If $\varepsilon_r = 1$, the finline becomes a ridged waveguide with zero-thickness ridges.[4] The guided wavelength in this case is given by

$$\lambda_g/\lambda = [1 - (\lambda/\lambda_c)^2]^{-1/2} \quad (7)$$

where λ_c can be found in Reference 5 for several values of w/b. Eqn. 7 was used to calculate λ_g/λ and these values were compared with the results by our method as shown in Table 1. It

Table 1 λ_g/λ OF SYMMETRICAL RIDGED WAVEGUIDES

w/b	0·1		0·4		1·0	
f	Eqn. 7	Our method	Eqn. 7	Our method	Exact values	Our method
GHz						
30	1·1255	1·1262	1·2407	1·2504	1·4053	1·4055
35	1·0887	1·0882	1·1604	1·1673	1·2526	1·2525
40	1·0651	1·0655	1·1160	1·1207	1·1766	1·1767
45	1·0504	1·0507	1·0883	1·0917	1·1318	1·1317

$h_1 + 2d = h_2 = a/2$; $s_1 + s_2 = b$; $2d = 0{\cdot}254$ mm; $\varepsilon_r = 1$
waveguide = WR(28)

can be seen that two results agree to within 0·07 percent when $w/b = 0{\cdot}1$, and 0·7 percent when $w/b = 0{\cdot}4$. When $w/b = 1{\cdot}0$, $\lambda_c = 2a$, the results obtained by eqn. 7 are exact. In this case, our results are different from the exact values by only 0·02 percent. It turns out that the higher the frequency, the more accurate the results obtained by our method are.

If $w/b = 1$ and $\varepsilon_r > 1$, the finline becomes a slab-loaded waveguide which can be analysed by the transverse resonance method (TRM).[8] Fig. 2 shows the variation of λ_g/λ with the

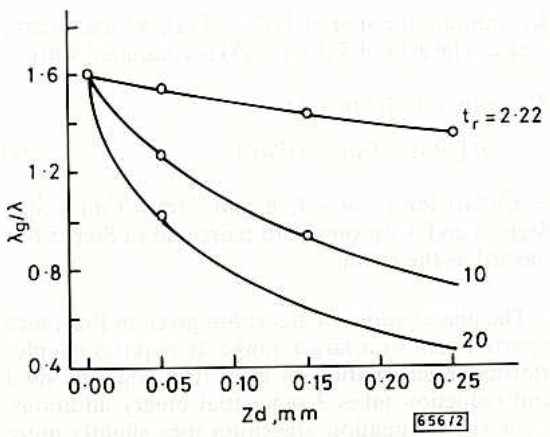

Fig. 2 *Variation of λ_g/λ with dielectric thickness*
$s_1 = 0$, $s_2 = b$, $h_1 = h_2 = (a - 2d)/2$, $f = 40$ GHz, waveguide = WR(19), ——— our method, ○○○ TRM

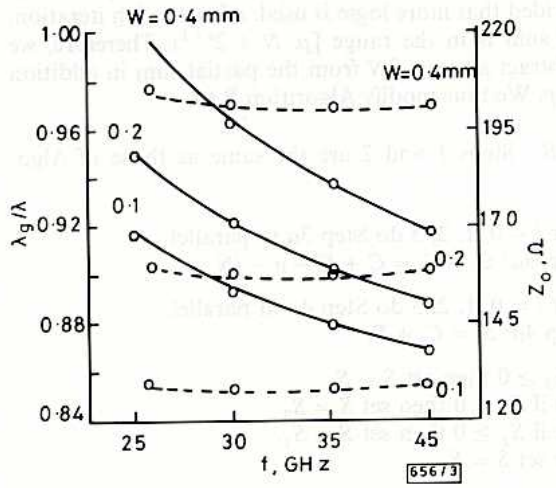

Fig. 3 *λ_g/λ and impedance against frequency*
$s_1 = s_2 = b$, $h_1 = 3{\cdot}302$ mm, $h_2 = 3{\cdot}556$ mm, $2d = 0{\cdot}254$ mm, $\varepsilon_r = 2{\cdot}22$, waveguide = WR(28), ○○○ SDM, ——— λ_g/λ, – – – – Z_0 our method

dielectric thickness ($2d$) for several values of ε_r. One can see that the agreement between the two results is within 0·4 percent.

Fig. 3 shows the plots of λ_g/λ and the impedance against frequency in several values of w by our method and the spectral method (SDM). The agreement with the results taken from Reference 7 is within 0·3 percent when $w = 0{\cdot}4$ mm and 0·2 percent when $w \leq 0{\cdot}2$ mm for a guided wavelength. It can also be seen that the two results agree very closely for the impedance.

Three computer programs were developed. The average time (CPU) for calculating a root is about 1·5 s by our method ($N = 3$), 3 s by the SIE technique ($N = 3$)^2 and 25 s by the SDM (2 basis functions, 70-term Fourier expansions) on a VAX11-780.

In conclusion, our method is clear in physical concepts, accurate in results and efficient in numerical computation.

Y. HUANG
L. XUE
W. LIN

Box 85B1
University of Electronic Science & Technology of China
Chengdu 610054, Sichuan, Peoples Republic of China

24th January 1990

References

1 MITTRA, R., and ITOH, T.: 'A new technique for the analysis of the dispersion characteristics of microstrip lines', *IEEE Trans.*, 1971, **MTT-19**, pp. 47–56
2 OMAR, A. S., and SCHÜNEMANN, K.: 'Formulation of the singular integral equation technique for planar transmission lines', *ibid.*, 1985, **MTT-33**, pp. 1313–1322
3 MITTRA, R., and LEE, S. W.: 'Analytical techniques in the theory of guided waves' (Macmillan, New York, 1971)
4 HOPFER, S.: 'The design of ridged waveguides', *IRE Trans.*, 1955, **MTT-3**, pp. 20–29
5 MARCUVITZ, N. (Ed.): 'Waveguide handbook' (Peregrinus, London, 1986), pp. 400–401
6 COLLIN, R. E.: 'Field theory of guided waves' (McGraw-Hill, New York, 1960)
7 BHAT, B., and KOUL, S. K.: 'Analysis, design and application of finline' (Artech House, Norwood, 1987), pp. 189–191

MULTI-OPERAND MODULO ADDITION USING CARRY SAVE ADDERS

Indexing terms: Logic and logic design, Adders, Algorithms

We compare two approaches for computing $S = \sum_{i=1}^{k} X_i \pmod N$ using carry save adders. The biased addition technique proposed in Reference 7 keeps the partial carry and sum vectors $C + S$ biased by $p = 2^n - N$ where $n = \log_2 N$. The sign estimation technique proposed in this paper computes the sign of $C + S - N$ using a 2-bit carry look-ahead logic. The sign estimation algorithm restricts the partial sum to a smaller range, and thus, performs the final reduction faster than the biased addition algorithm.

Introduction: The multi-operand modulo addition, i.e. the computation of

$$\sum_{i=1}^{k} X_i \pmod N \quad (1)$$

where $X_i \in [0, N)$ for $1 \leq i \leq k$, can be performed in two different ways. First, we can sum all X_is and then perform a division operation to find the remainder. This technique could utilise the algorithms and fast digital structures developed for multi-operand binary addition.[5,2,3] However, it is slower owing to the final division operation of an $(n + \log k)$-bit number by an n-bit number.

The second approach reduces the partial sum after addition of each operand. This is similar to the modular multiplication algorithm which reduces the partial product after each shift and add operation.[1,4]

The following code describes the multi-operand modulo N addition, with reduction at each step.

Algorithm A:

Step 1: Set $S = X_1$

Step 2: Repeat Step 2*a* and 2*b* for $i = 2, 3, \ldots, k$
Step 2*a*: $S = S + X_i$
Step 2*b*: If $S \geq N$ then $S = S - N$

Step 3: Stop

Biased addition technique: To avoid the carry propagation delay with an ordinary binary adder, we can use a carry save adder to implement Steps 2*a* and 2*b*.[3,6] This approach makes the comparison operation in Step 2*b* difficult, since an exact comparison cannot be made without the full precision operation $S + C - N$. Assuming the number of bits of the modulus N is n, this operation takes at least O(log n) time.

The use of a biased partial sum for the computation of eqn. 1 was proposed in Reference 7. This approach uses unsigned binary representation: n bits for the sum and $n + 2$ bits for the carry. The partial sum $C + S$ is kept biased by $p = 2^n - N$, and once $C \geq 2^n$, N or $2N$ is subtracted from $C + S$. After Step 2 is finished, the bias p and up to 3 times the modulus N are subtracted from the partial sum to obtain the final result S in the range $[0, N)$.

Algorithm B:

Step 1: Set $C = p$ and $S = X_1$

Step 2: Repeat Step 2*a* and 2*b* for $i = 2, 3, \ldots, k$
Step 2*a*: If $C \geq 2^{n+1}$ then $C + S = C + S - 2^{n+1} + 2p$
else if $C \geq 2^n$ then $C + S = C + S - 2^n + p$
Step 2*b*: $C + S = C + S + X_i$

Step 3: If $T(C) + T(S) \geq 2^{n+1}$ then $S = C + S - 2^{n+1} + 2p$
else if $T(C) + T(S) \geq 2^n$ then $S = C + S - 2^n + p$
else $S = C + S$

Step 4: If $S \geq 2^n$ then $S^* = S - 2^n + p$
else $S^* = S$

Step 5: If $S^* \geq 2^n$ then $S = S^* - 2^n$.
else $S = S^* - p$

Step 6: Stop

The carry save addition in Step 2 uses an n-bit unsigned sum and an $(n + 2)$-bit unsigned carry, hence it actually involves a 2-bit carry propagate addition besides an n-bit carry save addition. In Step 2 the partial sum $C + S$ remains in the range $[p, N + 2^{n+1})$, and in Steps 3, 4, and 5 the final sum $C + S$ is reduced and the bias is removed. The function $T(x)$ in Step 3 is computed as

$$T(x) = 2^t(x \text{ div } 2^t) \tag{2}$$

i.e. it fills the binary number x with t zeros from bit 0 to bit $t - 1$. Thus we have

$$T(x) \leq x < T(x) + 2^t \tag{3}$$

In Reference 7, t is chosen to be $n - 1$. Thus the operation $T(C) + T(S)$ requires the addition of a 3-bit number and a 1-bit number. This is achieved by using a carry propagate adder.

Sign estimation technique: We propose the use of two's complement carry save addition and sign estimation to perform the multi-operand modulo addition. In Step 2*b* of Algorithm A, we need to perform a comparison and a subtraction. This can be achieved by first performing the subtraction and then either keeping the difference if it is positive or restoring the partial sum otherwise. Adding X_i and subtracting N is performed in one clock cycle with a carry save adder, but in order to compute the exact sign of the partial sum $C + S$, we have to sum C and S in full precision, which takes at least O(log n) time. Our algorithm estimates the sign to a certain precision in the process of summing X_is and postpones the full precision computation of $C + S$ until the final reduction stage.

Algorithm C:

Step 1: Set $C = 0$ and $S = X_1$

Step 2: Repeat Step 2*a* and 2*b* for $i = 2, 3, \ldots, k$
Step 2*a*: $C + S = C + S + X_i$
Step 2*b*: $\hat{C} + \hat{S} = C + S - N$
Step 2*c*: If $T(\hat{C}) + T(\hat{S}) \geq 0$ then set $C = \hat{C}$ and $S = \hat{S}$

Step 3: $S = C + S$

Step 4: $\hat{S} = \hat{C} + \hat{S}$

Step 5: If $\hat{S} \geq 0$ then set $S = \hat{S}$

Step 6: Stop

In Step 2*b*, N is subtracted from the partial sum $C + S$ to form $\hat{C} + \hat{S}$. If the sign of $\hat{C} + \hat{S}$ is estimated to be positive, the partial sum is changed to the new value, resulting in a reduction of N; otherwise, the partial sum remains unchanged. We can prove that after each iteration of Step 2*c*, $C + S$ and $\hat{C} + \hat{S}$ are in the range $[0, N + 2^t)$ and $[-N, 2^t)$, respectively. Consequently, if $t \leq n - 1$ then either $C + S$ or $\hat{C} + \hat{S}$ is in the range $[0, N)$ after the kth iteration. Choosing $t = n - 1$, we see that in Step 2, $C + S$ and $\hat{C} + \hat{S}$ are in the ranges $[0, 2N + 2^{n-1})$ and $[-N, N + 2^{n-1})$, respectively. We thus allocate $n + 3$ bits for C and S, and $n + 2$ bits for $\hat{C}$ and $\hat{S}$.

In order to compute the sign of $T(C) + T(S)$, we use a carry look-ahead logic. The sign of $T(\hat{C}) + T(\hat{S})$ is computed with

$$SIGN = \hat{C}(n+1) \oplus \hat{S}(n+1) \oplus [G(n) + G(n-1)P(n)] \tag{4}$$

where $G(i) = \hat{C}(i)\hat{S}(i)$ for $i = n - 1$, n and $P(n) = \hat{C}(n) + \hat{S}(n)$. Finally, in Steps 3 and 4 we sum both pairs and in Step 5 one of them is selected as the result.

Conclusions: The biased addition algorithm given in Reference 7 keeps the partial sum in a larger range. It requires simpler logic to perform the summation of each item, but the final correction and reduction takes 3 sequential binary additions. In contrast, our sign estimation algorithm uses slightly more logic in summing each item, but requires only 2 binary additions in the final stage, which can be performed in parallel.

Steps 3 to 5 in Algorithm B, i.e., the final correction and reduction of biased addition algorithm, can be improved in speed, provided that more logic is used. After the kth iteration, the partial sum is in the range $[p, N + 2^{n+1})$. Therefore, we need to subtract at most $3N$ from the partial sum in addition to the bias p. We thus modify Algorithm B to

Algorithm B': Steps 1 and 2 are the same as those of Algorithm B.

Step 3: For $i = 0, 1, 2, 3$ do Step 3*a* in parallel
Step 3*a*: $C_i + S_i = C + S - p - iN$

Step 4: For $i = 0, 1, 2, 3$ do Step 4*a* in parallel
Step 4*a*: $S_i = C_i + S_i$

Step 5: If $S_3 \geq 0$ then set $S = S_3$
else if $S_2 \geq 0$ then set $S = S_2$
else if $S_1 \geq 0$ then set $S = S_1$.
else set $S = S_0$.

Step 6: Stop

We note that the above modification is by no means the only possible one. There are several designs using fewer carry

save adders and requiring more clock cycles during the final reduction stage. In Table 1 we compare Algorithms B, B' and C implemented with a one-level or multilevel carry save adder. In contrast to Reference 7, the analyses presented in Table 1

Table 1 CHARACTERISTICS OF ALGORITHMS

		B	B'	C
Summation of items	Length of CSA	n	n	$n+3$
	Length of CPA	2	2	0
	Length of CLA	0	0	2
	Number of cycles	$2k-2$	$2k-2$	$2k-2$
Final reduction	Length of CSA	$n+2$	$n+2$	$n+2$
	Length of CLA	3	0	0
	Level of CSA	1	4	1 or 2
	Number of cycles	$3n+4$	$n+2$	$2n+4$ or $n+2$

do not use carry look-ahead adders or other fast structures for the summation of the final carry and sum vectors. Extension of the analysis to this direction is straightforward and thus omitted.

Ç. K. KOÇ
C. Y. HUNG

Department of Electrical Engineering
University of Houston
Houston, TX 77204, USA

30th January 1990

References

1 BLAKLEY, G. R.: 'A computer algorithm for the product AB modulo M', *IEEE Trans. Comput.*, 1983, **32**, (5), pp. 497–500
2 HO, I. T., and CHEN, T. C.: 'Multiple addition by residue threshold functions and their representation by array logic', *IEEE Trans. Comput.*, 1973, **22**, (8), pp. 762–767
3 HWANG, K.: 'Computer arithmetic, principles, architecture, and design' (John Wiley and Sons, Inc., 1979)
4 SLOAN, K. R., JUN.: 'A computer algorithm for the product AB modulo M', *IEEE Trans. Comput.*, 1985, **34**, (3), pp. 290–292
5 SINGH, S., and WAXMAN, R.: 'Multiple operand addition and multiplication', *IEEE Trans. Comput.*, 1973, **22**, (2), pp. 113–119
6 WASER, S., and FLYNN, M. J.: 'Introduction to arithmetic for digital system designers' (Holt, Rinehart and Winston, CBS College Publishing, 1982)
7 ZHANG, C. N., SHIRAZI, B., and YUN, D. Y. Y.: 'Parallel designs for chinese remainder conversion'. Proc. Int. Conf. on Parallel Processing, 17th–21st August 1987, pp. 557–559

CORRELATION OF INTERFERENCE AND BIT ERROR ACTIVITY IN A DIGITAL TRANSMISSION SYSTEM

Indexing terms: Data transmission, Digital transmission errors, Burst errors

By injecting electrical interference into a digital transmission system an analysis technique has been demonstrated which allows the nature of the interference to be identified from the correlation of the resulting decision errors.

Introduction: A first attempt has been made to establish a means by which the error activity in a digital transmission system may be used as a diagnostic tool rather than, as in current practice, just a performance parameter. Such a tool would allow ailing systems to be identified and removed from service prior to their performance becoming unacceptable, and provide an indicator to the root cause which would also expedite the repair of systems and their early restoration.

It is postulated that the diagnostic information related to system failure is inherent within the error activity, and that the analysis of the interval between errors[1] may be sufficient to estimate the primary mechanism. Preliminary results from laboratory trials are presented to support this thesis.

The test bed: A system was configured to allow known interference waveforms to be injected into a nominally error free transmission system (Fig. 1). A commercial BER tester was

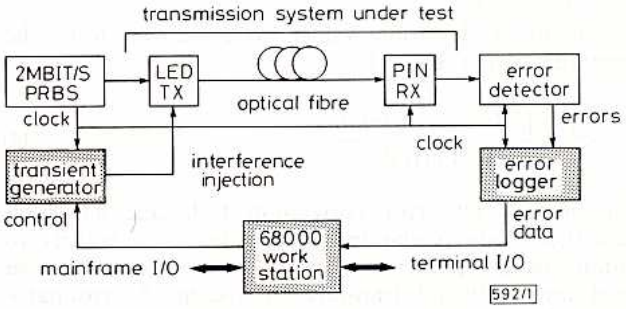

Fig. 1 *Burst error acquisition and analysis system*

used to supply data and detect errors whilst the interference waveform was produced from a pre-programmed table via a D/A convertor. The interference waveform was injected at the LED drive of the optical fibre transmission system forming the link under test. Both the generation of the interference waveform and the error interval acquisition were controlled by a workstation. The recorded error activity was uploaded to a host computer for offline analysis, which involved comparing the statistics of the autocorrelation functions (ACF) of the actual and predicted error activity:

$$\Gamma_{\varepsilon\varepsilon}(\tau) = \Sigma_i \varepsilon(t_i)\varepsilon(t_i + \tau) \quad (1)$$

Where $\varepsilon(t)$ = the error pattern generated by a given interferer; t_i = a bit epoch time; τ = a bit period; i = an integer.

Laboratory measurements and simulations: A series of error activity measurements and analyses were performed for two basic types of interference waveform. The first was a peak and decay waveform given by:

$$v(t) = At \exp(-t/T) \quad (2)$$

and the second was a decaying sine wave of the form:

$$v(t) = A \exp(-t/T) \sin(\omega t) \quad (3)$$

These were selected to be the most likely candidates for practical models of interference waveforms.[2] A computer simulation of the behaviour of the transmission system in the presence of interference was written and the predictions compared with measured results. To simplify the simulation and measurements procedure, the work reported here focussed on the behaviour of the decision circuit alone.

Typical results are given in Fig. 2 for the two cited classes of interferer. The amplitude and decay times of the interfering

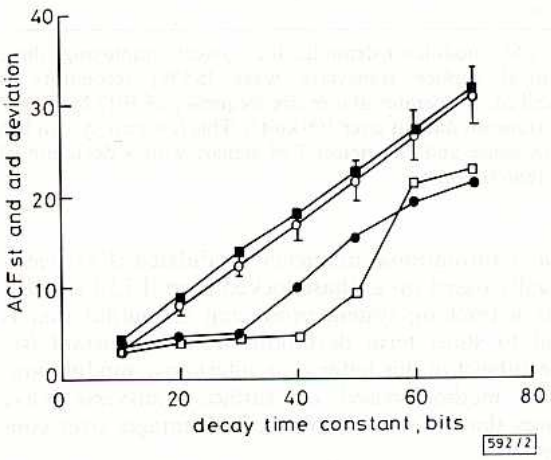

Fig. 2 *Standard deviations of waveforms*
■ measured decaying waveform
○ predicted decaying waveform
● measured sinusoidal decaying waveform
□ predicted sinusoidal decaying waveform