classic algorithms (basically, the Lambertian nature of all reflections) with a computational complexity determined by the accuracy desired by the user. It is also a structure that can easily be assumed by a parallel computer architecture. Conversely, its main drawback is that, for a regular sized room, we need to send many more rays than the number of components that we will receive. This is because rays are not usually intercepted by the receiver. We have developed a mixed Monte Carlo deterministic algorithm which assures that each ray contributes to the final channel response function each time it rebounds against an obstacle. It dramatically increases the number of contributions and reduces the time required for an accurate simulation.

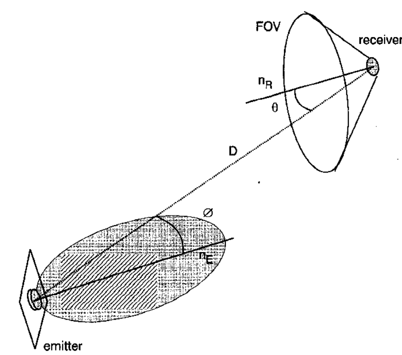

Fig. 1 *Angles used to calculate received power for LOS component*

Description of algorithm: The Monte Carlo diffuse IR channel response estimation procedure is split into three stages: ray generation, wall processing and calculation of the channel response. The walls (including the ceiling and floor) are defined by their reflection coefficient ρ at each point. We also define a diffuse-to-mirror ratio (DMR) depending on the incidence angle, which determines the amount of the incoming power that is specularly reflected or scattered. The DMR is a function of the roughness of the material [3]. The emitter is considered to follow a generalised-Lambertian profile [4], with a modal number m that defines the directivity of the emitted beam and is given by its position and its pointing direction. Monte Carlo modelling allows us to easily transfer this assumption to other radiation profiles, such as Gaussian or user-defined. The receiver is considered to be a semiconductor photodiode, defined also by its position and its pointing direction, and by two other characteristics: active area and the field of vision (FOV) [4]. It is also easily included that the effect of lenses in the emitter, receiver or both, is usually considered aspheric and is defined by its diameter (d) and focal distance (f).

The ray generation step depends on both the radiation profile and (for extended Lambertian sources) on the modal number m, as a measure of beam directivity. Rays are generated using the method proposed in [3]. In the wall processing stage, the first issue to be considered is the *impact* point of the ray, and whether it is on the receiver (line-of-sight (LOS) contribution) or in a wall (and in which wall). This is done through the use of a rotation matrix which relates the emitter position and the ray direction with the absolute coordinates of the room. Once the impact point is calculated, we can separate two components: the *LOS* (if it exists) and the *diffuse* components. The LOS component from the reflector to the receiver is calculated by

$$P_{LOS} = \frac{1}{\pi \cdot R^2} \cdot \cos(\theta) \cdot \cos(\varphi) \cdot A_r \cdot K(FOV) \qquad (1)$$

where we consider $m = 1$ (as it is a pure Lambertian scattering). A_r is the active area of the receiver and $K(FOV)$ is a retine function that indicates whether or not the incoming ray is seen by the receiver (depending on its FOV). R is the distance between the emitter and the receiver and will decide to which position of the received power vector this power should be added (as is detailed in the following). The angles indicate the ratio as a function of deviation from the centre of the radiation profile (Fig. 1). Note that, most of the time, all rays produce as many components as they suffer reflections. This makes this algorithm far more efficient