According to Haug [4], B for a quantum-well laser may be expressed as $B = B_1 T^{-1} d$ where T is the absolute temperature, d is the width of the well and B_1 is a function of momentum matrix element for the transition, the photon energy and the heavy hole, light hole and electron masses but is relatively temperature insensitive. The Auger coefficient C may be expressed as $C = a \exp(-\Delta E/kT)$ where a is a constant, k is the Boltzmann constant and ΔE is the activation energy for the Auger process. Thus a plot of ln $(C/B^{3/2}T^{3/2})$ against $1/kT$ should yield a straight line of gradient ΔE. This approach has been adopted for the values of $C/B^{3/2}$ obtained from Fig. 1 and the results plotted in Fig. 2. Reasonable straight lines are obtained for both devices. The gradients correspond to activation energies of 248 and 160 meV for the strained and unstrained devices, respectively. Providing that the hole effective mass m_H remains appreciably larger than the electron effective mass m_e, we may expect $\Delta E \propto 1/m_H$ [1].

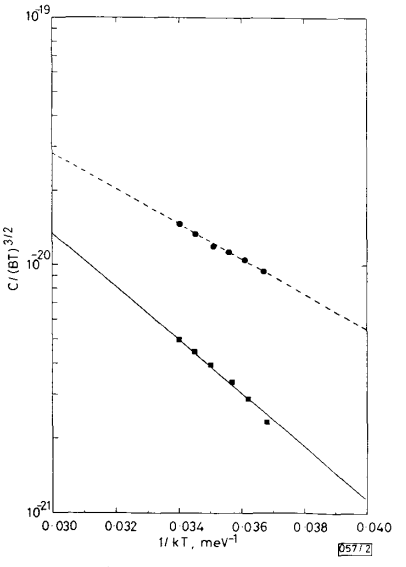

Fig. 2 *ln $C/(BT)^{3/2}$ against $1/kT$ using data from Fig. 1 for unstrained and strained lasers*

Gradient gives activation energy ΔE for Auger process
- - - - activation energy $\Delta E = 160$ meV
——— $\Delta E = 248$ meV
● unstrained
■ strained

Thus we may conclude that the effective value of m_H has been decreased to ~0·645 of its original value as we go from the unstrained to the strained quantum wells. It is now interesting to consider the ratio of $C/B^{3/2}$ for the two different devices. As seen in Fig. 2 there is a decrease of $C/B^{3/2}$ of approximately 4·4 at room temperature. Considering first the Auger coefficient C, this should decrease by a factor 30 due to the change in activation energy. Again assuming that m_H is appreciably larger than m_e, it would be expected that $B_1 \propto 1/m_H$. Therefore $B \propto m_H^{-1} d$ should decrease by 1·93 going from the unstrained 75 Å well width device to the 25 Å well width strained device and hence $B^{3/2}$ should decrease by 2·69. This means $C/B^{3/2}$ should decrease by a factor 11. In view of the simplicity of the models used and the approximations made the difference between the factor 11 predicted and the factor 4·4 observed may not be significant but it is interesting to note that the decrease in $C/B^{3/2}$ is less than the change in activation energy would predict. Most importantly the results support the conclusion that the effect of compressive strain significantly reduces the relative rate of Auger recombination to radiative recombination in compressively strained layer lasers.

In conclusion we have observed that $C/B^{3/2}$ is significantly reduced as we go from unstrained to compressively strained MQW lasers. This change can be adequately explained in terms of a reduced hole effective mass. It is interesting to note that while the Auger coefficient is decreased by strain its rate of change with temperature is increased and this may be the cause of the high temperature sensitivity remaining in some strained devices. Clearly the values of C and B depend on the exact device structure. Indeed similar measurements on an unstrained device reported earlier [3] yielded a value of $\Delta E = 107$ meV and a value of $C/B^{3/2}$ nearly four times larger than for the unstrained device measured here. More complex calculations taking into account the nonparabolicity of the valence band and the penetration of the hole quasi-Fermi level into the band are clearly required so that the optimum ways to reduce C may be arrived at.

Acknowledgments: The authors are grateful to K. Poguntke and E. P. O'Reilly for very helpful discussions and to the SERC and DTI who supported this work under a LINK programme. Thanks also to C. Seltzer and D. Cooper (BTL) for supplying the lasers.

10th February 1992

A. R. Adams, J. Braithwaite and V. A. Wilkinson (*Physics Department, University of Surrey, Guildford GU2 5XH, United Kingdom*)

References

1 ADAMS, A. R.: 'Band-structure engineering for low threshold high efficiency semiconductor lasers', *Electron. Lett.*, 1986, **22**, pp. 249–250
2 THIJS, P. A. J., BINSMA, J. J. M., TIEMEIJER, L. F., and VAN DONGEN, T.: 'Improved performance 1·5 μm wavelength tensile and compressively strained InGaAs-InGaAsP quantum well lasers (invited), to be published in ECOC Proc., September 1991
3 POGUNTKE, K. R., and ADAMS, A. R.: 'Analysis of radiative efficiency of long wavelength semiconductor lasers', *Electron. Lett.*, 1992, **28**, (1), pp. 41–42
4 HAUG, A.: 'Relations between the T_0 values of bulk and quantum well GaAs', *Appl. Phys.* 1987, **B44**, pp. 151–153

ROM-BASED SPECIAL PURPOSE MULTIPLICATION AND ITS APPLICATIONS

H.-M. Jong, L.-G. Chen and T.-D. Chiueh

Indexing terms: Read-only memories, Digital signal processing, Digital circuits, Computer architecture

A special-purpose ROM-based component is proposed to calculate $x \sin(\theta)$ and $x \cos(\theta)$. The method significantly reduces the memory requirement of look-up tables. Applying the method to the design of a co-ordinate rotator, a low-cost pipelined architecture is proposed that provides a throughput of 0·25 rotations per addition time.

Introduction: Multiplication is an essential but costly computation in digital signal processing. There exist some special forms of multiplication that are useful in many applications. The calculations of $x \sin(\theta)$ and $x \cos(\theta)$ are two typical examples. For these special tasks, table look-up is a suitable alternative to general purpose hardware multipliers. Several methods have been proposed to reduce the size of the required table ROMs [1, 2]. Another approach to perform such computations is the well known CORDIC algorithm [3]. These previous methods are general purpose, and versatile in many types of computation. However, for the specified calculations $x \sin(\theta)$ and $x \cos(\theta)$, they seem redundant and costly because they fail to make use of the specific property of the required calculation. The purpose of this Letter is to present a method for these calculations. The proposed method significantly reduces the memory requirement and can be implemented for

low cost. It is then applied to the co-ordinate rotator design and results in a high throughput pipelined rotator.

Proposed method: To compute $x \cos(\theta)$, the following relation is used:

$$2 \sin(e) \cos(\theta) = \sin(e + \theta) + \sin(e - \theta)$$

Let $x = 2 \sin(e)$, i.e., $e = \sin^{-1}(x/2)$, then

$$x \cos(\theta) = \sin(e + \theta) + \sin(e - \theta)$$

The computation can be completed by three table look-ups and three additions. To reduce the memory requirement further, an approximation can be applied:

$$\sin^{-1}(x) \simeq x \qquad \text{for } |x| \ll 1$$

The input data x is at first divided by $N = 2^k$ to fit the condition $|x/N| \ll 1$. The process requires no physical computation for fixed point binary systems. Now let $x/(N/2) = 2 \sin(e)$, i.e. $e = \sin^{-1}(x/N)$, then

$$x \cos(\theta) = \frac{N}{2}[\sin(e + \theta) + \sin(e - \theta)]$$

$$\simeq \frac{N}{2}\left[\sin\left(\frac{x}{N} + \theta\right) + \sin\left(\frac{x}{N} - \theta\right)\right] \qquad (1)$$

It means that the computation of $x \cos(\theta)$ can be executed by the procedure shown in Fig. 1. The time required to complete an $x \cos(\theta)$ calculation comprises two additions and one table look-up. In pipelined cases, the proposed architecture can provide an $x \cos(\theta)$ per addition or table look-up time.

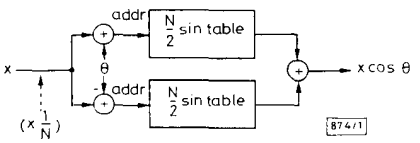

Fig. 1 *Computation of x cos θ using an approximation*

Similar techniques can be applied to the computation of $x \sin(\theta)$:

$$x \sin(\theta) \simeq \frac{N}{2}\left[\cos\left(\frac{x}{N} - \theta\right) - \cos\left(\frac{x}{N} + \theta\right)\right] \qquad (2)$$

An exact-bit emulator is built to evaluate the error results from adopted approximations. The simulation result is shown in Table 1. Comparing to conventional multiplication, the resulting errors of the proposed method are acceptable for many applications, especially when considering the significant speed up and reduction in cost. The appropriate choice value of N depends on the tradeoff between specified accuracy and ROM size.

Table 1 SIMULATION RESULTS

Method		$Error_{max}$	ROM size
		%	
8 bit × 8 bit		27·32	
8 bit × 9 bit		4·51	
Proposed	$N = 256$	8.49	2^{10} words
16 bit	$N = 512$	5.54	2^{11} words

To avoid the exponential increase of ROM size as the address is lengthened, $x \cos(\theta)$ can be computed by

$$\begin{aligned} x \cos(\theta) &\simeq \tfrac{1}{2}[\sin(x_H + x_L + \theta_H + \theta_L) \\ &\quad + \sin(x_H + x_L - \theta_H - \theta_L)] \\ &= \tfrac{1}{2}[\sin(x_H + \theta_H) \cos(x_L + \theta_L) \\ &\quad + \cos(x_H + \theta_H) \sin(x_L + \theta_L) \\ &\quad + \sin(x_H - \theta_H) \cos(x_L - \theta_L) \\ &\quad + \cos(x_H - \theta_H) \sin(x_L - \theta_L)] \\ &\simeq \tfrac{1}{2}[\sin(x_H + \theta_H) + \sin(x_L + \theta_L)] \\ &\quad + \tfrac{1}{2}(x_L + \theta_L) \cos(x_H + \theta_H) \\ &\quad + \tfrac{1}{2}(x_L - \theta_L) \cos(x_H - \theta_H) \end{aligned}$$

where x and θ are fixed point $2W$ bit numbers and $x = x_H + x_L$, $\theta = \theta_H + \theta_L$. Subscripts H and L represent the halves of more significant and less significant W bits, respectively. It shows that an $2W$ bit operation can be calculated via three W bit operations and the memory requirement is reduced by a factor of 2^{W-1}.

Application: a rotator: The method presented above has many direct applications, such as various sinusoidal transforms. A general application is co-ordinate rotation that is useful for signal processing and computer graphics. Applying eqn. 1 and eqn. 2 to equations of co-ordinate rotation, we obtain the following expressions:

$$x' \simeq \tfrac{1}{2}[\sin(x + \theta) + \sin(x - \theta) + \cos(y + \theta) - \cos(y - \theta)]$$

$$y' \simeq \tfrac{1}{2}[\sin(y + \theta) + \sin(y - \theta) - \cos(x + \theta) + \cos(x - \theta)]$$

where (x, y) and (x', y') are scaled Cartesian co-ordinates of a point before and after rotating co-ordinate axes counterclockwise by angle θ, respectively. The resulting architecture is shown in Fig. 2. The operation of this pipelined rotator can be described by the reservation table shown in Fig. 3. The computation latency of a rotation is six cycles and the throughput is 0·25 rotations per cycle, where the cycle time is the larger one of addition and table look-up delay. The required ROM size depends on the specified accuracy of application. The rotator can be applied to various computations, such as the butterfly computation for fast Fourier transform, to construct high-throughput low-cost computing modules.

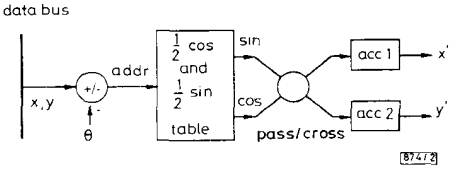

Fig. 2 *Proposed pipelined rotator*

Operator \ Time	1	2	3	4	5	6	7
$\pm\theta$	$x + \theta$	$x - \theta$	$y + \theta$	$y - \theta$			
ROM		$x + \theta$	$x - \theta$	$y + \theta$	$y - \theta$		
pass/cross			pass	pass	cross	cross	
acc 1			$\sin(x + \theta)$	$\sin(x - \theta)$	$\cos(y + \theta)$	$\cos(y - \theta)$	- - -
acc 2			$-\cos(x + \theta)$	$\cos(x - \theta)$	$\sin(y + \theta)$	$\sin(y - \theta)$	- - y' x'

Fig. 3 *Reservation table of pipelined rotator*

Conclusion: This Letter has described ROM-based algorithms and architectures for computing $x \sin(\theta)$ and $x \cos(\theta)$. Some properties of sine and cosine functions are used to reduce the required table size. The proposed architecture can be pipelined to provide an $x \sin(\theta)$ or $x \cos(\theta)$ per addition or table look-up time. Simulation results show that acceptable precision can be achieved in a feasible ROM size. An application of the proposed method is also presented. The proposed coordinate rotator is pipelined to provide a throughput of 0·25 rotations per addition or table look-up time.

20th January 1992

H.-M. Jong, L.-G. Chen and T.-D. Chiueh (*Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China*)

References

1 HWANG, K. (Ed.): 'Computer arithmetic' (John Wiley & Sons, New York, 1979), pp. 201–206
2 TAYLOR, E. J., GILL, R., JOSEPH, J., and RADKE, J.: 'A 20 bit logarithmic number system processor', *IEEE Trans.*, 1988, **C-37**, pp. 190–200
3 CONSIDINE, V.: 'CORDIC trigonometric function generator for DSP'. Proc. IEEE ICASSP, 1989, pp. 2381–2384

NEURAL NETWORK SYNTHESISER OF PAUSE DURATION FOR MANDARIN TEXT-TO-SPEECH

Shaw-Hwa Hwang and Sin-Horng Chen

Indexing terms: Speech synthesis; Signal processing, Neural networks

A neural network based approach of pause-duration synthesis for Mandarin text-to-speech is proposed. It uses an MLP to replace explicit synthesis rules for generating pause duration from input text. By properly training the MLP using a large set of utterances, phonological rules of producing pause duration are automatically learned. Experimental results confirmed that this is a promising approach.

Introduction: In Mandarin Chinese, each character is pronounced as a monosyllable. Pause duration between two successive monosyllables plays an important role in the naturalness of sentential speech; pause duration is thus important prosodic information in synthesising Mandarin text-to-speech. Traditionally, it is synthesised by a rule-based approach [1–2]. Phonological rules are invoked to imitate the human pronunciation process of generating pause duration from a given text. Although a rule-based approach is simple, the process of rule inference is tedious. Besides, because various linguistic features may interactively affect the pronunciation of pause duration, rules are usually incomplete.

In this Letter, a neural network based approach of pause-duration synthesis is proposed. The basic idea is similar to the NETalk used for assigning parameters of allophones to each English character according to the context [3]. In our method, a multilayer perceptron (MLP) is employed to replace explicit synthesis rules to generate pause duration according to the context. By properly training the MLP with a large training set using the error back-propagation algorithm, phonological rules are expected to be automatically deduced and implicitly memorised. The MLP can hence be taken as a mechanism of pause-duration synthesis. Intensive study of linguistics for rule inference is therefore unnecessary.

Proposed system: Fig. 1 shows the block diagram of the proposed system. It consists of two main parts. The first one is a text analysis in which some linguistic features representing the context are extracted. The second one is a single-output MLP serving as the mechanism of generating pause duration from these linguistic features. It is noted that the nonlinear operation in the output node of an MLP is removed for linearly generating multilevel values of pause duration.

Although linguistic features on various levels may affect the pronunciation of pause duration, in this study only some relevant linguistic features, listed below, are extracted from neighbouring context in the text analysis:

(1) The type of initial of the ensuing syllable: six broad types of initial listed below were used:

(a) /m, n, l, r, 'null'/

(b) /h, sh, shi/

(c) /b, d, g/

(*d*) /tz, j, ji/

(*e*) /p, t, k/

(*f*) /ts, ch, chi, f, s/

(2) The tone of the ensuing syllable: five lexical tones were used

(3) One positional parameter: is the ensuing syllable the ending syllable of a sentence?

(4) Two phrasal parameters:

(*a*) does the processing location precede a polysyllabic phrase?

(*b*) does the processing location lie within a polysyllabic phrase?

(5) Others: does there exist an intentional pause or breath?

From the above discussions, a total of 15 binary contextual features were used. We note that the last feature is used to compensate for the effect of unusual pauses that occasionally occurred in the training utterances.

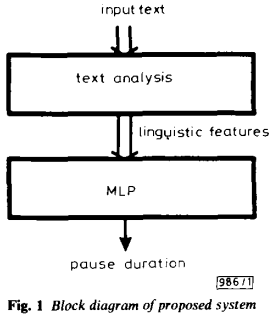

Fig. 1 *Block diagram of proposed system*

Simulations: The performance of the proposed approach was examined by simulations. Two sets of sentential utterances spoken by a female announcer were recorded from TV news. The first database comprising 2278 monosyllables was employed to train the MLP and the second database comprising 584 syllables was used for outside testing. All utterances are natural and fluent. They were manually segmented into syllable periods for extracting pause durations. Contextual features were also manually extracted.

Table 1 lists the average mismatch error between original and synthesised pause durations. Average mismatch errors of

Table 1 AVERAGE MISMATCH ERROR BETWEEN ORIGINAL AND SYNTHESISED PAUSE DURATIONS

	Mean	Absolute variation
Statistics of pause duration	7·33 frames	8·10 frames
	Average mismatch error	
Inside test	1·84 frames	
Outside test	1·88 frames	

1 frame = 3·75 ms.