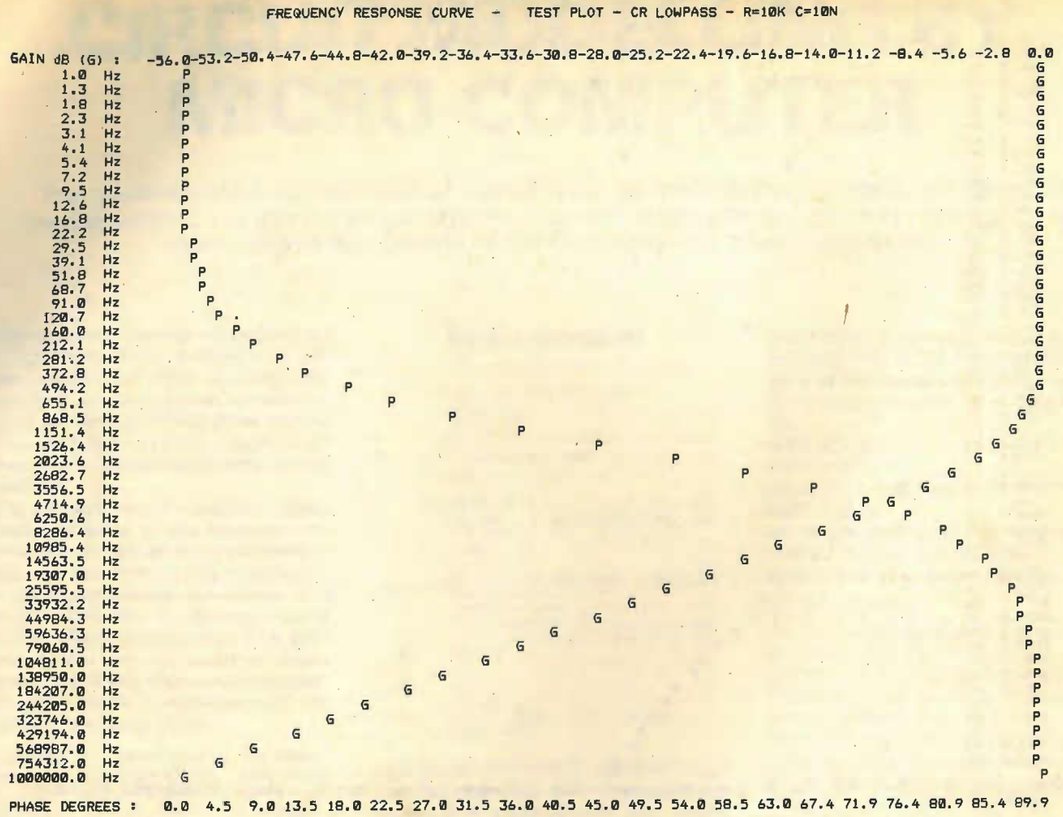

While working on loudspeaker crossovers, I wanted to model a crossover feeding two drive units, one of which was connected out of phase. For speakers in phase it was easy to find the sum by specifying a summing network, but for an out-of-phase speaker, I found it necessary to invent a non-existent circuit element. I have called this an 'inverter' which is a two terminal device with the property of 'losing' any current flowing into it while taking in an equal current at its other end which is also lost. This violates Kirchhoff's law and the charge conservation laws, but the method works on the computer. The 'inverter' has admittance determinant:

$$\begin{matrix} +1E5 & +1E5 \\ +1E5 & +1E5 \end{matrix}$$

and the value used (always a positive value added to the YR array) was +1E5, so that the net effect was of a small resistor connecting an out-of-phase speaker to the output node.

Having cut the time for our typical 16-node circuit frequency plot to 16 minutes from two hours, I then tried the effect of a BASIC compiler and used the Microsoft compiler. This produces true machine code and the time for 50 reductions from 16 nodes to a 2-by-2 was now 2 minutes 48 seconds. With the addition of the macro assembler codes to the two inner loops (P and Q), this was cut to 2 minutes 6 seconds, a saving over the original running time of 98%.

So it can be seen that with a little effort, much time can be saved. The purchase of a Basic compiler compatible with the interpreter can turn the home computer into a useful designer's tool.

Appendix

Macro-codes for fast reduction

The macro-code used was as follows: Each operation is shown with its equivalent in Basic:

```
MSB   Y3,Y5,Y4,Y6,B   ;B=Y3*Y5+Y4*Y6
MAD   Y5,Y4,Y6,Y3,C   ;C=Y4*Y5+Y6*Y3
MAD   B,Y1,C,Y2,D     ;D=B*Y1+C*Y2
MSB   C,Y1,B,Y2,E     ;E=C*Y1+B*Y2
DIV   D,A,D           ;D=D/A
DIV   E,A,E           ;E=E/A
SUB   YRP@,D,YRP@     ;YR(P,@)=YR(P,@)-D
SUB   YIP@,E,YIP@     ;YI(P,@)=YI(P,@)-E
RET
END
```

The macro definitions, which should precede their use, are:

```
MAD   MACRO   M1,M2,M3,M4,ANS
      MPY     M1,M2,T1
      MPY     M3,M4,T2
      ADD     T1,T2,ANS
      ENDM

MSB   MACRO   M1,M2,M3,M4,ANS
      MPY     M1,M2,T1
      MPY     M3,M4,T2
      SUB     T1,T2,ANS
      ENDM
```

All other macro definitions (ADD, SUB,MPY,DIV) are machine dependant, and are not shown here.

Note: A version of the circuit modelling program, called ACM, suitable for TRS80 micro-computers, will be available from Molimerx Ltd, 1 Buckhurst Road, Town Hall Square, Bexhill-on-Sea, E. Sussex.

References

This article is an extension of "Circuit analysis by small computer," by A. S. Beasley, *Wireless World*, Feb. and April 1980. Photocopies of this are available from WW, Editorial, at a price of 90p inclusive. An interesting discussion of the theory may be found in "Two-port representation of multi-mode networks by matrix partitioning," by R. T. Kennedy, J.I.E.R.E. Feb. 1969. □

Orchestral sounds, halls and timbre — a correction

Denis Vaughan has kindly pointed out to us one or two misprints which crept into his article in the May 1982 issue: Just under the heading 'First reflections' on p.32, the phrase should read: "Their timing is exactly controlled by the width (1 foot ≈ 1ms)." In the middle of page 33, reference is made to Guildford and this should read Gilford. In the third column of the same page, there are two references to reflection times which should read: "this means that the effectively larger reflections start about 81ms after the original sound'. and; "Kingsway has quite a lot of powerful reflections to offer within the first 105ms. Because the larger reflections continue to return up to 147ms, the substantial and lengthy support of the musicians is assured". The figures printed (18 and 14ms) could be misleading, especially to those interested in modelling electronically the initial reflection pattern of the hall.

DIGITAL DIVIDERS WITH SYMMETRICAL OUTPUTS

The author uses Johnson counters with controlled feedback to give symmetrical even and odd-numbered divisions of a clock pulse.

By Cornelius van Holten

Time and again, in literature on digital circuitry, ideas are published on the problem how to obtain a 50% duty cycle when a regular pulse train is divided by an odd number. Some clever (and less clever) methods are proposed, e.g. the use of exclusive-or gates in the clock pulse lines, a separate flip-flop with a delay of half a pulse period, the output of which is combined with the normal flip-flops, etc.

In my opinion, the use of EXOR-gates in clock lines should be avoided, since spikes on the output-signals of the flip-flops may occur; a better way is to combine the outputs signals of the flip-flops. The ideas, found in Refs. 3 and 4 are broadened in this paper, and a generalized scheme is proposed which may be easily expanded. Moreover, the control input is pure binary and there is no attempt to change the (odd or even) sequence length. Standard i.cs are used.

The Theory

When a Johnson or Möbius ring counter is fed back, a sequence length of n or 2n is derived, depending on whether a straight or twisted loop is used. The maximum sequence-length is 2n for n bits, and sequences of 2(n-1) etc, are derived when outputs, other than the last, are chosen. When two adjacent outputs are fed back via an AND-gate and negated, (Fig. 1.) any length between 2n and 2 may be obtained.

If an auxiliary flip-flop is connected to the chain and is switched on the opposite pulse edge, the output is shifted over ½T, where T is the clock pulse period. It is necessary for the incomong pulse train to have a duty cycle of 50%; if not, a divider is needed which will halve the frequency. In Fig. 2. the outputs of 2 flip-flops, FF_1, the last in the chain, and FF_2, the extra flip-flop, are combined in an OR-gate to obtain an odd sequence length (9) with a symmetrical output. In this case, $\bar{D}$ and $\bar{E}$ are fed back (see Table 1).

Table 1. Feedback signals and sequence length.

Feedback	Sequence/length
A	2
AB	3
B	4
BC	5
C	6
CD	7
D	8
DE	9
E	10
EF	11
F	12
FG	13
G	14
GH	15
H	16

When an even sequence length is chosen, a symmetrical output is derived from the last flip-flop in the chain, only one (negated) output is fed back and no OR process is needed. In the Table 2, a list is given of all possible combinations; I through VIII are the controls signals which switch the (negated) I for $\bar{A}$, II for $\bar{B}$, . . . VIII for $\bar{H}$.

Table 2. Control inputs and corresponding sequence lengths.

Control inputs I	II	III	IV	V	VI	VII	VIII	Output (*T)	
1								1 +1	= 2
1	1							1½+1½	= 3
	1							2 +2	= 4
	1	1						2½+2½	= 5
		1						3 +3	= 6
		1	1					3½+3½	= 7
			1					4 +4	= 8
			1	1				4½+4½	= 9
				1				5 +5	=10
				1	1			5½+5½	=11
					1			6 +6	=12
					1	1		6½+6½	=13
						1		7 +7	=14
						1	1	7½+7½	=15
							1	8 +8	=16

Complete circuit

In Fig. 3, the complete diagram is given, consisting of 8 flip-flops (a shift register), a pulse circuit, an output, feedback gates controlled by the inputs I to VIII, and a decision making circuit with 4 full adders for odd and even lengths.

The latter operates as an EXOR-gate with 8 inputs: Y = I ⊕ III ⊕ IV. . . . ⊕ VIII and therefore Y = '1' for odd and '0' for even lengths; the unused input of the full adder at the bottom is permanently held at a logical '1' level.

In the output circuit, the function H + YZ is realized. For Y = O, the output becomes H (for even length sequences) and for Y = 1, the output is H + Z (for an odd length) as shown in the time charts in Fig. 4a and 4b respectively.

The flip-flops A to H are D flip-flops, operating in the leading clock pulse edge and Z (auxiliary flip-flop) reacting on the trailing edge of it. The P flip-flop is needed when the input pulses are not symmetrical, and a buffer gate is used for amplification. The correction and enabling circuit is described in the Appendix. In normal circumstances, this circuit is inoperative and the shift register is loaded with all zeros by the enabling input, and cycles via 10000000, 11000000, 11100000, . . . through 11111111, 01111111, etc. back to the all zero condition. This is the "normal" sequence, 1 out of the 16 possible cycles. Of course, other values of n than 8 are possible, this number has been chosen for comparison with the circuit described by Girolami and Bamberger[2].

Modification

In Fig. 3(a), there are 8 control inputs which are used separately (for even lengths) or in groups of adjacent pairs (for odd lengths). If one wishes to control the sequence length via a binary weighted control input, a decoder is needed as described in Table 3.

In Fig. 3(b), a read only memory is programmed as a decoder, and the input 1 may be used to control the output circuit: even or odd; the output function is H +

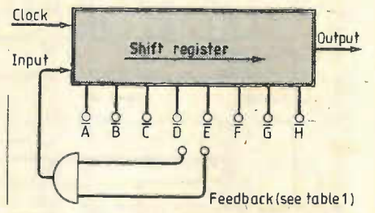

Fig 1. Basic principle of a variable length counter.

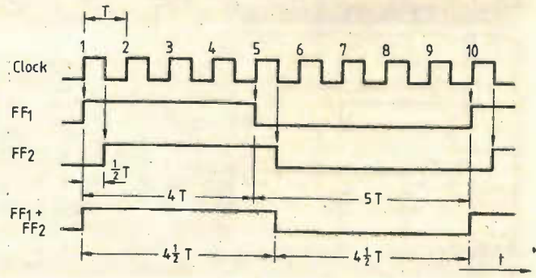

Fig 2. The addition of two asymmetrical flip-flop outputs leads to a symmetrical output.