For comparison, results for the approach presented in [4] are shown in the third row of Table 1. In that approach, test points (t. p. in Table 1) are added to the data path and the state register has parallel loading (p. l. in Table 1) from primary inputs.

Conclusions: In this Letter, the significant influence of the controller on the testability of the whole circuit at RT level has been demonstrated. By simply changing the controller, test generation results can become standard. This influence has not previously been highlighted.

25 June 1997
Electronics Letters Online No: 19971114

V. Fernández and P. Sánchez (*Microelectronics Engineering Group, TEISA, University of Cantabria, Avda. de los Castros s/n, 39005 Santander, Spain*)

E-mail: victor@teisa.unican.es

References

1 PAPACHRISTOU, C., CHIU, S., and HARMANANI, H.: 'SYNTEST: A method for highlevel synthesis with self-testability'. Proc. ICCD, 1991, pp. 458–462
2 DEY, S., POTKONJAK, M., and ROY, R.K.: 'Exploiting hardware sharing in high-level synthesis for partial scan optimization'. Proc. ICCAD, 1993, pp. 20–25
3 FERNANDEZ, V., and SANCHEZ, P.: 'Partial scan high-level synthesis'. Proc. ED&TC, 1996, pp. 481–485
4 DEY, S., and POTKONJAK, M.: 'Non-scan design-for-testability of RT-level data paths'. Proc. ICCAD, 1994, pp. 640–645
5 NIERMANN, T.M., and PATEL, J.H.: 'HITEC: A test generation package for sequential circuits'. Proc. EDAC, 1991, pp. 214–218

Design of low-error fixed-width multiplier for DSP applications

Jer Min Jou and Shiann Rong Kuang

Indexing terms: Multiplying circuits, Digital signal processing

A low-error design of the fixed-width parallel multiplier for digital signal processing (DSP) applications is proposed. Applying two n bit inputs, it generates the n bit, instead of $2n$ bit, product with lower relative product errors, but uses only about half the area of a standard parallel multiplier. These features make it very suitable for use in many DSP applications such as arithmetic coding, wavelet transformation, digital filtering.

Introduction: Generally speaking, the multiplier is one of the core components in multimedia and DSP chips because it dominates the performance and area of these chips. To obtain a higher speed, parallel multipliers are always adopted at the expense of high area complexity. For many multimedia and DSP applications, the multiplication operations used in these applications usually have the special fixed-width property. That is, their input data and output results have the same bit width. For example, we have designed a data compression chip [1] which used an 8 bit fixed-width multiplier, and are now designing another wavelet chip for image compression which also requires a 12 bit fixed-width multiplier. In addition, many digital filters also use fixed-width multipliers [2]. Therefore, the aim of this Letter is to design a multiplier with lower product errors, without sacrificing performance, to make full use of the fixed-width property to save chip area.

Design of fixed-width multiplier: Considering the multiplication of two n bit inputs X and Y, the standard multiplier performs the following operations to obtain the $2n$ bit product P:

$$P = XY = \sum_{i=0}^{2n-1} P_i 2^i = \left(\sum_{i=0}^{n-1} x_i 2^i\right)\left(\sum_{i=0}^{n-1} y_i 2^i\right) \qquad (1)$$

where x_i, y_i and P_i denote the ith bit of X, Y and P, respectively. The product in eqn. 1 can be expressed by the sum of two segments: the most and the least significant segments MP and LP:

$$P = MP + LP = \sum_{i=n}^{2n-1} P_i 2^i + \sum_{i=0}^{n-1} P_i 2^i \qquad (2)$$

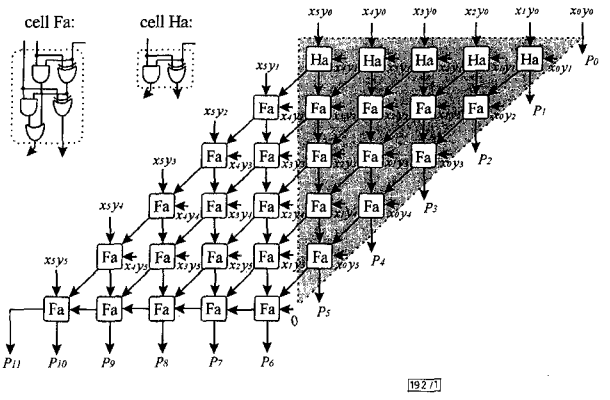

Fig. 1 *Block diagram of standard 6×6 parallel multiplier*

Fig. 1 shows the standard architecture of the parallel multiplier generating MP and LP for $n = 6$. The column (column circuit) generating P_i in the architecture is also denoted as P_i, and the circuit in the shaded region producing LP is also denoted as LP. The fixed-width multiplier described previously can be directly obtained by removing the shaded region and feeding each carry input of the residual part with a 0 to form a circuit denoted as MP', which is different from the circuit used to generate MP. This will save half of the area of the circuit. However, significant errors will be introduced in the fixed-width product which is undesirable for many fixed-width multimedia and DSP applications. To solve the problem, we propose another way to design a fixed-width multiplier with lower errors. The basic concept of our method is to add a small carry-generating circuit Cg to feed each carry input of MP', to effectively reduce errors. However, we encounter the problem of how to design such a circuit. Note that the maximal and the most accurate Cg is the shaded region circuit shown in Fig. 1 for $n = 6$.

Table 1: Average value of $|\alpha-\alpha'|/\alpha$ for different bit-width n

n	4	8	12	16
Average $\lvert\alpha-\alpha'\rvert/\alpha$	0.03363	0.11391	0.14551	0.13784

We first analyse the source of errors generated by MP', and then an improved design of Cg can be derived. Let ε be the difference between the two products produced by the standard multiplier and circuit MP', caused by the carries generated from column P_{n-1} (i.e. the column generating P_5 in Fig. 1 for $n = 6$) in circuit LP. Let α denote the sum of carries generated from P_{n-1}; then $\varepsilon = \alpha * 2^n$. Since $0 \le \alpha \le n-1$, then $0 \le \varepsilon \le (n-1)*2^n$. In fact, the values of those carries generated from P_{n-1} are in turn determined by the sum of product terms $x_i y_j$ (where $i+j = n-1$) and the carries produced by P_{n-2}. Continuing this process, we find that the more $x_i y_j$s in LP that are 1, the larger α is. Moreover, if product terms $x_i y_j = 1$ (where $i+j = n-1$), then the probability of each product term being equal to 1 in LP with x_i or y_j as the input (i.e. terms such as $x_i y_k$ or $x_k y_j$) is not < 1/2. Based on the above discussion and experimental verification, (see Table 1), we obtain the following experienced approximate relationship:

$$\alpha \simeq \alpha' = \begin{cases} 0 & \text{if } \beta = 0 \\ \beta - 1 & \text{if } \beta > 0 \end{cases} \quad \text{where } \beta = \sum_{i+j=n-1} x_i y_j \qquad (3)$$

Table 1 lists the average value of $|\alpha-\alpha'|/\alpha$ for different bit-widths n. The results show that α' is indeed approximate to α. According to eqn. 3, circuit Cg is designed to experience the value of α'. It consists of $n-2$ AO cells and one 2-input AND gate; each AO cell contains one 2-input AND gate and one 2-input OR gate. Each input of Cg is fed with a proper product term $x_i y_j$ in P_{n-1}, i.e.

the product term $x_i y_j$ where $i+j = n-1$. An example of Cg for $n = 6$ is given in the shaded area of Fig. 2*a* and *b*. To feed the outputs of proposed circuit Cg to the carry inputs of circuit MP' properly, a new fixed-width multiplier with lower product errors is proposed. The circuit of Fig. 2*a* is an example where $n = 6$. The speed of the proposed fixed-width multiplier is almost the same as that of MP', but it is better than that of the standard parallel multiplier. The product error and area analysis of this multiplier are given in the following Section.

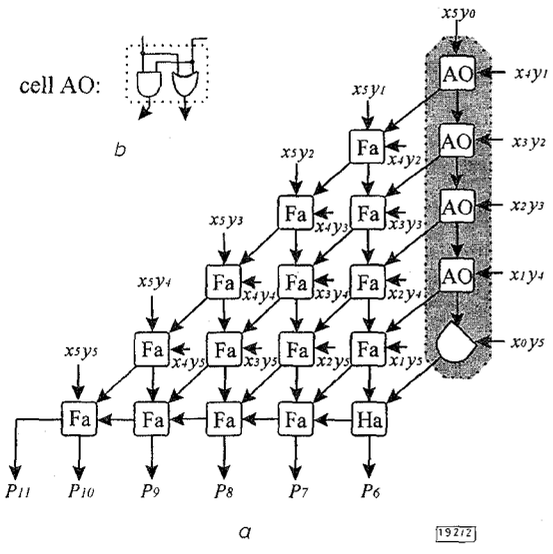

Fig. 2 *Block diagram of 6×6 proposed multiplier*

Error and area comparisons: Let fP be the product result of the proposed fixed-width multiplier, then its maximal absolute error ε_M is the maximum value of $|MP-fP|$s for all input pairs, and its relative error ε_R is $|MP-fP|/MP$. Given a value of w ($0 < w \leq 1$), we denote the percentage of ε_R which is larger than w as $\varepsilon_R P(w)$. It can be used as an accuracy measure of the fixed-width multiplier. For some w, the smaller $\varepsilon_R P(w)$ is, the more accurate the multiplier is. The experimental comparison results of ε_M and $\varepsilon_R P(w)$ for different fixed-width multipliers are listed in Table 2. Multiplier

Table 2: Comparison results of errors for different fixed-width multipliers

Multipliers	Errors	$n = 4$	$n = 8$	$n = 12$	$n = 16$
MP'	ε_M	48	1792	45056	983040
M_1		32	1536	40960	917540
M_2		16	768	20480	458752
Proposed		16	512	8192	196608
MP'	$\varepsilon_R P(0.01)$	30.9%	67.3%	17.4%	2.0%
M_1		73.8%	49.2%	9.9%	1.1%
M_2		9.4%	35.1%	8.5%	1.0%
Proposed		5.9%	20.7%	5.2%	0.6%
MP'	$\varepsilon_R P(0.0001)$	30.9%	75.8%	93.6%	81.8%
M_1		73.8%	60.9%	77.0%	64.8%
M_2		9.4%	47.2%	77.2%	52.9%
Proposed		5.9%	25.4%	37.9%	22.1%

M_1 in Table 2 is a fixed-width multiplier proposed in [3], where all its carry inputs are set to 0 except for the bottom one which is set to 1 to reduce the error. Multiplier M_2 in Table 2 is another fixed-width multiplier which consists of circuit MP' and column circuit P_{n-1} with each carry input fed with 0. The results show that our fixed-width multiplier is more accurate than any other. In addition, its area is also competitive. If we use CMOS technology and assume that each 2-input NAND or NOR gate needs four transistors, then the number of transistors required in an $n \times n$ standard multiplier and our proposed multiplier is $48n^2-66n+24$ and $24n^2-12n-42$, respectively. When n is large enough, the proposed multiplier consists of only half of the transistors of a standard multiplier. Conversely, let A_A, A_H, and A_F be the areas of an AND gate, a half adder, and a full adder, respectively, and assume that the multipliers are made up of these modules. Using the standard cells of 0.8μm SPDM technology [4], we have that $A_H \simeq 0.43A_F$ and $A_A \simeq 0.1A_F$. The comparison results of the ratio of areas between fixed-width and standard multipliers are listed in Table 3. The area of the proposed multiplier is nearly half of the area of a standard multiplier and is less than the area of M_2.

Table 3: Comparison results of area ratio for different fixed-width multipliers

Multipliers	$n = 8$	$n = 16$	$n = 32$
MP'	0.443	0.477	0.490
M_1	0.460	0.481	0.491
M_2	0.584	0.543	0.522
Proposed multiplier	0.539	0.522	0.512

Conclusion: The design of a low-error fixed-width multiplier has been proposed. It is useful in fixed-width data path architectures for multimedia and DSP applications in which a uniform word width is usually required. By using this type of multiplier, the chip area can be significantly reduced and no performance degradation is introduced.

24 June 1997
Electronics Letters Online No: 19971087

Jer Min Jou and Shiann Rong Kuang (*Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, Republic of China*)

References

1 JOU, J.M., KUANG, S.R., CHEN, Y.L., and CHIANG, C.Y.: 'An on line adaptive data compression chip using arithmetic codes'. ISCAS, 1996, Vol. IV, pp. 360–363
2 TERRELL, T.J.: 'Introduction to digital filters' (Halsted Press, John Wiley & Sons Inc., NY, 1988)
3 KIDAMBI, S.S., EL-GUIBALY, F., and ANTONIOU, A.: 'Area-efficient multipliers for digital signal processing applications', *IEEE Trans. Circuits Syst. II*, 1996, **43**, (2), pp. 90–94
4 0.8μm SPDM Technology Manual, Computer & Communication Laboratory, Industry Technology Research Institute, Republic of China

Discriminative temporal feature extraction for robust speech recognition

Jia-lin Shen

Indexing terms: Speech recognition, Feature extraction

A discriminative temporal feature processing method for robust speech recognition is presented by combining the knowledge and the statistical methods. The cepstral features are first filtered by a RASTA method based on human hearing perception and then processed using the minimum classification error algorithm. Improved recognition performance can be achieved in both quiet and noisy environments.

Introduction: Many speech recognition systems use cepstral coefficients as the feature parameters, which perform well in a quiet environment. However, the recognition performance degrades drastically in the presence of noise. To increase their robustness, the first time-derivative of the cepstral features can be approximated by a first-order orthogonal polynomial over a finite length window, which is denoted the well-known delta cepstral coefficients [1]. Alternatively, the RASTA (relative spectral) method was proposed by adding an extra pole to an FIR bandpass filter used in delta processing based on knowledge of human hearing perception, in which a further improvement in recognition performance under noisy environments can be achieved [2]. Although these temporal features (delta, RASTA features) provide better performance in noisy environments, the long-term processing window will