Walsh Functions:

A Digital Fourier Series

Benjamin Jacoby PhD

Using a mathematical technique called Fourier analysis, it is possible to build arbitrary wave forms by adding together various "components."

While a full appreciation of the inner workings of the Fourier series requires a knowledge of advanced mathematics far beyond the capacity of many persons interested in electronics, that in no way deters them from using the concepts or even simplified portions of the math in practical applications. Even beginners are aware that wave forms can be broken into a set of harmonics and that a set of sinewaves of integer multiple frequencies can be summed to build up a complex wave form. In a like manner, Walsh function concepts can be put to work once a few fundamental ideas are mastered. A key to generating complicated sounds in computerized music and voice outputs is the ability to generate arbitrary wave forms from digital codes.

In these days of digital computers, a person familiar with Fourier concepts might ask the question: Is it possible to build up any wave form out of a sum of square waves of some type? Such a system would be ideal for use with digital logic. This question has been answered in the affirmative by the German mathematician H Rademacher, not in 1972 or 1962, but in 1922. His set of square waves, called "Rademacher functions," consists of a fundamental square wave of 50% duty cycle at some frequency plus harmonics of square waves of 2,4,8,16,32 and higher powers of two times the fundamental frequency. A deficiency of this system, however, is that it is not possible to generate any arbitrary wave shape from only a simple sum of these square wave harmonics.

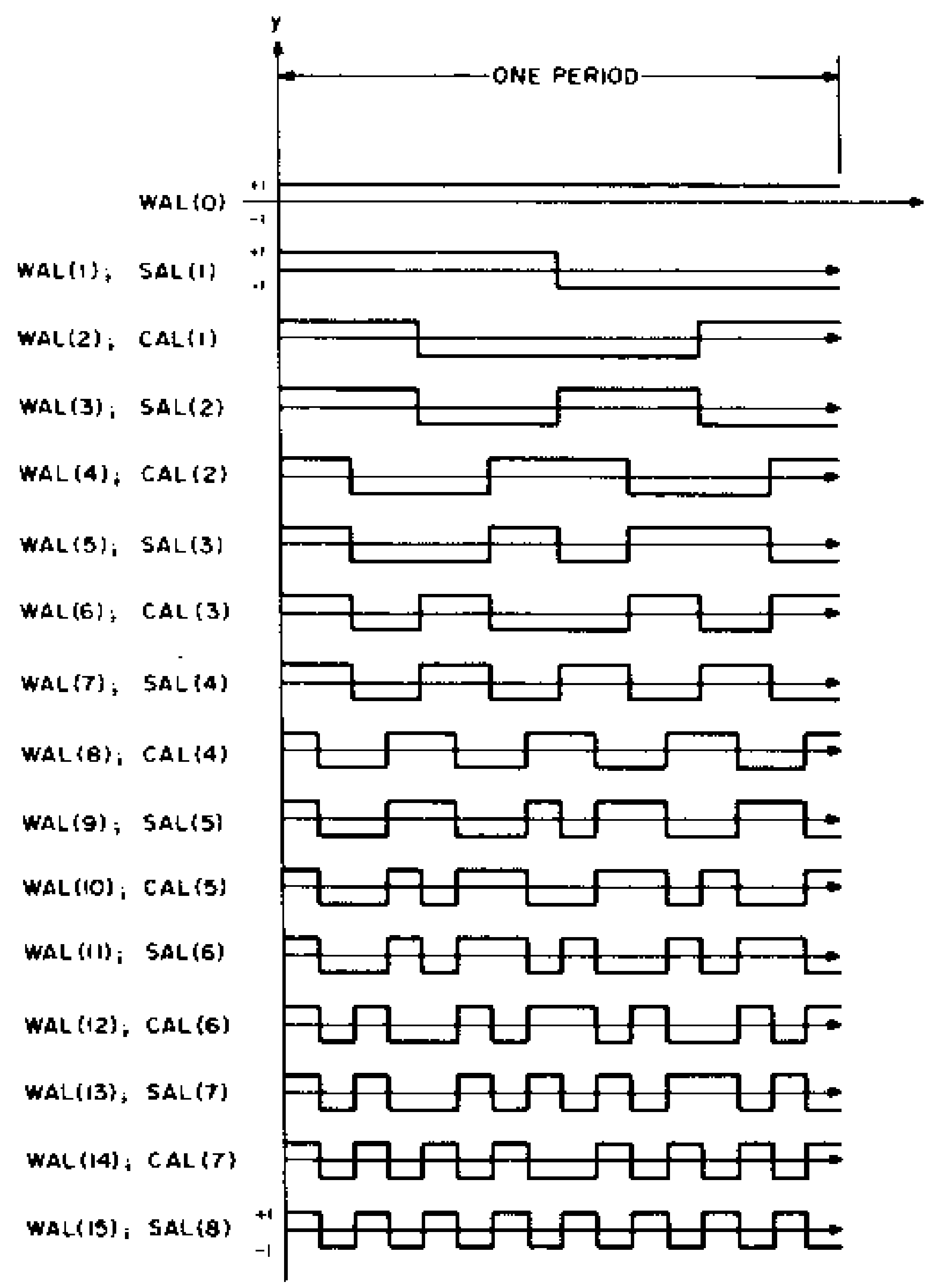

Figure 1: The Walsh Functions WAL(0) through WAL(15). The fact that Walsh functions lend themselves to digital generation is evident in the nature of the basic wave forms. The notations SAL and CAL emphasize the resemblance of Walsh functions to the Fourier series trigonometric functions SIN and COS.

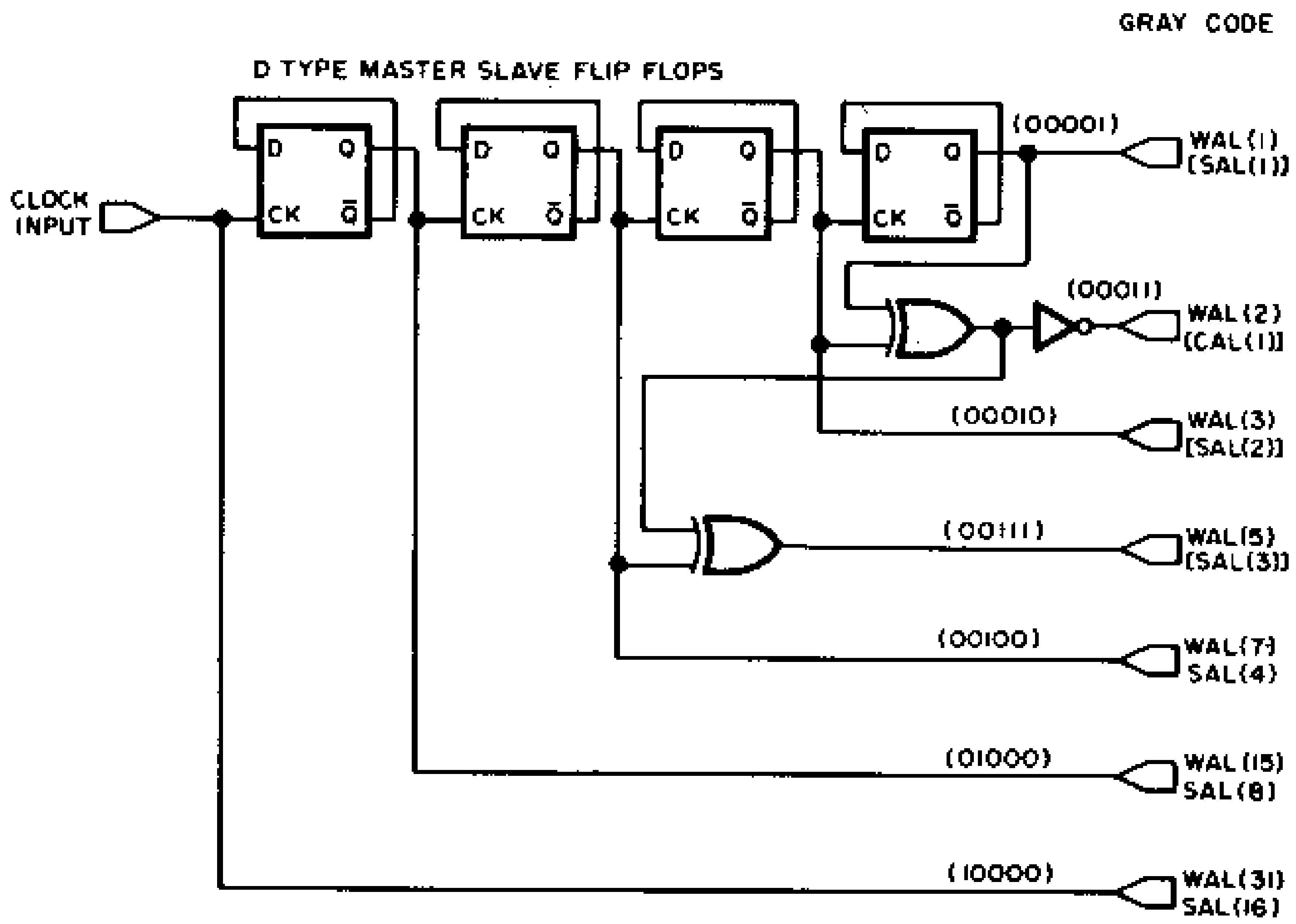

Figure 2: The logic of a digital circuit which generates a set of Walsh functions using a string of flip flops and some external gating. The flip flops are connected as toggles (division by 2 at each stage). The exclusive OR gates combine terms to produce the more complicated Walsh wave forms indicated.

Fourier series are used to create wave forms as the sum of pure sine and cosine waves at selected frequencies; this leads to the obvious question: Is it possible to use a similar mechanism which builds a complex wave form out of digital wave forms with sharp edges?

Walsh functions are the digital answer to sines and cosines used in Fourier analysis.

In translating a mathematical summation into a physical circuit, the operational amplifier provides the summing element and the resistors from inputs to the summing node form the coefficients of the component signals.

Also in 1922, J L Walsh presented his independently developed system to the American Mathematical Society. His system was later shown by the Polish mathematician Kaczmarz in 1929 to include the Rademacher system as a subset of the Walsh complete set of orthonormal functions, which, in plain English, says that some of the Walsh functions are square waves and that if all Walsh functions are allowed (you may not need to use them all, however) then any arbitrary periodic wave form can be built up by adding them together in a manner totally analagous to sinewave summation in Fourier series.

Interest in the engineering applications of Walsh functions was sparked by an article in the IEEE *Spectrum* by Dr H F Harmuth of the University of Maryland in 1968 and is continuing because of the suitability of Walsh functions to generation by digital systems.

The fastest way to understand what Walsh functions are is simply to look at a picture of some wave forms. Figure 1 shows the Walsh functions WAL(0) through WAL(15). It is seen that WAL(0) is merely a DC level which we will usually ignore in practical applications since offsets are easily handled by other means and that WAL(1), WAL(3), WAL(7), and WAL(15) are really the square wave Rademacher functions. You will note that in addition to the WAL(n) designation, the functions are also labeled with CAL or SAL. These labels are also commonly used and are acronyms for the terms Cosine wALsh and Sine wALsh by analogy to Fourier analysis. In short all WAL (even n) are called CAL and all WAL (odd n) are called SAL. CAL and SAL are also numbered but the numbers do not correspond to the WAL designation though they are easy to figure out. Also by analogy to Fourier analysis, a Walsh spectrum is called a sequency spectrum as opposed to a Fourier frequency spectrum.

Enter Mr Gray and His Code

However, knowing what Walsh functions look like and knowing how to generate them digitally are two different things. It is clear that the generation of WAL(1), WAL(3), WAL(7), WAL(15), etc, is a snap since they are simple square waves. A string of flip flops does the job, as shown in figure 2. The generation of the remaining functions, while a little more difficult, is not impossibly complex once the mathematics is shaken down into a few simple rules:

1. To generate WAL(n), first write the number n in Gray code. Gray code is a modified binary code having only one bit changing at a time when going to the next higher or next lower number. A table of Gray code numbers is

Table 1: Gray Code Bit Patterns for the Walsh Functions WAL(0) Through WAL(31). The corresponding SAL and CAL notation of each WAL function is shown down the right hand column of the table.

WALSH FUNCTION	DIGIT	WAL(31)	WAL(15)	WAL(7)	WAL(3)	WAL(1)	WALSH FUNCTION
WAL(0)	0	0	0	0	0	0	
WAL(1)	1	0	0	0	0	1	SAL(1)
WAL(2)	2	0	0	0	1	1	CAL(1)
WAL(3)	3	0	0	0	1	0	SAL(2)
WAL(4)	4	0	0	1	1	0	CAL(2)
WAL(5)	5	0	0	1	1	1	SAL(3)
WAL(6)	6	0	0	1	0	1	CAL(3)
WAL(7)	7	0	0	1	0	0	SAL(4)
WAL(8)	8	0	1	1	0	0	CAL(4)
WAL(9)	9	0	1	1	0	1	SAL(5)
WAL(10)	10	0	1	1	1	1	CAL(5)
WAL(11)	11	0	1	1	1	0	SAL(6)
WAL(12)	12	0	1	0	1	0	CAL(6)
WAL(13)	13	0	1	0	1	1	SAL(7)
WAL(14)	14	0	1	0	0	1	CAL(7)
WAL(15)	15	0	1	0	0	0	SAL(8)
WAL(16)	16	1	1	0	0	0	CAL(8)
WAL(17)	17	1	1	0	0	1	SAL(9)
WAL(18)	18	1	1	0	1	1	CAL(9)
WAL(19)	19	1	1	0	1	0	SAL(10)
WAL(20)	20	1	1	1	1	0	CAL(10)
WAL(21)	21	1	1	1	1	1	SAL(11)
WAL(22)	22	1	1	1	0	1	CAL(11)
WAL(23)	23	1	1	1	0	0	SAL(12)
WAL(24)	24	1	0	1	0	0	CAL(12)
WAL(25)	25	1	0	1	0	1	SAL(13)
WAL(26)	26	1	0	1	1	1	CAL(13)
WAL(27)	27	1	0	1	1	0	SAL(14)
WAL(28)	28	1	0	0	1	0	CAL(14)
WAL(29)	29	1	0	0	1	1	SAL(15)
WAL(30)	30	1	0	0	0	1	CAL(15)
WAL(31)	31	1	0	0	0	0	SAL(16)

GRAY CODE

shown in table 1; and with a little study, the pattern can easily be extended to any value.

2. Starting with the least significant bit, assign a square wave Rademacher function to each bit. Assign WAL(1) to the LSB, WAL(3) to the next, WAL(7) to the next, etc.
3. Any Rademacher function whose bit is 0 is not used. Those whose bits are 1 are combined by modulo 2 addition, which is to say by exclusive OR gates to give the Walsh output of that order.
4. All Walsh functions must begin positive so that the composite Walsh output may need to be inverted depending upon how many exclusive OR gates were used to produce it.

A couple of examples are shown in figure 2 and a complete generator producing all Walsh functions from WAL(1) through WAL(15) is shown in figure 3.

It should be noted that although a Walsh function is mathematically defined as going from +1 to −1, and it is possible to obtain positive and negative swings with CMOS logic with positive and negative supplies, in practice little is gained by going this route since all that is involved is a DC offset which is easily handled by the summing amplifier. Thus, 0–5 volt TTL logic outputs are fine.

Now that a set of Walsh functions has been generated, it only remains to add them in a summing amplifier with appropriate magnitudes and signs to simulate any wave form with a stair step approximation. The general expression of a Walsh function representation is a summation analogous to that found in Fourier analysis:

$$\text{Arbitrary wave form} \equiv X(t) = A_0 + \sum_{i=1}^{\infty} (A_i\, SAL(i) + B_i\, CAL(i))$$

where A_i and B_i are weighting constants which correspond to the resistors used in the summing amplifier inputs. The size of the steps and the number present will be determined by how many harmonics are combined. The more you use, the smaller and more numerous the steps, hence the better will be your approximation to your original wave form. The determination of these combining coefficients from the wave form desired requires a bit more detailed consideration.

Wave Form Synthesis

Before proceeding any further into the theoretical aspects of Walsh applications, a review of what we are attempting to do and how we intend to do it will help get our feet on solid ground. The device we wish to build using Walsh functions could be called "a square wave to arbitrary wave form converter." It will be a circuit into which you put a square wave of some frequency and out of which comes a periodic analog signal with a frequency related to that of the input wave (perhaps some submultiple) and a wave form that can be made to take any shape desired by adjusting a set of controls, switches or internal resistors. With such a device, digital logic could be used to synthesize a frequency and the converter could then be set to produce a sinewave for use in standard applications, or given sufficient accuracy of conversion, a computer could be made to talk or even sing. Both have been done by engineers working in this area.

The converter consists of two parts: The

So you want to produce a sine wave? Calculate the values at 16 evenly spaced locations in the period, then use these values to calculate the Walsh coefficients using a tabulator method. Then wire in resistors of values derived from the Walsh coefficients and the output of the circuit will be a step function approximation of the desired sine wave.

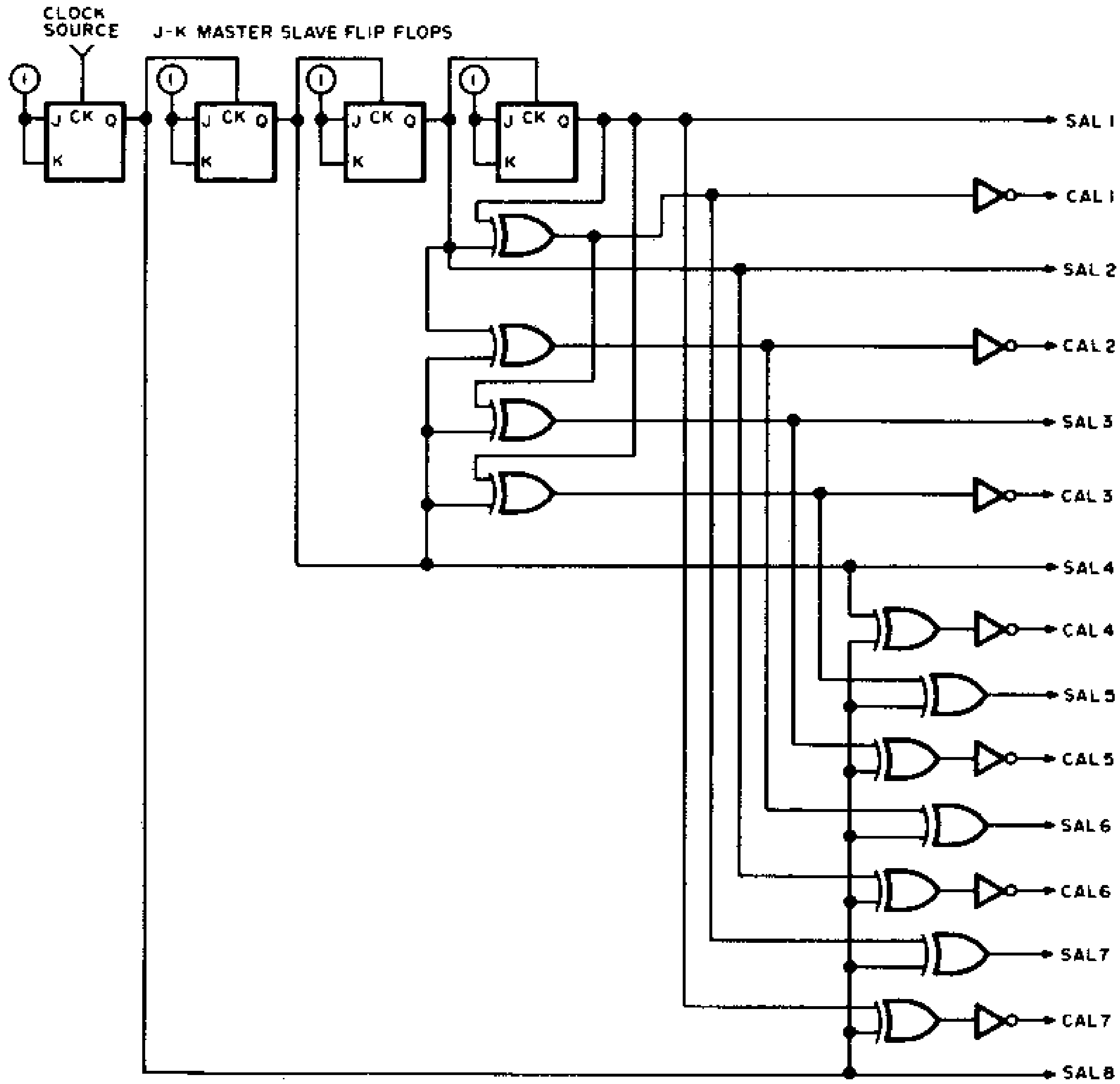

Figure 3: Extending the logic of figure 2, this circuit generates all the Walsh functions WAL(1) through WAL(15) as illustrated in figure 1. This circuit uses an alternate kind of flip flop, the JK master slave flip flop connected as a toggle. This circuit could be built with two 7473 ICs, three 7486 ICs and one 7404 circuit. (One of the 12 exclusive OR sections. is used as an inverter.)

When Walsh function analysis is applied to a linear ramp, what's the result? A set of resistor values which form an ordinary DA converter operating upon the binary value in the counter used for the Walsh function generator.

first is the digital expander which expands the input square wave into a variety of digital wave forms, and the second is the analog combiner which adds up these wave forms to produce the periodic analog output. The expander is, of course, the Walsh generator shown earlier and the combiner will be dicussed below.

All of the Walsh outputs will be fed into the summing junction of an operational amplifier, but they will not have the same strength or sign. It is the strength and sign of each component which will determine the net analog output so that once we have chosen the analog output we desire, the relative strength and sign of each Walsh harmonic must be calculated from that desired wave form. Once these values are known, a negative sign can be handled with a digital inverter and the magnitude by the choice of the resistor value into the summing junction. The net output will then be a stair step approximation to the desired output which can then be made more perfect by low pass filtering to smooth the wave shape.

Theoretically, the calculation of the coefficients from the analog wave form desired involves complex operations with the integral calculus; but it turns out that it is possible to shortcut the high powered math by starting, not with the analog signal, but rather with the stair step approximating function itself. This function can be easily determined by eyeball or by just taking the height of each step to be the value of the analog output at the center of each time interval. Figure 4 shows two examples: a linear ramp and a sinewave with 16 step approximations. The height of each step is shown.

Before proceeding to an actual calculation we will give some time and work saving rules, which are illustrated in figure 5.

1. The waveform to be synthesized must be repetitive (as in Fourier synthesis), although it is easy to start and stop at any point by control of the digital input.
2. It is especially advantageous to use 2^n steps in one period as this gives an automatic cutoff to the number of Walsh harmonics required.

Thus: With a 4 step output no functions

beyond WAL(3) are required, with an 8 step output no functions beyond WAL(7) are required, with a 16 step output no functions beyond WAL(15) are required . . .etc.

3. If the coefficients for a higher order approximation are calculated (say 16 steps), and a less accurate approximation can be used (say 8 steps) then one only need disconnect WAL(8) through WAL(15) since the lower order coefficients will have the same value in either case (or nearly so). This effect is demonstrated in the sine generator circuit.

   If your wave form to be synthesized possesses certain symmetries or can be made to do so by a DC baseline shift, many Walsh component coefficients will be zero which will not only simplify the calculations, but the circuitry as well.
4. If the wave form to be synthesized is even, which is to say that any value that the function takes to the left of center is the same as the value an equal distance to the right of center, then only CAL functions will be used and all SAL coefficients will be zero.
5. If the wave form is odd, or can be made so by a baseline shift, then only SAL functions will be used and all CAL coefficients will be zero. Here any value to the left of center equals minus the value to the right of center.

6A. If the wave form is even as in point 4 above and in addition it is even about the 1/4 point, then only CAL(k) where k is an even number will be present and all CAL(k) where k is an odd number will be zero.

6B. If the wave form is even as in point 4 above and in addition is odd about the 1/4 point, then only CAL(k) where k is an odd number will be present and all CAL(k) with k an even number will be zero.

7A. If the wave form is odd as in point 5 above and in addition is even about the 1/4 point, then only SAL(k) where k is an odd number will be present and all SAL(k) where k is an even number will be zero.

7B. If the wave form is odd as in point 5, and in addition is odd about the 1/4 point, then only SAL(k) with k an even number will be present and all SAL(k) where k is an odd number will be zero.

In the calculations that follow it will also be observed that if a wave form is even or odd, the signed sums of the step values need only be calculated for the first half of the wave form since that value will be exactly half the sum of all steps. This is probably best understood by examining some practical examples.

Two Examples

The first example will be the linear ramp. This function can be made odd by adjusting the baseline, so by rule 5 it is seen that only SAL coefficients need be calculated and no CAL functions need be generated.

The best way to get your mind right in calculating coefficients is to make a table as shown in table 2. The value desired for each

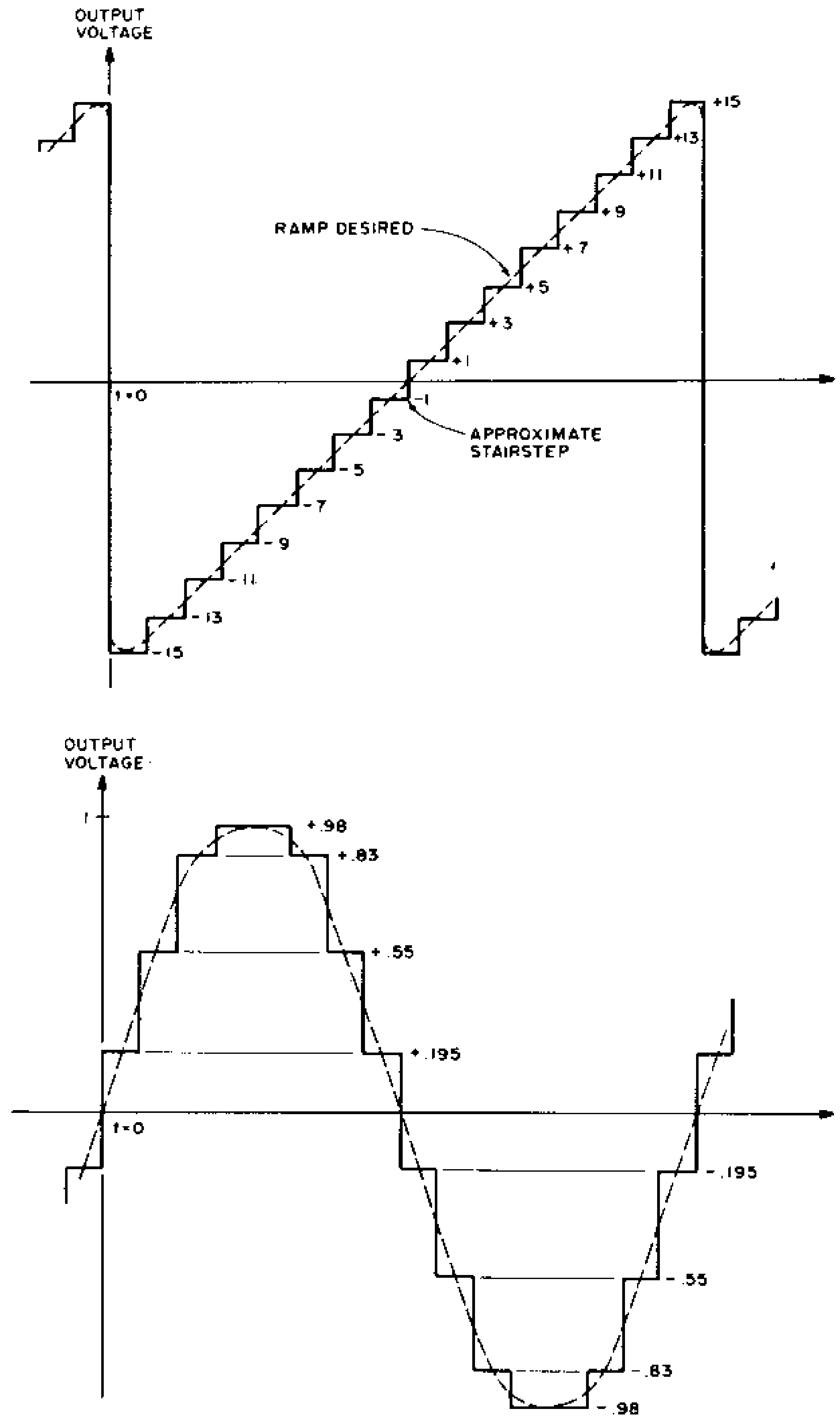

Figure 4: By picking a series of weighting constants for each Walsh function term, the outputs of figure 3 can be summed by an operational amplifier to produce arbitrary wave forms. Here are examples of the ramp and sine wave approximations generated by the Walsh function method. The smooth curve is the desired one in each case, obtained by filtering the output of the summing amplifier.

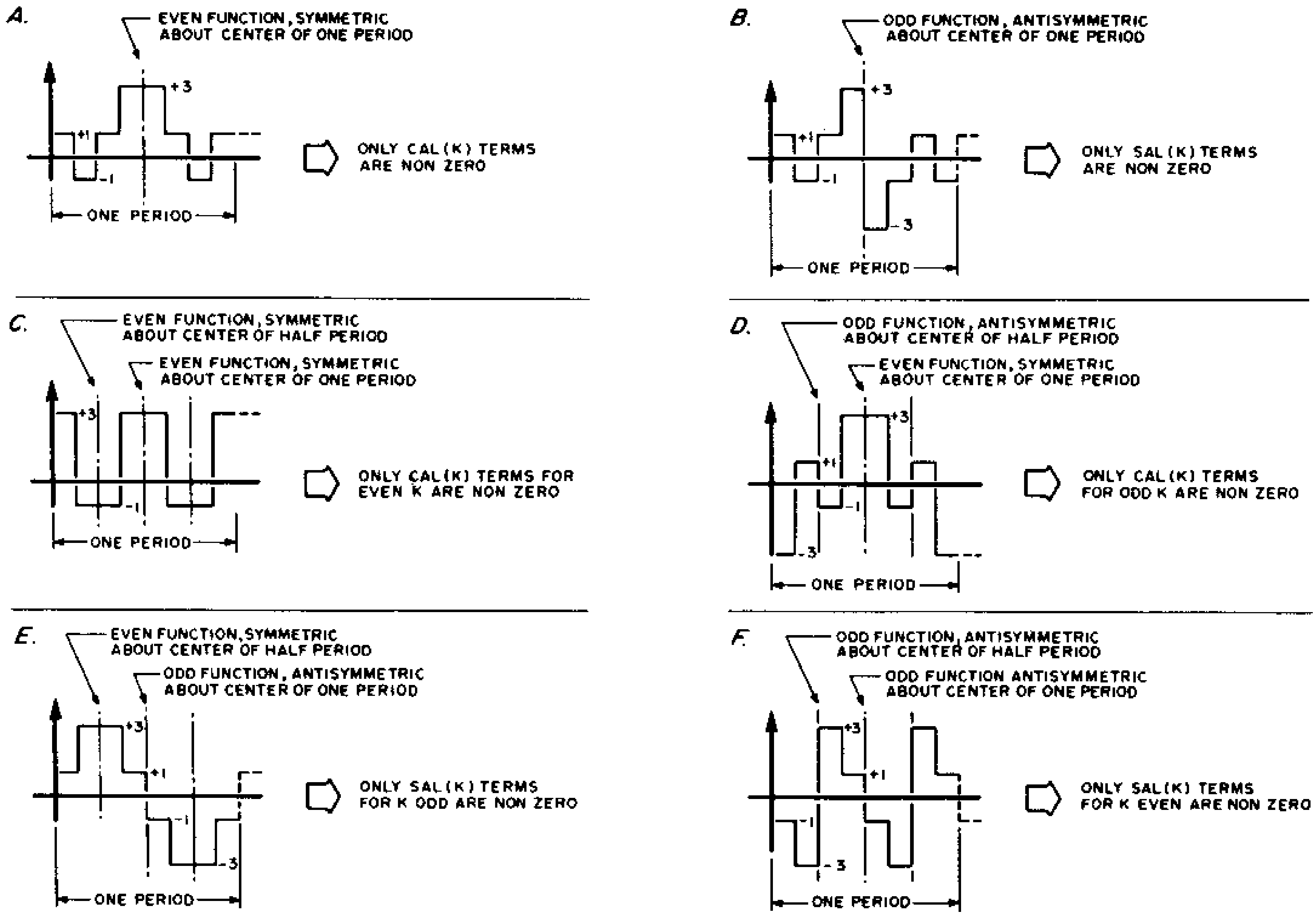

Figure 5: The properties of even and odd functions give constraints on the weighting constants needed for a given wave form. Here are illustrations of six different special cases of symmetry which give zero terms in the Walsh function sum.

Desired Function Values	−15	−13	−11	−9	−7	−5	−3	−1	+1	+3	+5	+7	+9	+11	+13	+15	Signed Sum
	ONE PERIOD																
SAL(1)	P	P	P	P	P	P	P	P	N	N	N	N	N	N	N	N	−128
SAL(2)	P	P	P	P	N	N	N	N	P	P	P	P	N	N	N	N	− 64
SAL(3)	P	P	N	N	N	N	P	P	N	N	P	P	P	P	N	N	0
SAL(4)	P	P	N	N	P	P	N	N	P	P	N	N	P	P	N	N	− 32
SAL(5)	P	N	N	P	P	N	N	P	N	P	P	N	N	P	P	N	0
SAL(6)	P	N	N	P	N	P	P	N	P	N	N	P	N	P	P	N	0
SAL(7)	P	N	P	N	N	P	N	P	N	P	N	P	P	N	P	N	0
SAL(8)	P	N	P	N	P	N	P	N	P	N	P	N	P	N	P	N	− 16

Table 2: A computational table used to help determine the Walsh function coefficients for the linear ramp. The relative strength of the SAL or CAL term in question is obtained by summing horizontally the +1(P) or −1(N) Walsh function value multiplied by the actual waveform value desired for that element of time. After figuring out the value of the signed sum for each term, the values should be normalized so that the largest magnitude is 1 (regardless of sign). Thus the normalized ratios shown below this picture were computed assuming −128 corresponded to −1.

SIGN OF WALSH FUNCTIONS

P = positive N = negative

Normalized Ratio

SAL(1) = −1 = −1
SAL(2) = −0.5 = −1/2
SAL(4) = −0.25 = −1/4
SAL(8) = −0.125 = −1/8

step comprising the output function is written in order along the top of the table. Since we are attempting to produce a linear ramp, our output will be a rising staircase with a fixed increase with each step (we used two units per step). This staircase will eventually be filtered to remove the jogs and give a linear ramp.

The body of the table shows the sign (positive or negative) each particular Walsh function takes in each of the 16 time intervals into which one period of the output wave form has been divided. As indicated earlier, we need not go past WAL(15) in this case. The Walsh sign values can be taken from the wave forms of figure 1 or from table 3 which is good for up to 32 segment approximations.

The numbers to the far right are the sums of the upper values when all signs are taken into account. Thus, for WAL(1) we see that it is positive in the first half period, but the step values are negative, so we get:

(−15) + (−13) + (−11) + (−9) + (−7) + (−5) + (−3) + (−1) = −64 and in the second half period where WAL(1) is negative and the values positive we get:

−(+1) − (+3) − (+5) − (+7) − (+9) − (+11) − (+13) − (+15) = −64 or a total of −128. This number gives the relative strength of WAL(1) in the output summation. We repeat the process for each Walsh function.

If we divide all nonzero values by the largest (WAL(1)), it is observed that the weighting is binary and further it is seen that only the square wave Rademacher functions are nonzero. Thus, it is seen that the way to generate a ramp is with a counter feeding a standard digital to analog converter. (So here we have a long, complicated way of arriving at an "obvious" result, but it also should be noted that D to A binary weighting is *only* "matched" to a ramp output.)

If another wave form such as a sinewave is desired, a D to A converter could be used, but a more accurate method would be to switch between 16 voltages of appropriate values. The Walsh system is just as accurate and is simpler for the more general case.

If we divide a sinewave into 16 portions, the value at the center of the first interval will be Sin (11.25°) = 0.19509 and the next will be Sin (33.75°) = 0.55557 and the next Sin (56.25°) = 0.83147, etc. This produces the top row of our table. Since Sin(x) is an odd function, even about the 1/4 point, only SAL(1), SAL(3), SAL(5) and SAL(7) are calculated over the first half period. Our chart with the calculated coefficient values is shown in table 4. Since in a standard operational amplifier summing circuit (we won't go into details here as they can be found in any book on operational amplifiers), the relative summing ratios are related to the inverse of the summing resistor values, we divide each normalized value into 1 and multiply by the feedback resistor value to obtain

$\frac{1}{A_i} \times 10k$	1%	5% EIA
10.00k	10.0k	10k
24.14k	24.3k	24k
121.4 k	121 k	120k
50.27k	49.9k	51k

Table 5: The EIA resistor equivalents for the calculated values of table 4. The 5% tolerance resistance values shown at the right were used in the circuit of figure 6.

The Sign of CAL and SAL in Each 1/32 Interval of Their Period

WAL(0)	PPPP	PPPP	PPPP	PPPP	PPPP	PPPP	PPPP	PPPP
SAL(1)	PPPP	PPPP	PPPP	PPPP	NNNN	NNNN	NNNN	NNNN
CAL(1)	PPPP	PPPP	NNNN	NNNN	NNNN	NNNN	PPPP	PPPP
SAL(2)	PPPP	PPPP	NNNN	NNNN	PPPP	PPPP	NNNN	NNNN
CAL(2)	PPPP	NNNN	NNNN	PPPP	PPPP	NNNN	NNNN	PPPP
SAL(3)	PPPP	NNNN	NNNN	PPPP	NNNN	PPPP	PPPP	NNNN
CAL(3)	PPPP	NNNN	PPPP	NNNN	NNNN	PPPP	NNNN	PPPP
SAL(4)	PPPP	NNNN	PPPP	NNNN	PPPP	NNNN	PPPP	NNNN
CAL(4)	PPNN	NNPP	PPNN	NNPP	PPNN	NNPP	PPNN	NNPP
SAL(5)	PPNN	NNPP	PPNN	NNPP	NNPP	PPNN	NNPP	PPNN
CAL(5)	PPNN	NNPP	NNPP	PPNN	NNPP	PPNN	PPNN	NNPP
SAL(6)	PPNN	NNPP	NNPP	PPNN	PPNN	NNPP	NNPP	PPNN
CAL(6)	PPNN	PPNN	NNPP	NNPP	PPNN	PPNN	NNPP	NNPP
SAL(7)	PPNN	PPNN	NNPP	NNPP	NNPP	NNPP	PPNN	PPNN
CAL(7)	PPNN	PPNN	PPNN	PPNN	NNPP	NNPP	NNPP	NNPP
SAL(8)	PPNN	PPNN	PPNN	PPNN	PPNN	PPNN	PPNN	PPNN
CAL(8)	PNNP	PNNP	PNNP	PNNP	PNNP	PNNP	PNNP	PNNP
SAL(9)	PNNP	PNNP	PNNP	PNNP	NPPN	NPPN	NPPN	NPPN
CAL(9)	PNNP	PNNP	NPPN	NPPN	NPPN	NPPN	PNNP	PNNP
SAL(10)	PNNP	PNNP	NPPN	NPPN	PNNP	PNNP	NPPN	NPPN
CAL(10)	PNNP	NPPN	NPPN	PNNP	PNNP	NPPN	NPPN	PNNP
SAL(11)	PNNP	NPPN	NPPN	PNNP	NPPN	PNNP	PNNP	NPPN
CAL(11)	PNNP	NPPN	PNNP	NPPN	NPPN	PNNP	NPPN	PNNP
SAL(12)	PNNP	NPPN	PNNP	NPPN	PNNP	NPPN	PNNP	NPPN
CAL(12)	PNPN	NPNP	PNPN	NPNP	PNPN	NPNP	PNPN	NPNP
SAL(13)	PNPN	NPNP	PNPN	NPNP	NPNP	PNPN	NPNP	PNPN
CAL(13)	PNPN	NPNP	NPNP	PNPN	NPNP	PNPN	PNPN	NPNP
SAL(14)	PNPN	NPNP	NPNP	PNPN	PNPN	NPNP	NPNP	PNPN
CAL(14)	PNPN	PNPN	NPNP	NPNP	PNPN	PNPN	NPNP	NPNP
SAL(15)	PNPN	PNPN	NPNP	NPNP	NPNP	NPNP	PNPN	PNPN
CAL(15)	PNPN	PNPN	PNPN	PNPN	NPNP	NPNP	NPNP	NPNP
SAL(16)	PNPN	PNPN	PNPN	PNPN	PNPN	PNPN	PNPN	PNPN

← 1 Period →

P = Positive N = Negative

(Columns only for ease of reading.)

Table 3: A larger computational table giving 32 Walsh function components and their signs during a 32 interval period.

	SIN(11.25°) = 0.19509	SIN(33.75°) = 0.55557	SIN(56.25°) = 0.83147	SIN(78.75°) = 0.98078	SIN(101.25) = 0.98078	SIN(123.75) = 0.83147	SIN(146.25) = 0.55557	SIN(168.75) = 0.19509	signed sum	Normalized coefficients = A_i
SAL(1)	P	P	P	P	P	P	P	P	−5.1258	−1
SAL(3)	P	P	N	N	N	N	P	P	+2.1232	+0.4142
SAL(5)	P	N	N	P	P	N	N	P	+0.4223	+0.08239
SAL(7)	P	N	P	N	N	P	N	P	+1.0196	+0.1989

← 1/2 period →

Table 4: Using the computational table to calculate the resistor values for a 16 step sine wave approximation. The specialized sine wave generator of figure 6 uses these results, subject to a further approximation shown in table 5. Note that the signs of the coefficients take into account the inverting op amp configuration and thus appear reversed.

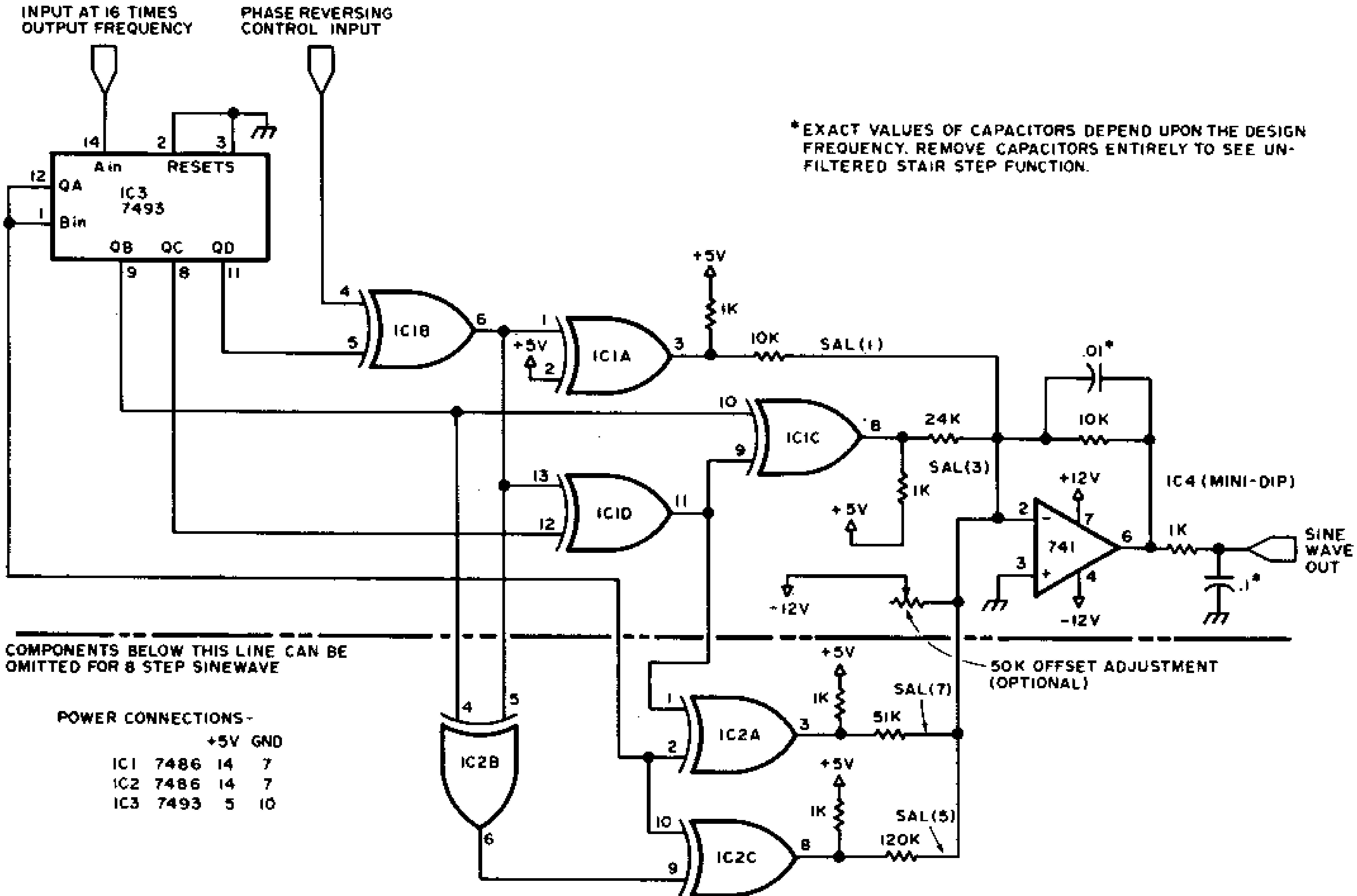

Figure 6: Applying Walsh Functions. Here is the circuit of a sine wave generator which produces a Walsh function approximation of the sine function. The frequency of the sine wave is set by the input to pin 14 of the 7493. Filtering components of the operational amplifier help smooth out the staircase wave form generated by summing the Walsh function components as weighted by resistors.

each summing resistor value in ohms. Table 5 shows the calculated values compared to 1% and 5% EIA resistor standard values.

The total sinewave converter circuit is shown in figure 6. While three of the coefficients were negative, a single inverter was used on the lone positive Walsh output since the op amp inverts the wave form. In addition, a gate has been added by which the phase of the entire output wave form can be inverted by simultaneously inverting all Walsh components. It is interesting to also note that if the components below the dotted line are removed, an 8 step sinewave approximation results. The feedback capacitor and output low pass filter can be added to smooth up the wave form to give a nearly perfect sinewave.

The Walsh methods presented here would seem to have wide application for experimentation and engineering. Although these concepts are based on advanced mathematics, nevertheless, as the philosopher Seneca observed so many years ago, "The language of truth is simple."

Walsh Functions for Music Synthesis?

Some background information on the use of orthogonal functions in music wave form synthesis has been generated by Hal Chamberlin, and published in *Electronotes Newsletter*, Volume 4, Number 25, July 20 1973. Hal also sent along a copy of a portion of a report by B A Hutchins, 60 Sheraton Dr, Ithaca NY 14850, on the use of Walsh functions in wave form generation. According to Hal, there was considerable analysis of Walsh functions in electronic music circles during a period of time approximately centered on 1973, but complexities of controlling the Walsh harmonic amplitudes digitally led to the demise of that interest. Hal's current approach is to employ a real time Fourier series evaluation module which digitally sums terms of the first 32 components of a Fourier series, specified to 8 bit accuracy both in amplitude and phase.

GLOSSARY

The following terms may be unfamiliar to some readers and are highlighted with further explanations.

Baseline: It is possible to add a fixed DC level to an analog signal, which will not affect its wave form. Using the 0 V and +5 V levels obtained with TTL circuits (using pull up resistors) as "Walsh functions" corresponds to a baseline adjustment of +2.5 volts to the ideal case of a symmetric positive or negative voltage value.

CAL: An acronym derived from Cosine wALsh. The CAL functions are the "even" Walsh functions, analogous to the Fourier cosine functions.

Duty cycle: For a digital wave form, the duty cycle is the percentage of time spent in the high state relative to the full period of the wave form.

Even function: An even function (or wave form) is one which is symmetric about the center point of its period. This means that its value a certain distance to the left of center is the same as its value the same distance to the right of center.

Fundamental: The lowest frequency in a Fourier or Walsh function summation.

Gray code: A binary code modified so that only one bit changes when going to the next higher or lower number. It is often used to deglitch position encoders.

Harmonic: A frequency which is a multiple of the fundamental frequency.

Integral calculus: The mathematical formalism used to calculate the area under a curve. The integral calculus is used together with the theory of orthogonal functions to evaluate analytically the coefficients of Fourier and Walsh function expansions. The example of Walsh function coefficient calculation in this article uses properties of Walsh functions to simplify the process of calculating integrals required for the coefficients. There is no such simplification for the Fourier coefficients of a wave form, thus making the application of Fourier analysis a more complicated problem.

Odd function: An odd function (or wave form) is one which is antisymmetric with respect to the center point of its period. This means that if at a fixed interval before the center point its value is X, then at the same interval past the centerpoint the value will be −X.

Orthonormal functions: The mathematical theory of orthonormal functions is one of the most powerful tools used by physicists, theoretical chemists and engineers. Among other applications, it provides the tools needed to analyze complex wave forms and synthesize such wave forms using the principle of superposition: That the whole is a linear sum of its parts. Fourier series and Walsh function analysis mentioned here are two particular choices of a set of orthonormal functions which have useful practical applications. (See also **spectrum** below.)

Periodic wave form: A periodic wave form is one which has a fixed shape which is constantly repeated. A simple example would be the clock oscillator signal of a typical home brew central processor. A more complicated example (subject to imperfections) would be a long steady tone played on a musical instrument.

Rademacher functions: The subset of Walsh components consisting of only the unmodified square waves.

SAL: An acronym derived from Sine wALsh. The SAL functions are the "odd" Walsh functions, analogous to the Fourier sine functions.

Sequency: Walsh function terminology referring to the Walsh components of a wave form in exactly the same way that frequency is used to refer to the Fourier components. Example: Sequency spectrum.

Spectrum: When orthonormal functions are used to analyze a wave form, the result frequently is a set of coefficients which weigh each of the basic functions found in a (theoretically) infinite sum which represents the wave form. Each coefficient corresponds to some parameter of the orthonormal functions, which might be, for example, a number "n." Whatever the parameter is, a spectrum for the analysis is obtained by plotting the coefficient values versus the parameter value for a large number of coefficients. For a Fourier analysis, the result is a plot of coefficient versus frequency (which at the low end corresponds to a small integer value). A Walsh spectrum would plot the coefficient of WAL (n) versus n.

Wave form: For the purposes of this article, a signal's wave form is a value of (for example) voltage as a function of time.■

REFERENCES

1. Corrington, M S, "Solution of Differential and Integral Equations with Walsh Functions," *IEEE Transactions on Circuit Theory,* volume CT-20, number 5, September 1973.
2. Harmuth, H F, "Applications of Walsh Functions in Communications," *IEEE Spectrum,* November 1969.
3. Rademacher, H, "Einige-Saltze von allegemeinen Orthogonalfunktionen," *Math Annalen,* volume 87, 1922, pages 112 to 138.
4. Walmsley, W M, "Walsh functions, transforms and their applications," *Electronic Engineering,* June 1974.
5. Walsh, J L, "A Closed Set of Orthonormal Functions," *American Journal of Math,* volume 45, 1923, pages 5 to 24.