AD-A057 033 ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 17/2
HARDWARE DESIGN OF A DIGITAL SINGLE-SIDEBAND GENERATOR.(U)
OCT 77 M H ETZEL DAAB07-72-C-0259
UNCLASSIFIED R-797 NL

| OF |
AD
A057033

END
DATE
FILMED
9-78
DDC

AD A057033

AD No.
DDC FILE COPY

REPORT R-797 OCTOBER, 1977 UILU-ENG 77-2243

CSL COORDINATED SCIENCE LABORATORY

LEVEL II

12

HARDWARE DESIGN OF A DIGITAL SINGLE-SIDEBAND GENERATOR

MARK HOWARD ETZEL

DDC
DRAPER
JUL 18 1978
E

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

Rept. no. UILU-ENG-77-2243 has used on
AD-A056110 also. Per Barbara Champagne, Ill. Univ,
leave as is.

M. Crumbacker
DDC-DDA-2
7 Aug 78

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

see note on cover (inside) re other AD same report

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. (9) Master's thesis, 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) HARDWARE DESIGN OF A DIGITAL SINGLE-SIDEBAND GENERAL	5. TYPE OF REPORT & PERIOD COVERED Technical Report
	6. PERFORMING ORG. REPORT NUMBER (14) R-797; UILU-ENG 77-2243
7. AUTHOR(s) (10) Mark Howard Etzel	8. CONTRACT OR GRANT NUMBER(s) (15) DAAB 07-72-C-0259, NSF-INT 76-84216
9. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (12) 52p.
11. CONTROLLING OFFICE NAME AND ADDRESS (11) Joint Services Electronics Program	12. REPORT DATE Oct 77; 13. NUMBER OF PAGES 45
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED; 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DDC RECEIVED JUL 18 1978 E

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Single-Sideband Generator

Weaver Modulator

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Single-sideband (SSB) modulation and transmission has become increasingly important in communications systems over the past fifty years. Its success stems partially from its well-known advantages over other standard modulation techniques. These advantages include lower transmitting power for equal signal to noise ratio, the fact that no carrier is necessary for transmission, minimum bandwidth in the transmission medium is utilized, and no phase information is necessary for detecting voice signals.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

78 07 13 020

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

097700

UILU-ENG 77-2243

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

HARDWARE DESIGN OF A DIGITAL SINGLE-SIDEBAND GENERATOR

by

Mark Howard Etzel

This work was supported in part by the Joint Services Electronics program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract DAAB-07-72-C-0259 and in part by the National Science Foundation under Grant NSF INT 76-84216.

Reproduction in whole or in part is permitted for any purpose of the United States Government.

Approved for public release. Distribution unlimited.

HARDWARE DESIGN OF A DIGITAL SINGLE-SIDEBAND GENERATOR

BY

MARK HOWARD ETZEL
B.S., Carnegie-Mellon University, 1975

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1978

Thesis Adviser: Professor Timothy N. Trick

Urbana, Illinois

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Timothy N. Trick, for his aid and guidance in the research and assembly of this thesis.

I would also like to express my appreciation to Hazel Corray for typing this paper, and to the draftsmen of the Coordinated Science Laboratory for preparing the figures.

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	WEAVER MODULATOR	3
III.	DESIGN	7
	A. General Discussion	7
	B. Sampling and Baseband Multiplication	9
	C. Hardware Digital Filter	13
	D. Low Pass Filter Characteristics	24
	E. Modulation and Cancellation	36
	F. Control Signals	38
IV.	CONCLUSIONS	42
REFERENCES		44

I. INTRODUCTION

Single-sideband (SSB) modulation and transmission has become increasingly important in communications systems over the past fifty years. Its success stems partially from its well-known advantages over other standard modulation techniques. These advantages include lower transmitting power for equal signal to noise ratio, the fact that no carrier is necessary for transmission, minimum bandwidth in the transmission medium is utilized, and no phase information is necessary for detecting voice signals [1].

The latter advantage mentioned, in particular, affords the telephone communications industry an economical method of modulation and transmission so that SSB is predominant in telephone systems around the world. SSB is primarily used with frequency-division multiplex (FDM) systems in which case several modulation steps occur. Twelve voiceband channels are single-sideband modulated and frequency-division multiplexed to form a new channel. Several of these in turn are modulated and frequency-multiplexed again. This can be repeated until a final channel is formulated for transmission. To recover the voiceband signals, an opposite but similar sequence is employed.

Different techniques for SSB generation exist, most notably amplitude modulation with elimination of the unwanted sideband by filtering, the Hartley modulator, and the Weaver modulator. However, only the first of these is commonly found in the analog systems in use today, the reason being that both the Hartley and Weaver modulators are cancellation types. That is, two paths are added to cancel the unwanted sideband, and limited long-term stability of analog networks in the cancellation process causes

inaccuracies that render these modulators impractical. Digital implementation on the other hand, can provide the necessary precision in the two signal paths to make cancellation techniques viable.

With this knowledge, it was the author's intent to design a digital hardware implementation of a Weaver modulator for the modulation and frequency-multiplexing of voiceband signals with a 4 KHz bandwidth, to the standard frequency range of 60-108 KHz, generally known in the telephone industry as the A-type channel bank [2]. The modulator should operate as fast as possible, limited only by the characteristics of available, standard transistor-transistor logic (TTL) integrated circuits (IC's). Thus it was to be determined exactly how economical this method is, and exactly how many voiceband channels or how much information can be time-multiplexed through a minimum of hardware. To this end, a new and substantially faster implementation of a digital filter, created by Peled and Liu [3,4], was designed for the low pass filter necessary in the Weaver technique. Yet despite this advantage, it was discovered that only data in one path of the Weaver modulator could be run through the hardware designed. For a minimum of hardware, one second-order filter section is used and time-multiplexed four times to yield effectively an eighth-order filter.

This paper presents the design of that hardware with an accompanying discussion of the tradeoffs, improvements, and comparisons with other designs.

II. WEAVER MODULATOR

In 1956 D. K. Weaver proposed a new method for SSB generation [5]. Known as a double quadrature modulation scheme, its block diagram is shown in Figure 2.1. It should be noted that a variation of this using a high pass filter also exists [6].

The sequence of frequency spectra shown in Figure 2.2 gives a graphic illustration of the modulator with reference to the signals labeled in Figure 2.1. Note that the spectra show overlapping sidebands as distinct. The input signal spectrum is that of a signal already sampled at the Nyquist rate, $4\omega_o$, to avoid aliasing. The first modulation at a frequency equalling half the bandwidth of the voiceband signal overlaps the two sidebands and introduces a 90° phase shift between the two paths. The low pass filter eliminates all frequencies above ω_o leaving one set of overlapped sidebands. The second modulation shifts the overlapped pair into the final desired frequency range, but because of the quadrature paths, another 90° phase shift occurs causing one sideband to be 180° out of phase with the same sideband in the other path. At this point, the two paths can either be added or subtracted yielding either the lower or upper sideband as the designer chooses.

It is important to note that the digital filter must operate fast enough to avoid aliasing during the final modulation. As a numerical example, consider the specifications used by the telephone company. Each voiceband channel is limited to 4 KHz, and twelve voiceband channels are to be frequency-multiplexed to the range 60-108 KHz. Taking advantage of the empty space from 0-60 KHz, the folding frequency can fall anywhere between 54 KHz and 60 KHz to allow total, accurate recovery of the signal.

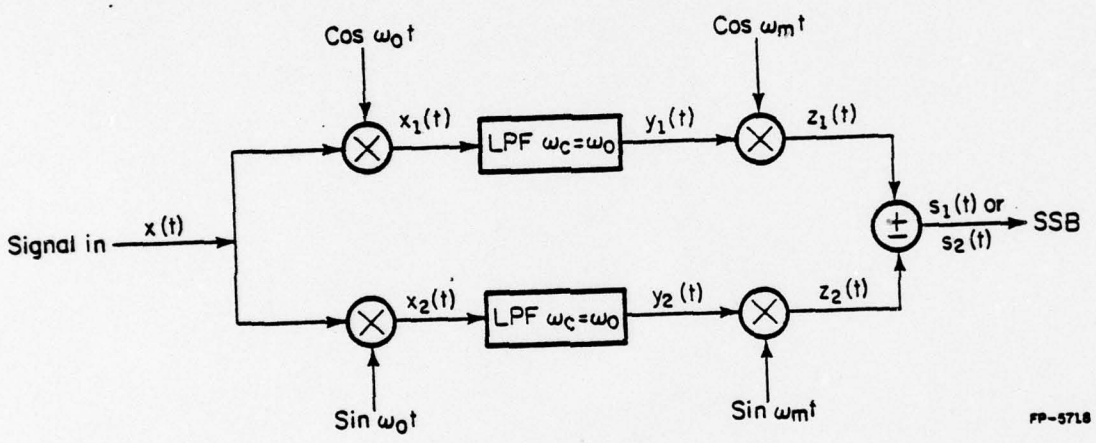


Figure 2.1. Block Diagram of Weaver Modulator

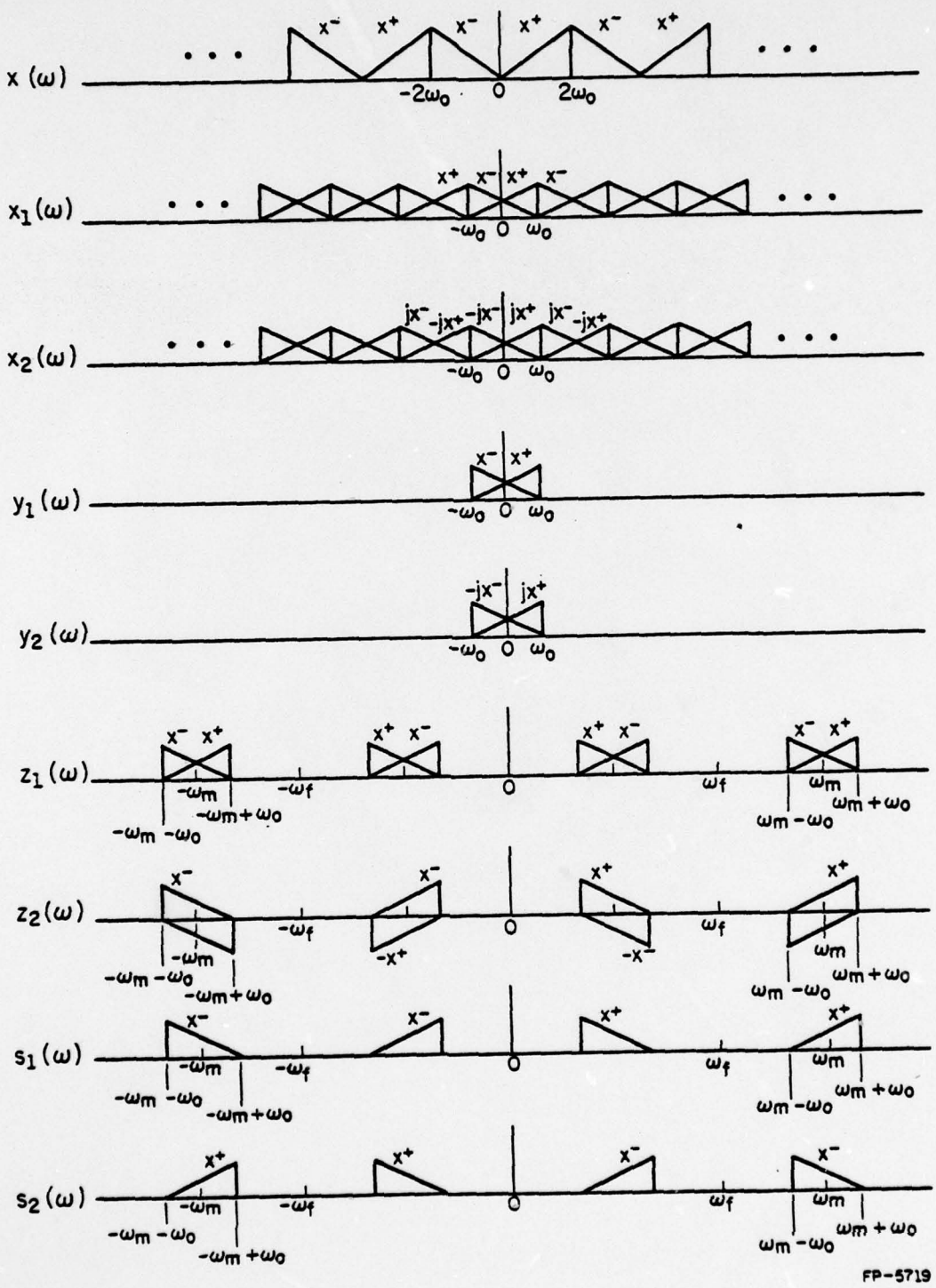


Figure 2.2. Frequency Spectra Sequence of Weaver Modulator

The desired signal will occur between 60-108 KHz with its image appearing between 0 and the folding frequency as shown in Figure 2.2. Since the signal and its image do not overlap, there is no distortion.

This, then, forces the word rate of the filter to be 108-120 KHz. Therefore an option arises. Either the voiceband can be sampled at 108-120 KHz or, since it need only be sampled at 8 KHz to avoid aliasing, zeros may be inserted between samples. This has the effect of leaving the voiceband spectrum unaltered, but shifts the folding frequency higher. The latter approach is used in this design. The folding frequency desired is 56 KHz leaving a guard band on either side, and so the correct filter rate was obtained by sampling the voiceband at an 8 KHz rate and inserting thirteen zeros between each sample.

Further simplifications in this structure and the hardware implementation of this modulator are discussed in the following chapter.

III. DESIGN

A. General Discussion

The design of a digital Weaver modulator was guided by several criteria. The first was a decision to use TTL logic, which, although not the fastest logic on the market, nor the lowest in power, is certainly the most prevalent, easily obtainable, relatively inexpensive, and offers the greatest variety of functions on a chip.

As previously mentioned, the second design consideration was speed. In the Weaver modulator, the low pass filter is the critical section for speed. The speed of the filter essentially determines the speed of the entire modulator, and thus the amount of information that can be time-multiplexed through it. For this reason, the filter was designed using the latest and, to the author's knowledge, fastest digital filter implementation, again only limited by the speed of Schottky TTL IC's.

Package count became the third criterion. Once maximum speed was acquired, the circuits were designed to keep the number of IC's to a minimum, thus generally insuring lower cost and lower power consumption. In sections where utmost speed was not a factor, low power devices are called for. Generally speaking, once the filter was designed for speed, the remainder of the modulator was developed as economically as possible with regard to cost and power consumption.

The implementation of the low pass filter and succeeding modulation are detailed for only one path of the Weaver modulator, the second path being identical in hardware. The only difference between paths is read-only memory (ROM) entries for different modulation coefficients. Hence the modulator can be separated into three basic sections as shown in Figure 3.1. Each block will be considered in following sections.

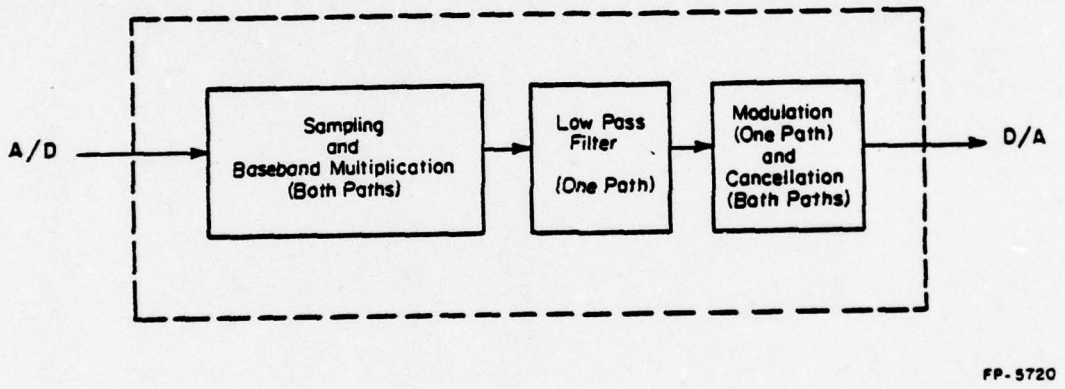


Figure 3.1. Sections of the Design of Weaver Modulator

B. Sampling and Baseband Multiplication

For the design of the sampling and baseband multiplication section of the Weaver modulator, it is assumed that a voiceband channel has a bandwidth of 4 KHz. In practice the energy to be transmitted is concentrated between .2 and 3.4 KHz, allowing a guard band so that realizable filters may be constructed [1]. To avoid aliasing, then, the sample rate of any given voiceband channel must be at least 8 KHz, the Nyquist rate. Note that the next step in a Weaver modulator is the multiplication of the signal by either a sine or cosine, depending which path is considered, at a frequency of half the voiceband bandwidth. In this case, the frequency is 2 KHz. If the sampling is performed precisely at the Nyquist rate, a fortunate set of numbers results. Taking the cosine path as an example, the multiplication term would be

$$\cos (2\pi\ 2000\ nT),\ n = 0,1,2,\ldots,$$

where n corresponds to the nth sample and T is the sample period. At the Nyquist rate,

$$T = \frac{1}{8000} \text{ sec},$$

so the multiplication term becomes

$$\cos n\frac{\pi}{2},\ n = 0,1,2,\ldots$$

Similarly for the quadrature path, the multiplication term is

$$\sin n\frac{\pi}{2},\ n = 0,1,2,\ldots$$

Both terms give rise to the sequence 1,0,-1,0, although when one term is zero, the other is not. This idea is illustrated in Figure 3.2 [7].

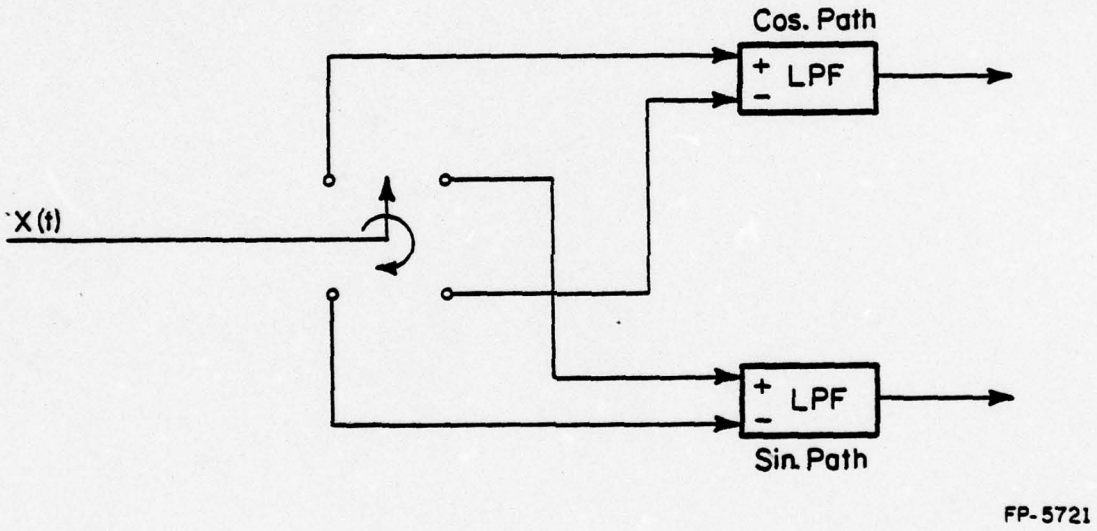


Figure 3.2. Block Diagram of Sampling and Baseband Multiplication

The mechanism is shown as a rotating arm that takes samples of the signal x(t) and alternately inputs them to the quadrature paths. The - and + indicate whether the sample is inverted or not, respectively, as it is input to the low pass filters. This important combination of sampling and voiceband modulation affords a great savings in hardware implementation.

Recall that zeros must be inserted to increase the folding frequency to such that aliasing presents no problem after the final modulation. Specifically, thirteen zeros must be inserted between each sample to change the folding frequency from 4 KHz to 56 KHz. Yet for a given path, every other sample is zero so that actually for each path, every non-zero sample is separated by twenty-seven zeros.

This section is detailed in Figure 3.3. The analog input signal is sampled at an 8 KHz rate by an analog to digital converter (ADC). No particular ADC has been specified and the necessary sample and hold signals have not been incorporated. It is assumed that the digital input and output of the Weaver modulator will be represented in 8-bit words.

The output of the ADC connects in parallel to the beginning of the two quadrature paths at the inputs of identical sets of arithmetic logic units (ALU's), 74S181's. These were chosen for their ease of selecting the output to be either zero, equal to an input, or equal to an input inverted. The choice is made simply by changing the select inputs S_o-S_3 and the carry input C_n. The outputs of the ALU's are connected to the input register, RXO, of the low pass filter for each path. The 74LS161 counters are clocked at a rate of 56 KHz, the input word rate of the low pass filters. Together the pair has a cycle of twenty-eight counts: after each fourteen, one path, but only one, receives

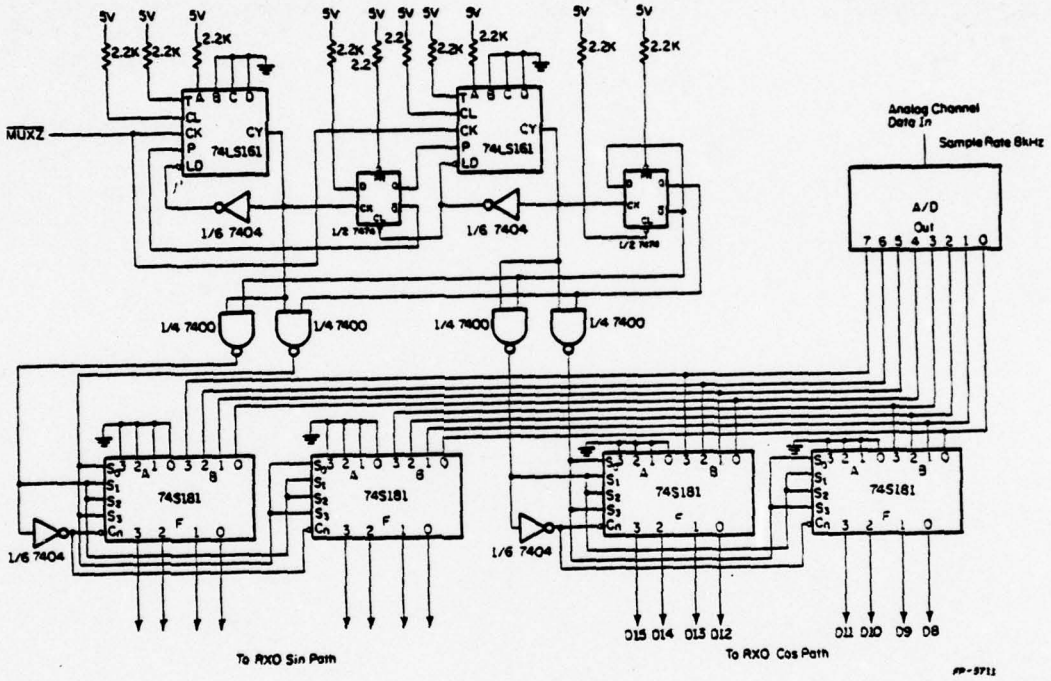


Figure 3.3. Schematic of Sampling and Baseband Multiplication

the sampled signal from the ADC, passed through the ALU's either directly or inverted depending upon the state of the corresponding flip-flop. Thus each path receives a sample every twenty-eight counts, realizing the alternate inversion as well as the necessary twenty-seven zeros between each sample. The complete sampling and voiceband modulation is accomplished, then, in the simple, concise circuitry shown.

C. Hardware Digital Filter

At the end of 1974, Peled and Liu introduced a new implementation of a digital filter which has basic advantages over previous hardware designs [3,4]. Its strength lies in the elimination of actual multipliers in the filter realization; only ROM look-up and successive additions are needed. Without multiplier chips, which are generally the elements limiting the speed of a filter, this new realization not only operates at a faster rate than possible before, but also operates with less hardware and reduced power consumption, and in turn, reduced cost.

In particular, Peled and Liu's method lends itself to the realization of a second-order section of a digital filter. Consider the standard equation for a second-order recursive digital filter:

$$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} - b_1 y_{n-1} - b_2 y_{n-2} \tag{1}$$

where $\{y_n\}$ represents the output sequence, $\{x_n\}$ the input sequence, and the a's and b's the corresponding filter coefficients. Using 2's-complement representation for numbers, a number x_k may be written

$$x_k = -x_k^0 + \sum_{j=1}^{B-1} x_k^j \, 2^{-j} \tag{2}$$

where B equals the number of bits in word x_k and the superscripts index the bits themselves with 0 being the most significant (sign) bit. Thus and x_k^j can only take values of 0 or 1. Substituting (2) into (1),

$$y_n = a_0\left(\sum_{j=1}^{B-1} x_n^j\, 2^{-j} - x_n^0\right) + a_1\left(\sum_{j=1}^{B-1} x_{n-1}^j 2^{-j} - x_{n-1}^0\right) + a_2\left(\sum_{j=1}^{B-1} x_{n-2}^j 2^{-j} - x_{n-2}^0\right)$$

$$-b_1\left(\sum_{j=1}^{B-1} y_{n-1}^j 2^{-j} - y_{n-1}^0\right) - b_2\left(\sum_{j=1}^{B-1} y_{n-2}^j 2^{-j} - y_{n-2}^0\right)$$

$$y_n = \sum_{j=1}^{B-1} 2^{-j}\left(a_0 x_n^j + a_1 x_{n-1}^j + a_2 x_{n-2}^j - b_1 y_{n-1}^j - b_2 y_{n-2}^j\right)$$

$$- \left(a_0 x_n^0 + a_1 x_{n-1}^0 + a_2 x_{n-2}^0 - b_1 y_{n-1}^0 - b_2 y_{n-2}^0\right)$$

$$y_n = \sum_{j=1}^{B-1} 2^{-j}\phi\left(x_n^j, x_{n-1}^j, x_{n-2}^j, y_{n-1}^j, y_{n-2}^j\right) - \phi\left(x_n^0, x_{n-1}^0, x_{n-2}^0, y_{n-1}^0, y_{n-2}^0\right) \qquad (3)$$

where $\phi(p,r,u,v,w) = a_0 p + a_1 r + a_2 u - b_1 v - b_2 w$

Peled and Liu's method is a direct implementation of (3). The basic block diagram is shown in Figure 3.4. Double lines indicate parallel paths for all B bits of a word; single lines indicate one bit paths. Beginning with Register A cleared, the least significant bits of x_n, x_{n-1}, x_{n-2}, y_{n-1}, and y_{n-2} comprise the address to the ROM which stores the function ϕ. This is added to the contents of Register A (zero), with the result then stored back in Register A. The x_n, x_{n-1}, x_{n-2}, y_{n-1}, and y_{n-2} registers are all shifted one place so that their second least significant bits form the address of the ROM. Applying ϕ again, the output of the ROM is added to the contents of Register A,

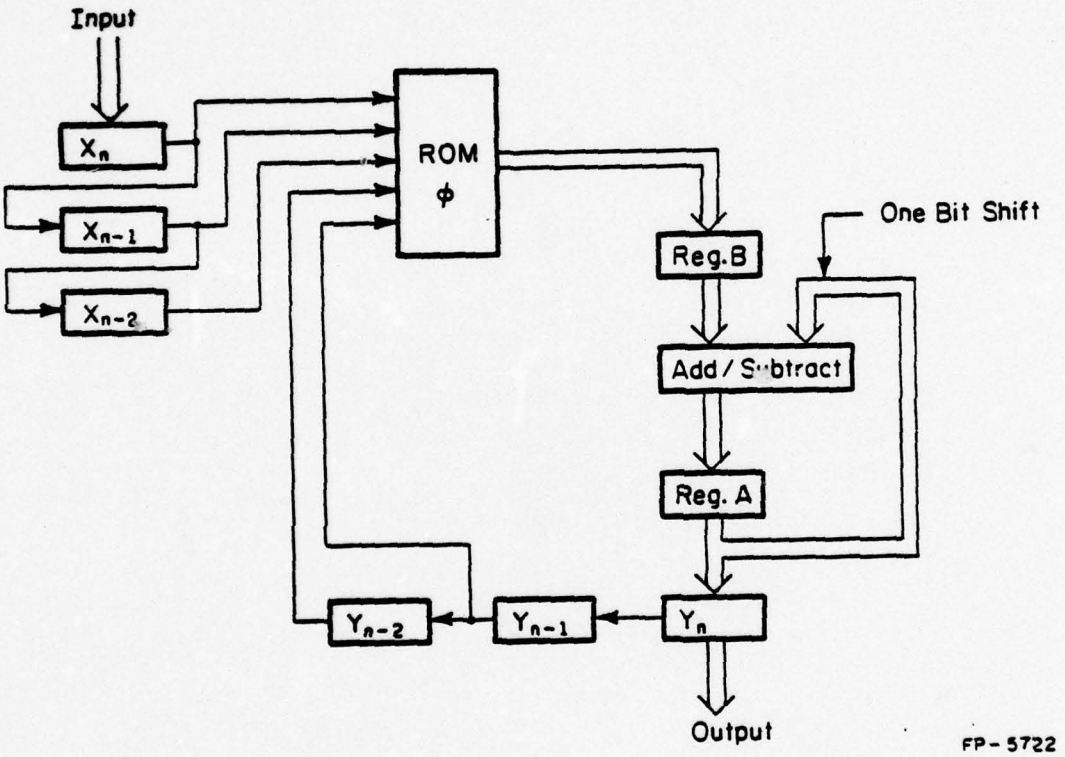


Figure 3.4. Block Diagram of Second-Order Section Implementation

except that a right shift is hardwired as the contents of Register A enter the adder. The shift corresponds to the 2^{-j} term in (3). This process continues with partial sums continuously stored in Register A until the most significant bit. At this point, the output of the ROM must be subtracted instead of added to the contents of Register A as required by (3). The new output and input words are parallel loaded into the y_n and x_n registers respectively, Register A is cleared, and the process repeats.

Register B appears between the ROM and the adder to increase the speed of the section. By storing the output of the ROM, the ROM look-up and the addition may be executed simultaneously.

For the Weaver modulator, the low pass filter must have a rather steep cutoff, in this case at 2 KHz. Generally about a ninth-order recursive filter is needed to obtain this specification [8]. Due to advantages in roundoff noise and coefficient length, and because Peled and Liu's implementation particularly lends itself to cascade form, an eighth-order filter of cascaded second-order sections was designed. An eighth-order filter proved adequate to meet noise specifications as shown in a later section.

It should be noted that in many instances it has been suggested that the low pass filter be divided into two filters, a slower recursive one and a faster transversal one, to reduce the amount of hardware by drastically reducing the number of multiplications necessary [2,7,9].

To improve accuracy and combat noise problems, all computations between the 8-bit input and 8-bit output are carried out with 16-bit words.

To conserve hardware, one second-order section can be time-multiplexed four times to effectively yield an eighth-order filter. Consider the following equations representing four cascaded second-order sections.

$$v_n = a_{10}x_n + a_{11}x_{n-1} + a_{12}x_{n-2} - b_{11}v_{n-1} - b_{12}v_{n-2} \quad (5)$$

$$w_n = a_{20}v_n + a_{21}v_{n-1} + a_{22}v_{n-2} - b_{21}w_{n-1} - b_{22}w_{n-2} \quad (6)$$

$$z_n = a_{30}w_n + a_{31}w_{n-1} + a_{32}w_{n-2} - b_{31}z_{n-1} - b_{32}z_{n-2} \quad (7)$$

$$y_n = a_{40}z_n + a_{41}z_{n-1} + a_{42}z_{n-2} - b_{41}y_{n-1} - b_{42}y_{n-2} \quad (8)$$

Expanding Peled and Liu's second-order section for this purpose requires a bank of registers and "switches" as shown in Figure 3.5. V_n, w_n, z_n, and y_n are computed consecutively by the second-order section, followed by v_{n+1}, w_{n+1}, etc. Each time a word is computed by the filter section, it is loaded into Register BUF1 from Register A. If v_n were computed, then while w_n is being computed, v_n is shifted from BUF1 into Register RV1 and the contents of RV1 into RV2. All other registers shift their contents back into themselves via the loops shown. Once w_n is calculated, during the calculation of z_n, w_n is shifted into RW1 and RW1 into RW2, while all other registers loop. The same applies for the calculation of y_n and v_n. In this way v_{n-1}, v_{n-2}, w_{n-1}, w_{n-2}, z_{n-1}, z_{n-2}, y_{n-1}, and y_{n-2} are stored so that they are available for filtering the next input word.

Since each section represented by (5)-(8) has different filter characteristics, a larger ROM is needed that can store four different ϕ's. Also, different words must be used to address the ROM depending upon which filter section is being implemented at a given time. In other words,

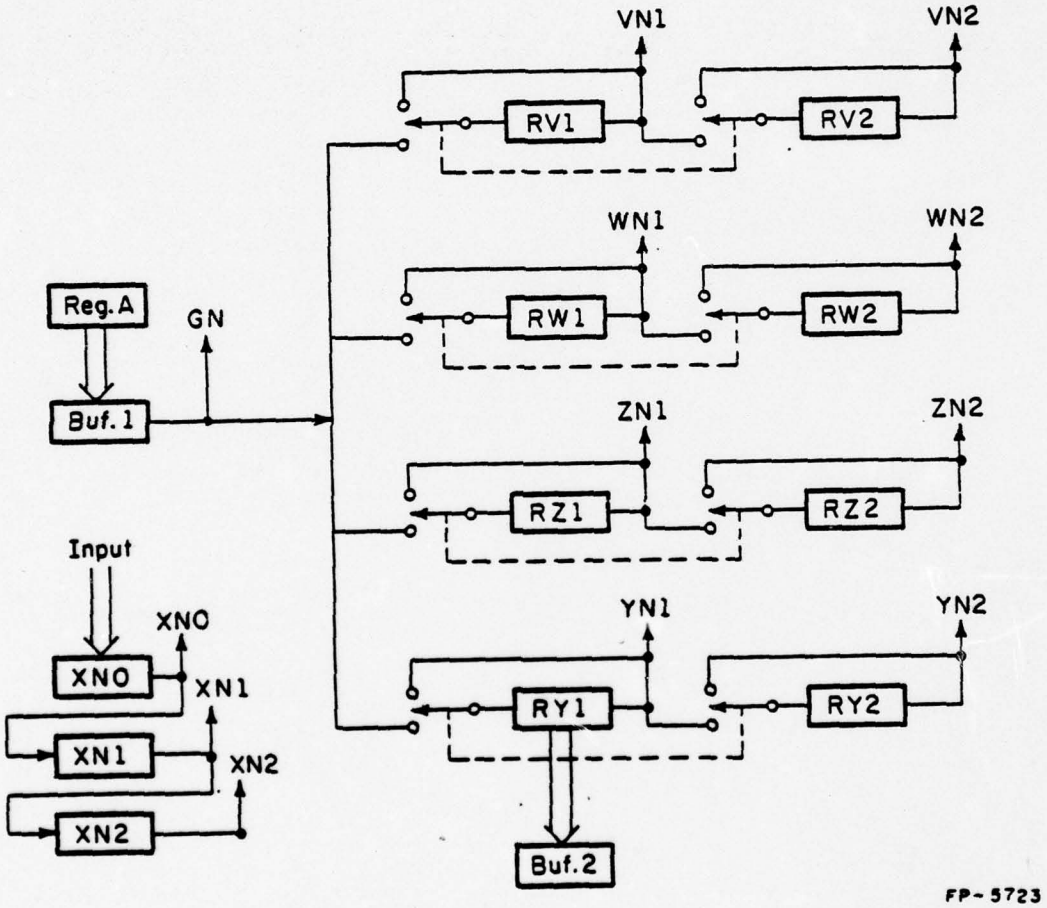


Figure 3.5. Register Bank for Four Cascaded Second-Order Sections

during the calculation of v_n, one bit from each of the following words is needed to address the ROM: x_n, x_{n-1}, x_{n-2}, v_{n-1}, v_{n-2}. This is easily seen from (5) and similarly for the other sections. These signals are labeled XN0, XN1, XN2, VN1, VN2, etc., with GN being either VN0, WN0, ZN0, or YN0, depending on which word was just calculated.

The register bank is detailed in Figure 3.6. All registers are implemented with 74LS166's, 8-bit shift registers that are serial-in, serial-out, except for RY1 which requires a parallel output to BUF2. RY1 is implemented with a 74LS164. The "switches" shown in Figure 3.5 are implemented with multiplexers, 74S157's. For each group, the multiplexers are controlled by signals labeled $\overline{\text{MUXV}}$, etc., which select the time the shift registers accept new data and the time they shift back into themselves. SC allows shifting to take place since during one clock pulse of every seventeen all registers must remain unchanged. This is explained later in the design of the second-order section. The registers RX0, RX1, and RX2 require special clocks since they do not shift back into themselves and thus shift less frequently than the rest. The 74LS164 also requires its own clock simply because it does not have the same control inputs as the 74LS166's. $\overline{\text{LDIO}}$ controls the loading of a new input word into RX0.

The addressing for the ROM is shown in Figure 3.7. Multiplexers, 74S153's, are used to choose which address lines reach the ROM, depending on which equation (5)-(8) is being implemented. This is accomplished by filter section controls FS0 and FS1. The ROM(s) used are Intel 3301A's, chosen for their tremendously fast access time [14]. They store the functions ϕ for each second-order section and their output is loaded into Register B, composed of 74S174's.

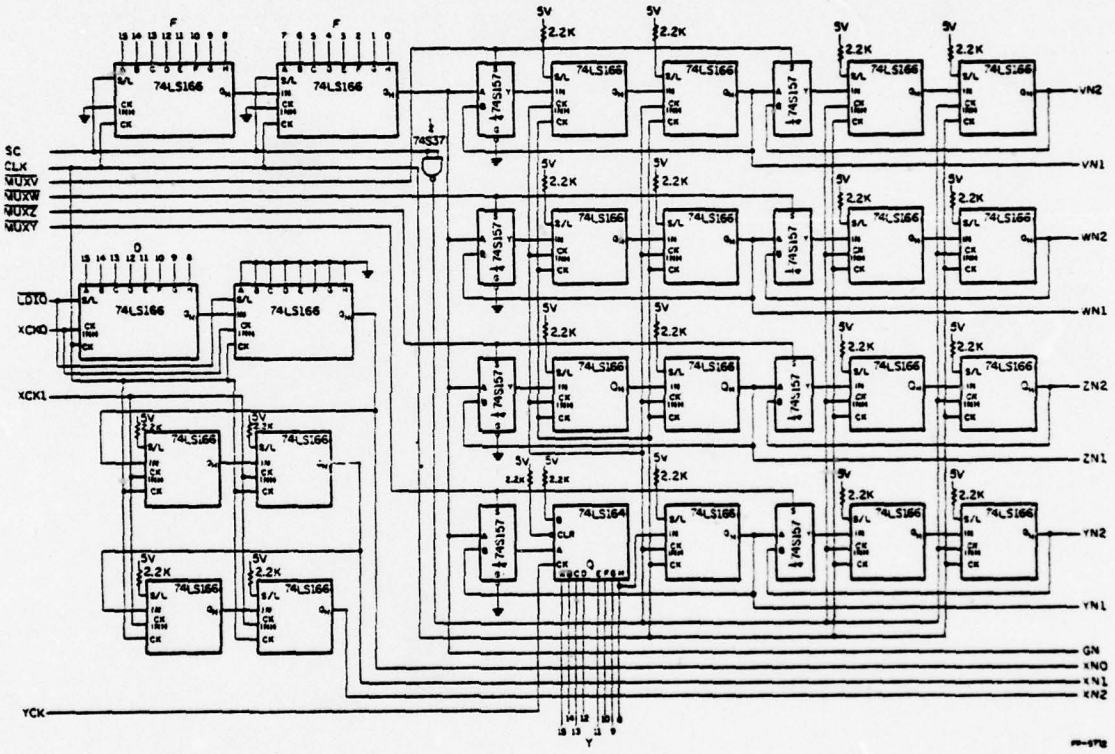


Figure 3.6. Schematic of Register Bank for Filter

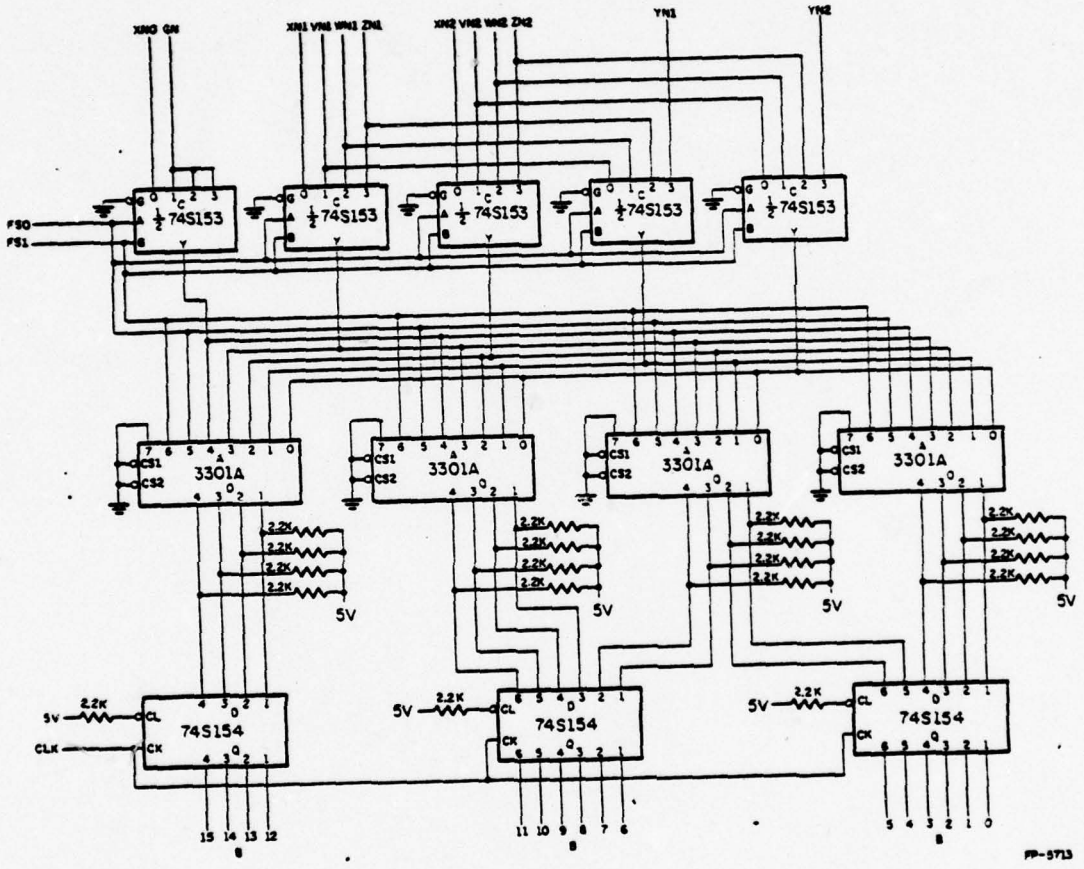


Figure 3.7. Schematic of ROM Addressing for Filter

Since ROM access time is 45 ns at most and 25 ns typically, the remainder of the second-order section which appears in Figure 3.8 constitutes the limiting factor in speed of the entire Weaver modulator. Therefore it was designed to operate as fast as possible. For the adder, 4-bit ALU's, 74S181's, were used since changing from addition to subtraction is simply a matter of changing select inputs. This is accomplished by the $\overline{A}/S$ control. The 74S182 provides one level of carry look-ahead to reduce the time of propagating carries. The network of exclusive-or gates provides the correct sign bit in the seventeenth position. That bit is needed since, as previously mentioned, the partial result is returned to one input of the adder with a hardwired right shift. With this design, using worst case propagation figures [10], an addition, that is, from the input of the ALU's to the input of Register A, can be computed in 40 ns, and a subtraction in 43 ns. Typical values are 26 ns for addition and 29 ns for subtraction. Taking into account worst case propagation and setup times in Registers A and B, an addition can occur every 62 ns; a subtraction takes 65 ns. If 74S114's are used instead of the 74S174's for Registers A and B, a 7 ns improvement in these times can be realized. However, an additional seventeen IC's would be used including 74S04's to create D flip-flops out of the 74S114's. This tradeoff was not deemed worthwhile. Leaving room for error, assume that the shortest clock period can be 70 ns. This is equivalent to a 14.29 MHz bit rate.

As previously stated, one addition (or subtraction) is needed for each bit of a word being filtered, in this case sixteen. However, to load BUF2 with the new output word and clear Register A to start the

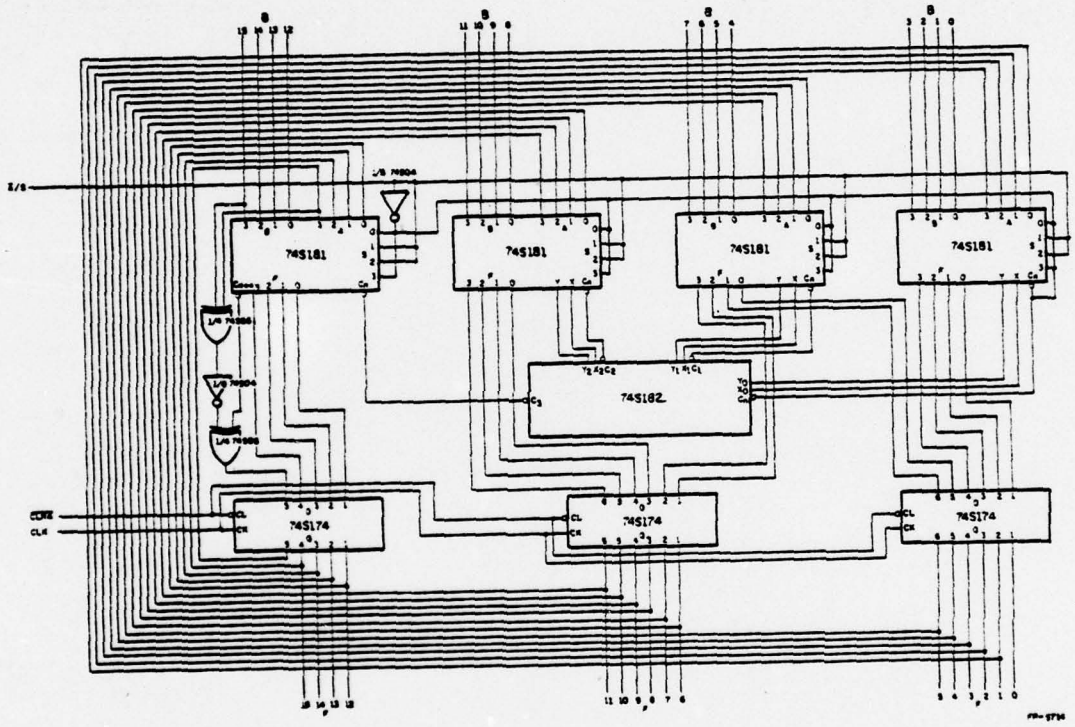


Figure 3.8. Schematic of Adders of Filter

next computation requires another clock pulse. Therefore seventeen clock pulses must occur for every word processed in the second-order section. Recall that for one path of a Weaver modulator the word rate will be 112 KHz. Seventeen clock pulses per word give a 1.904 MHz bit rate, but the second-order section must be time-multiplexed four times. The final result is a 7.616 MHz bit rate. To use the same hardware for both quadrature paths of the modulator would require a 15.23 MHz bit rate (as well as another bank of registers), and it has already been demonstrated that 14.29 MHz is the highest possible bit rate with this design.

Therefore, it is obvious that the hardware designed, even with the fastest possible filter section, can only accommodate one path of a Weaver modulator.

D. Low Pass Filter Characteristics

The low pass filter of the Weaver modulator must have a sharp cutoff at 2 KHz since the sample rate of 8 KHz causes the waveform from -2 KHz to 2 KHz to be repeated from 2 KHz to 6 KHz. Designing for .5 db ripple in the passband and 85 db down in the stopband meets the basic specifications for the A bank [8]. With the passband ending at approximately 1.4 KHz and the stopband beginning at 2 KHz, the normalized values become $\Omega_p = 1$, $\Omega_s = 1.41421$. From filter tables [11], an eighth-order elliptic filter meets the requirements with .27 db ripple in the passband and 85.35 db down in the stopband. The analog transfer function for the filter is

$$H_a(s) = 5.4026 \cdot 10^{-5} \frac{(s^2 + 2.055186473)(s^2 + 2.627791246)}{(s^2 + .0698902422s + 1.021987333)(s^2 + .2312565752s + .8276472029)}$$

$$\cdot \frac{(s^2 + 5.185791826)(s^2 + 38.24064783)}{(s^2 + .4308906176s + .4739384204)(s^2 + .5973440484s + .1490688473)}$$

Using a sample rate of 56 KHz and passband ending at 1.6 KHz, the digital cutoff frequency $\omega_c = .08976$. Pre-warping for the bilinear transformation gives the corresponding analog cutoff frequency $\Omega_c = \tan \frac{\omega_c}{2} = .04491$. Scaling the analog transfer function by this value gives

$$H_a'(s) = 5.4026 \cdot 10^{-5} \frac{(s^2 + .0041451056)(s^2 + .0052999922)}{(s^2 + .0031387708s + .0720612463)(s^2 + .0103857328s + .0016672816)}$$

$$\cdot \frac{(s^2 + .0104592235)(s^2 + .0771275626)}{(s^2 + .0193512976s + .0009558864)(s^2 + .0268267212s + .000300657)}$$

Applying the bilinear transformation $s = \frac{z-1}{z+1}$ yields the digital transfer function

$$H(z) = 5.521267848 \cdot 10^{-5} \frac{(z^2 - 1.983488021z + 1)}{(z^2 - 1.985552165z + .993754933)}$$

$$\cdot \frac{(z^2 - 1.978911785z + 1)(z^2 - 1.958596159z + 1)}{(z^2 - 1.972878361z + .9794759525)(z^2 - 1.95832025z + .9620677029)}$$

$$\cdot \frac{(z^2 - 1.713580582z + 1)}{(z^2 - 1.946592729z + .9477635945)}$$

The poles and zeros of this transfer function are listed in Table 3.1. A quick sketch of the positions of these poles and zeros reveals that pairing the first zeros with the first poles, the second zeros with the second poles, etc., will result in optimum pairing according

TABLE 3.1

Poles and Zeros of H(z)

Zeros

$.9917440105 \pm j.1282334498 = 1 \angle 7.367522042°$

$.9894558925 \pm j.1448345152 = 1 \angle 8.327696451°$

$.9792980795 \pm j.2024234954 = 1 \angle 11.67871434°$

$.856790291 \pm j.5156650049 = 1 \angle 31.0419152°$

Poles

$.9927760825 \pm j.0902805794 = .9968725761 \angle 5.196043534°$

$.9864391805 \pm j.0800855522 = .9896847743 \angle 4.641464429°$

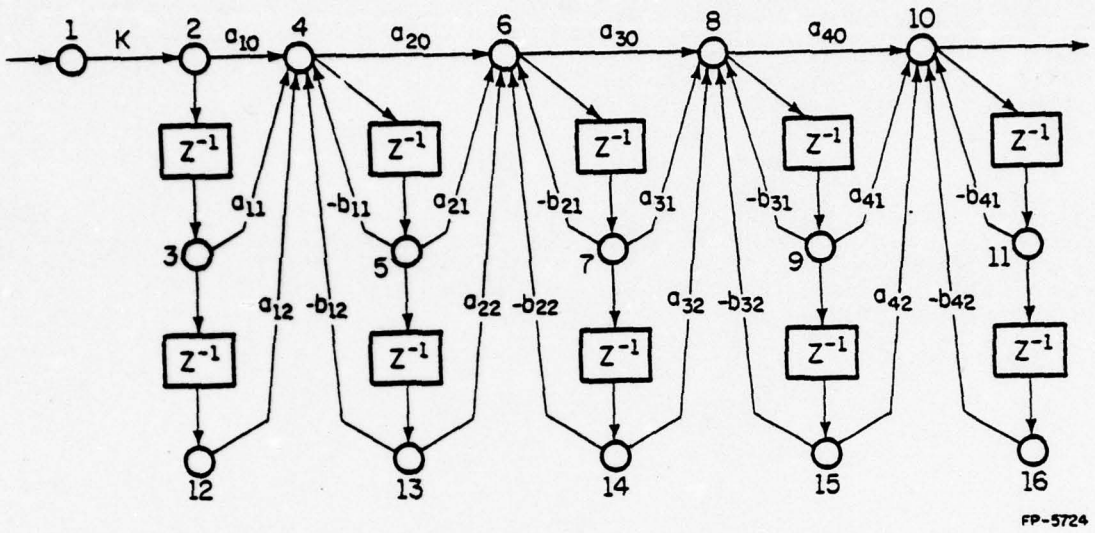
$.9791601285 \pm j.057559931 = .9808504998 \angle 3.364260864°$

$.9732963645 \pm j.0213958256 = .9735315067 \angle 1.259321558°$

to Jackson's criterion [12,13]. Jackson also discusses the ordering of the cascade sections to achieve minimum noise at the output of the filter, the noise being generated by roundoff in multipliers within the filter.

By using software developed at the Coordinated Science Laboratory (CSL), it was determined that for this case, the optimum order of cascade sections with respect to noise was from the "most peaked" to "least peaked" sections: that is, the section with highest Q is placed first. This is the order given in Table 3.1. Given the form of the filter on a node-to-node basis, the computer program not only determines noise at the output, but also provides proper scaling factors for each node to avoid overflow, and can calculate the sensitivity of any branch. Since Peled and Liu's realization of a second-order section is not standard, it can not be represented directly by a block diagram. However, with a slight modification it can still be analyzed with this program. Peled and Liu's system is actually a Direct I form without multipliers. The block diagram of the Direct I form used is shown in Figure 3.9. With this as input, the program calculates the scaling factors at each node to avoid overflow. With scaling, the final values of the coefficients are listed in Table 3.2. Using these values, the contents of the ROM are determined as explained in a previous section; the contents are listed in Tables 3.3 - 3.6. These 16-bit numbers have been rounded and are, of course, in 2's complement representation.

It is interesting to note that the magnitudes of the contents of the ROM vary from section to section. The first three sections have ROM entries all less than eight; the last section has all entries less than four. Thus the binary point occurs in a different place. Since these entries are the numbers actually summed in the second-order realization, the output of



$a_{11} = -1.983488021$ $\quad b_{11} = -1.985552165$ $\quad b_{12} = .993754933$

$a_{21} = -1.978911785$ $\quad b_{21} = -1.972878361$ $\quad b_{22} = .9794759525$

$a_{31} = -1.958596159$ $\quad b_{31} = -1.958320257$ $\quad b_{32} = .9620677029$

$a_{41} = -1.713580582$ $\quad b_{41} = -1.94659729$ $\quad b_{42} = .9477635945$

$a_{i0} = a_{i2} = 1, \quad i = 1,2,3,4$

$K = 5.521267848 \cdot 10^{-5}$

Figure 3.9. Direct I Form of Recursive Digital Filter

TABLE 3.2

Final Filter Coefficients After Scaling

$a_{10} = .3411364$	$a_{11} = -.6766399629$	$a_{12} = .3411364$	$-b_{11} = 1.985552165$	$-b_{12} = -.993754933$
$a_{20} = .1740882$	$a_{21} = -.3445051906$	$a_{22} = .1740882$	$-b_{21} = 1.972878361$	$-b_{22} = -.9794759525$
$a_{30} = .06367324$	$a_{31} = -.1247101633$	$a_{32} = .06367324$	$-b_{31} = 1.958320257$	$-b_{32} = -.9620677029$
$a_{40} = .01497823$	$a_{41} = -.025666404$	$a_{42} = .01497823$	$-b_{41} = 1.94659729$	$-b_{42} = -.9477635945$

K = 1

TABLE 3.3

Address		Contents			
00	00000	000.0	0000	0000	0000
00	00001	111.0	0000	0011	0011
00	00010	001.1	1111	1000	1010
00	00011	000.1	1111	1011	1101
00	00100	000.0	1010	1110	1011
00	00101	111.0	1011	0001	1110
00	00110	010.0	1010	0111	0100
00	00111	001.0	1010	1010	0111
00	01000	111.0	1010	0101	1001
00	01001	110.0	1010	1000	1100
00	01010	001.0	1001	1110	0011
00	01011	000.0	1010	0001	0110
00	01100	111.1	0101	0100	0100
00	01101	110.1	0101	0111	0111
00	01110	001.1	0100	1100	1101
00	01111	000.1	0101	0000	0000
00	10000	000.0	1010	1110	1011
00	10001	111.0	1011	0001	1110
00	10010	010.0	1010	0111	0100
00	10011	001.0	1010	1010	0111
00	10100	000.1	0101	1101	0101
00	10101	111.1	0110	0000	1000
00	10110	010.1	0101	0101	1111
00	10111	001.1	0101	1001	0010
00	11000	111.1	0101	0100	0100
00	11001	110.1	0101	0111	0111
00	11010	001.1	0100	1100	1101
00	11011	000.1	0101	0000	0000
00	11100	000.0	0000	0010	1110
00	11101	111.0	0000	0110	0001
00	11110	001.1	1111	1011	1000
00	11111	000.1	1111	1110	1011

TABLE 3.4

Address		Contents			
01	00000	000.0	0000	0000	0000
01	00001	111.0	0000	1010	1000
01	00010	001.1	1111	1000	0000
01	00011	000.1	1111	1100	1010
01	00100	000.0	0101	1001	0010
01	00101	111.0	0110	0011	1010
01	00110	010.0	0110	1011	0100
01	00111	001.0	0101	0101	1100
01	01000	111.1	0100	1111	1010
01	01001	110.1	0101	1010	0010
01	01010	001.1	0100	0001	1100
01	01011	000.1	0100	1100	0100
01	01100	111.1	1010	1000	1100
01	01101	110.1	1011	0011	0100
01	01110	001.1	1001	1010	1110
01	01111	000.1	1010	0101	0110
01	10000	000.0	0101	1001	0010
01	10001	111.0	0110	0011	1010
01	10010	010.0	0100	1011	0100
01	10011	001.0	0101	0101	1100
01	10100	000.0	1011	0010	0100
01	10101	111.0	1011	1100	1100
01	10110	010.0	1010	0100	0110
01	10111	001.0	1010	1110	1110
01	11000	111.1	1010	1000	1100
01	11001	110.1	1011	0011	0100
01	11010	001.1	1001	1010	1110
01	11011	000.1	1010	0101	0110
01	11100	000.0	0000	0001	1110
01	11101	111.0	0000	1100	0110
01	11110	001.1	1111	0100	0000
01	11111	000.1	1111	1110	1000

TABLE 3.5

Address		Contents			
10	00000	000.0	0000	0000	0000
10	00001	111.0	0001	0011	0111
10	00010	001.1	1110	1010	1011
10	00011	000.1	1111	1110	0001
10	00100	000.0	0010	0000	1010
10	00101	111.0	0011	0100	0000
10	00110	010.0	0000	1011	0100
10	00111	001.0	0001	1110	1011
10	01000	111.1	1100	0000	0010
10	01001	110.1	1101	0011	1001
10	01010	001.1	1010	1010	1101
10	01011	000.1	1011	1110	0100
10	01100	111.1	1110	0000	1100
10	01101	110.1	1111	0101	0011
10	01110	001.1	1100	1011	0111
10	01111	000.1	1101	1110	1101
10	10000	000.0	0010	0000	1010
10	10001	111.0	0011	0100	0000
10	10010	010.0	0000	1011	0100
10	10011	001.0	0001	1110	1011
10	10100	000.0	0100	0001	0011
10	10101	111.0	0101	0100	1010
10	10110	010.0	0010	1011	1110
10	10111	001.0	0011	1111	0101
10	11000	111.1	1110	0000	1100
10	11001	110.1	1111	0101	0011
10	11010	001.1	1100	1011	0111
10	11011	000.1	1101	1110	1101
10	11100	000.0	0000	0001	0110
10	11101	111.0	0001	0100	1100
10	11110	001.1	1110	1100	0001
10	11111	000.1	1111	1111	0111

TABLE 3.6

Address		Contents			
11	00000	00.00	0000	0000	0000
11	00001	11.00	0011	0101	1000
11	00010	01.11	1100	1001	0101
11	00011	00.11	1111	1110	1101
11	00100	00.00	0000	1111	0101
11	00101	11.00	0100	0100	1101
11	00110	01.11	1101	1000	1010
11	00111	01.00	0000	1110	0010
11	01000	11.11	1110	0101	1011
11	01001	11.00	0001	1011	0011
11	01010	01.11	1010	1111	0001
11	01011	00.11	1110	0100	1000
11	01100	11.11	1111	0101	0001
11	01101	11.00	0010	1010	1001
11	01110	01.11	1011	1110	0110
11	01111	00.11	1111	0011	1110
11	10000	00.00	0000	1111	0101
11	10001	11.00	0100	0100	1101
11	10010	01.11	1101	1000	1010
11	10011	01.00	0000	1110	0010
11	10100	00.00	0001	1110	1011
11	10101	11.00	0101	0100	0011
11	10110	01.11	1110	1000	0000
11	10111	01.00	0001	1101	1000
11	11000	11.11	1111	0101	0001
11	11001	11.00	0010	1010	1001
11	11010	01.11	1011	1110	0110
11	11011	00.11	1111	0011	1110
11	11100	00.00	0000	0100	0110
11	11101	11.00	0011	1001	1110
11	11110	01.11	1100	1101	1011
11	11111	01.00	0000	0011	0011

each section must have two or three bits to the left of the binary point, depending on which section one considers. It is necessary for the input word to have the same representation and it is important that it be in the same range so that the greatest number of significant bits may be used. Ordinarily the magnitude of the input word is less than or equal to one. In this case the input word should be less than or equal to eight for the first three sections, and less than or equal to four for the last section. This, however, requires no special attention in implementation since each section has unity gain so that no overflow can occur. If the system is considered to have multipliers and dividers as shown in Figure 3.10, no adjustment is necessary since the hardware does not "know" where the binary point should be.

Note that the first multiplication by eight does not lose significant bits since the input from the ADC is only eight bits and the system uses sixteen.

The program indicates that for this scaled system, variance of the noise at the output is approximately 51 db. However, recalling that this is based on the Direct I form, and the actual implementation requires no multipliers, the number of noise sources is decreased by a factor of five. This corresponds to a decrease of about 7 db resulting in 44 db noise. This in turn corresponds to roughly 7 bits of noise, which will be discarded anyway when only the eight most significant bits are retained at the output of the filter.

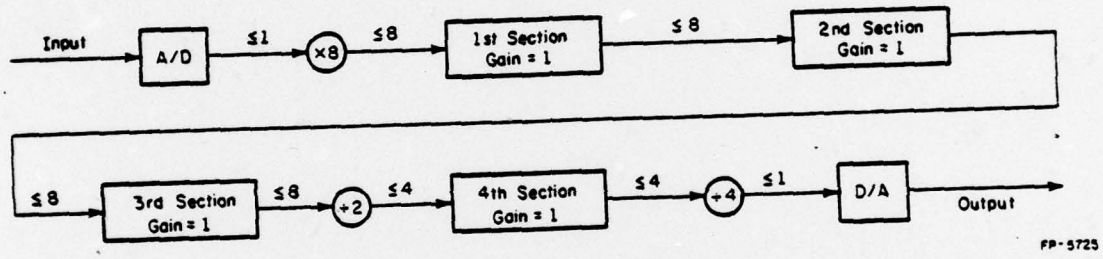


Figure 3.10. Block Diagram Indicating Binary Point Shifting

E. Modulation and Cancellation

The modulation of the baseband signal to a 4 KHz slot in the range of 60-108 KHz. and the subsequent cancellation of one of the sidebands are detailed in Figure 3.11. The output of the digital filter, labeled Y, is stored in a 74LS377 register corresponding to BUF2 in Figure 3.5. Loading of the register is controlled by $\overline{\text{LDIO}}$ which becomes active immediately following the calculation of y_n in the digital filter. The contents of this register address the modulation ROM, a 3304A-6, which executes multiplication in a direct look-up fashion. The ROM contents depend on two factors: in which quadrature path the ROM is located, and to which 4 KHz slot the signal is to be modulated.

The CS signal is a chip select which has not been generated in this design. Depending on overall system design this signal should only be active as long as required for the ROM to perform each modulation multiplication, since disabling the chip select greatly reduces the power consumption of the ROM.

The output of the modulation ROM from the cosine path added to that of the sine path cancels the lower sideband leaving only the upper sideband in the appropriate frequency slot. The addition is performed by two 4-bit adders, 74LS283's. The result is rounded by setting the carry in, and using the carry out as the sign bit gives correct results regardless of overflow. The output of the adder, then, is SSB and may be converted back to an analog signal for filtering and transmission or may be input to another level of frequency-multiplexing. Note that before this output may be passed through a DAC, the sample rate must be increased by a factor of two so that the signals in the range 60-108 KHz may be

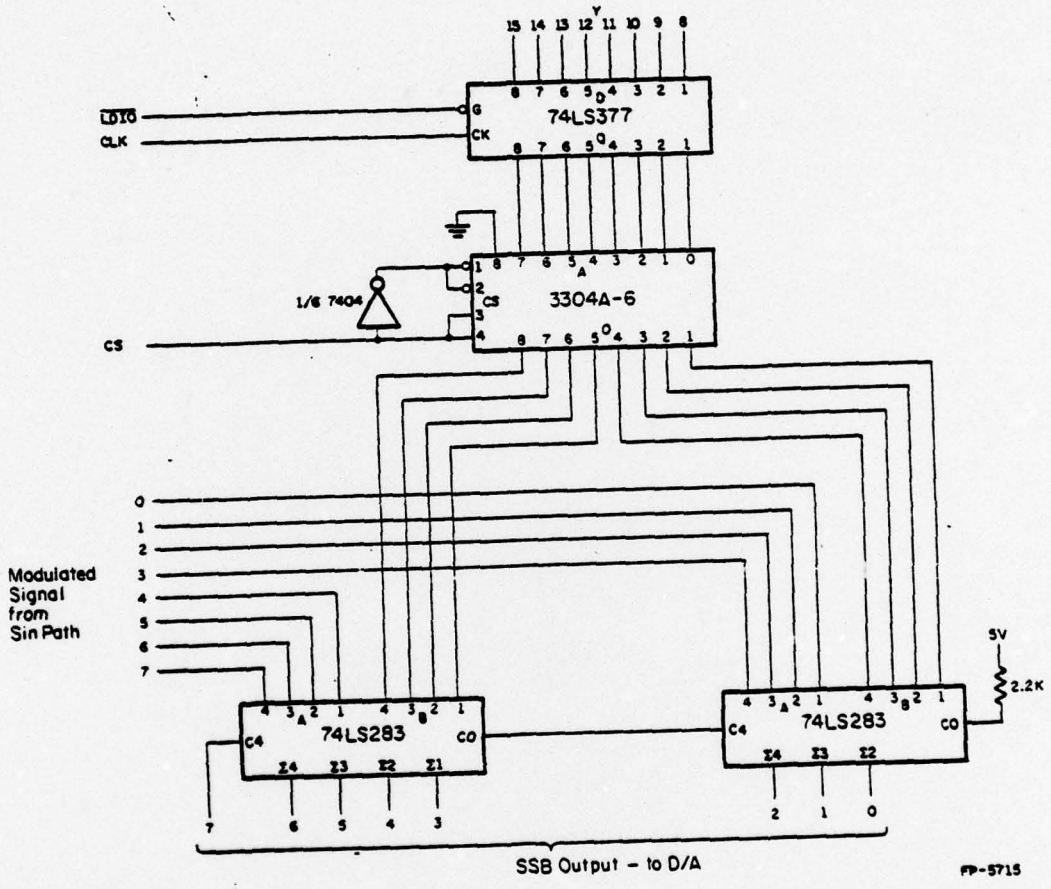


Figure 3.11. Schematic of Modulation and Cancellation

recovered. This would be most easily accomplished by inserting a zero between each output sample, and then passing the signal through a band-pass filter for the range 60-108 KHz.

In an effort to minimize hardware, only 8-bit words were used in the cancellation. This introduces one bit of noise in the least significant bit of the result, that would not be present had all sixteen bits been used in the cancellation. If more accuracy is needed, 16-bit arithmetic may be easily implemented by adding another register, another ROM, and two more adders. However, for an additional 3 db of noise, the tradeoff was considered worthwhile.

Assuming only 8-bit words are necessary for the cancellation, the input to the ROM can be only eight bits since multiplication by a sine or cosine cannot increase the size of the word. Therefore, only the eight most significant bits of y_n, the output of the filter, are used for the remainder of the Weaver modulator.

F. Control Signals

The generation of all control signals is shown in Figure 3.12. All are based on a system clock, CLK, the origin of which is not shown. It may be obtained from any convenient oscillator, a crystal, or a multivibrator. In any case, it should be buffered through a driver since it must drive approximately thirty TTL loads.

Since the sample rate is 8 KHz, there are thirteen zeros to be inserted between each sample, and sixty-eight clock pulses are needed for the filter for each word, the clock rate must be 7.616 MHz.

The timing of all control signals can be clearly seen in Figure 3.13. These pulses represent the ideal case: the pulses are rectangular

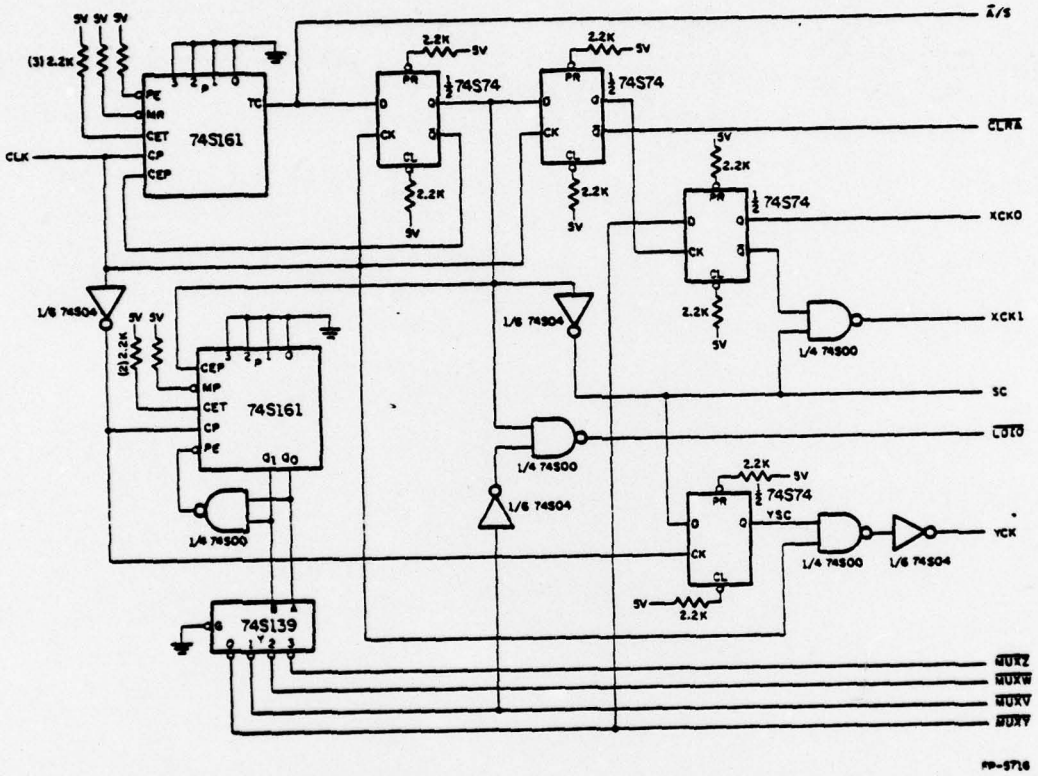


Figure 3.12. Schematic of Control Signals

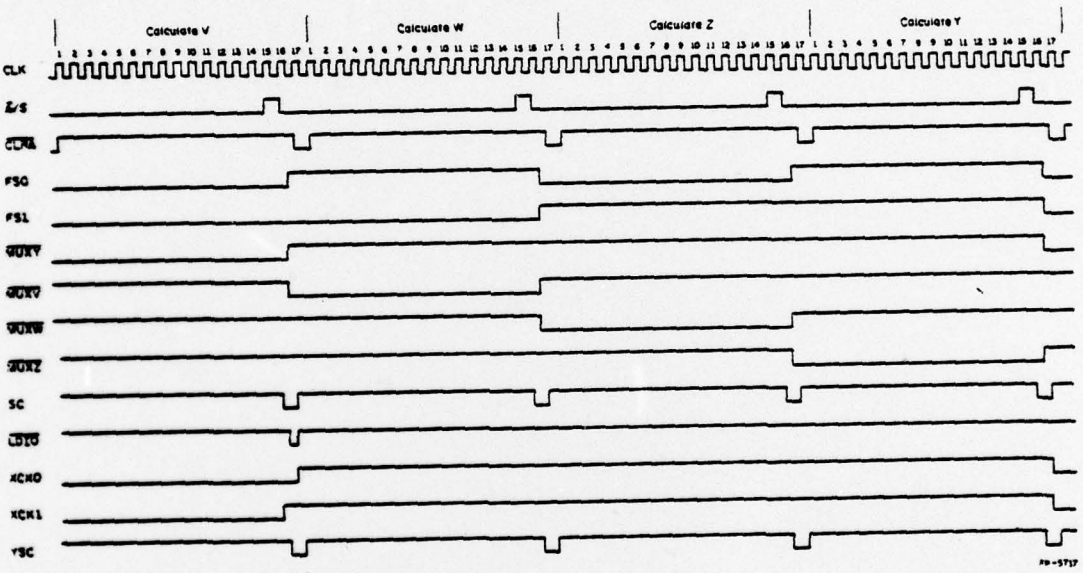


Figure 3.13. Timing Diagram of Control Signals

and no propagation times through IC's are taken into account in the drawing.

Each second-order section requires seventeen pulses, one for each bit of a word, plus one more to clear registers and output results. This timing is accomplished with a binary counter, 74S161, and a flip-flop, 74S74. The $\overline{A}/S$ determines whether the ALU's add or subtract. $\overline{CLRA}$ clears Register A. XCK0 and XCK1 are clocks controlling the shifting of the input registers of the filter. YCK is similarly a clock controlling one section of the output register. Shift control of all other registers in the filter is actuated by SC and all loading of input and output words is controlled by $\overline{LDIO}$. $\overline{MUXZ}$, $\overline{MUXW}$, $\overline{MUXV}$, and $\overline{MUXY}$ are the signals that switch the multiplexers in the storage section of the filter. These are derived from a decoder, 74S139, which takes its input from another 74S161 counter. This counter increases every seventeen clock pulses, denoting the beginning of another cycle through the second-order filter section. All components of the control section are Schottky types to minimize propagation delay in generating the control signals.

IV. CONCLUSIONS

The hardware design presented in this paper will accommodate only one path of a Weaver modulator for a single voiceband channel, due to limitations in speed. However, this implementation shows several advantages over other designs, primarily in chip count and accompanying power consumption. The number of IC's used excluding the ADC totalled sixty-seven, resulting in an approximate typical power consumption of 16.4 watts [10,14]. Therefore, to build a complete Weaver modulator for one voiceband channel, less than double the number of IC's given above would be required and less than double the power consumption cited would result, since parts of the design already include both quadrature paths.

Several modifications are possible to improve certain characteristics at the expense of others. For a simple increase in speed, a faster logic may be considered such as emitter-coupled logic (ECL). This would undoubtedly increase power consumption and most likely the chip count as well since a smaller variety of IC's would force a new design of the circuit. Of course the cost would be greater. A second method to increase speed is to use a completely parallel implementation of Peled and Liu's second-order section, or perhaps some combination of a series-parallel scheme [3]. This increases the bit rate of the second-order section, but requires many more ROM's and adders. Yet another possibility is actually to build four second-order sections in cascade eliminating the time-multiplexing and increasing the bit rate by four.

Assuming that the speed of the hardware could be increased significantly, it would be possible to time-multiplex both quadrature

paths through this system. Basically this would require another full bank of registers duplicating RV1, RV2, RW1, RW2, RZ1, RZ2, RY1, and RY2 so that the partial results needed to filter each subsequent word would be stored for each quadrature path. This involves expanding the multiplexing system and addressing of the ROM in the second-order section, as well as generating additional control signals similar to those already designed. The final modulation ROM would also have to be expanded to execute multiplication by either cosine or sine.

Further considerations include building and testing this design which was not done, to debug it and actually measure its performance. As an offshoot, it may prove interesting to study the possibilities of using a microprocessor and random-access memory (RAM) in place of the ROM in the second-order section. Where speed is not critical, this would allow the filter characteristics to be changed at will, providing an adaptive filter.

REFERENCES

1. Kurth, Carl F., "Generation of Single-Sideband Signals in Multiplex Communications Systems," IEEE Transactions on Circuits and Systems, Vol. CAS-23, No. 1, January 1976, pp. 1-17.

2. Freeny, Stanley L., et al., "Systems Analysis of a TDM-FDM Translator/Digital A-Type Channel Bank," IEEE Transactions on Communications Technology, Vol. COM-19, No. 6, December 1971, pp. 1050-1058.

3. Peled, Abraham and Bede Liu, "A New Hardware Realization of Digital Filters," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-22, No. 6, December 1974, pp. 456-462.

4. Peled, Abraham and Bede Liu, Digital Signal Processing Theory, Design, and Implementation, John Wiley and Sons, New York, 1976, pp. 218-247.

5. Weaver, D. K., "A Third Method of Generation and Detection of Single-Sideband Signals," Proceedings of the IRE, December 1956, pp. 1703-1705.

6. Kurth, Carl F., "SSB/FDM Utilizing TDM Digital Filters," IEEE Transactions on Communication Technology, Vol. COM-19, No. 1, February 1971, pp. 63-71.

7. Darlington, Sidney, "On Digital Single-Sideband Modulators," IEEE Transactions on Circuit Theory, Vol. CT-17, No. 3, August 1970, pp. 409-414.

8. Freeny, Stanley L., et al., "Design of Digital Filters for an All Digital Frequency Division Multiplex- Time Division Multiplex Translator," IEEE Transactions on Circuit Theory, Vol. CT-18, No. 6, November 1971, pp. 702-710.

9. Singh, S., et al., "Digital Single-Sideband Modulation," IEEE Transactions on Communications, Vol. COM-21, No. 3, March 1973, pp. 255-262.

10. The TTL Data Book for Design Engineers, Second Edition, Texas Instruments, Incorporated, 1976.

11. Christian, Erich and Egon Eisenmann, Filter Design Tables and Graphs, John Wiley and Sons, Inc., 1966, p. 201.

12. Jackson, L. B., "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," The Bell System Technical Journal, Vol. 49, No. 2, February 1970, pp. 159-184.

13. Jackson, L. B., "Roundoff-Noise Analysis for Fixed-Point Digital Filters Realized in Cascade or Parallel Form," IEEE Transactions on Audio and Electroacoustics, Vol. AU-18, No. 2, June 1970, pp. 107-122.

14. Intel Data Catalog 1975, Intel Corporation, 1975.

15. Liu, Bede and Abraham Peled, "Heuristic Optimization of the Cascade Realization of Fixed-Point Digital Filters," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-23, No. 5, October 1975, pp. 464-473.

16. Kurth, Carl F., "Analog and Digital Filtering in Multiplex Communications Systems," IEEE Transactions on Circuit Theory, Vol. CT-20, No. 4, July 1973, pp. 408-415.

17. Terrell, P. M. and P. J. W. Rayner, "A Digital Block-Processor for SSB-FDM Modulation and Demodulation," IEEE Transactions on Communications, Vol. COM-23, No. 2, February 1975, pp. 282-286.

18. Oppenheim, A. V. and R. W. Schafer, Digital Signal Processing, Prentice-Hall, Inc., New Jersey, 1975.