8.1A DECODING: CODES AND HARDWARE IMPLEMENTATION

M. P. Sulzer* and R. F. Woodman**

*Arecibo Observatory, Box 995, Arecibo, Puerto Rico
**Instituto Geofisico del Peru, Apartado 3747, Lima, Peru

INTRODUCTION

MST radars vary considerably from one installation to the next in the type of hardware, operating schedule, associated personnel, and amount of funding. Most such systems do not have the computing power to decode in software when the decoding must be performed for each received pulse, as is required for certain sets of phase codes. These sets allow one to obtain the best signal-to-sidelobe ratio when operating at the minimum baud length allowed by the bandwidth of the transmitter. We discuss here a number of realizations of the hardware phase decoder, and discuss the applicability of each to decoding MST radar signals. We present a new design for a decoder which is very inexpensive to build, easy to add to an existing system and is capable of decoding on each received pulse using codes with a baud length as short as one microsecond.

CODES USED WITH MST RADARS

In nearly all MST radar sounding, the received signal is stationary over a time equal to at least several periods of the transmitter pulse sequence. This fact has a strong influence on the type of codes which can be used. The most-used codes have been the binary complementary codes as described in WOODMAN (1980). Because these codes have zero sidelobes, there has been no need to use multiphase codes which are much more difficult to implement. Recently however, binary code sets have been developed which use a different code for each transmitted pulse (SULZER and WOODMAN, 1982; 1983). In a perfect system, these codes offer no advantage over the complementary pair, since the latter has no sidelobes. In a real system, however, there are imperfections, especially in the transmitter, and the quasi-complementary codes offer significant advantages. The use of these new codes has one large disadvantage: decoding must be performed on each received pulse. On the other hand, the use of the complementary code pair means that the returns from each of the two codes can be added before decoding so that only two decodings are necessary for each of the two codes coherent interpretation period. Thus the new codes require a lot more computation.

HARDWARE VERSUS SOFTWARE DECODING

The low computation rates associated with decoding the complementary pair when the coherent interpretation time is long allow the decoding to be performed in software in many systems. This is not the case if each pulse must be decoded. In this case a special purpose hardware device (DECODER) must be used. Decoders vary tremendously in their complexity, speed, the method used to perform the decoding and flexibility. As a consequence, they also vary tremendously in their cost.

The advantages of hardware decoding are the following:

(1) Much higher data rate than a general purpose computer, even higher than an array processor.

(2) Faster "programming" of code changes than in a computer. (Some types of decoders only.)

(3) On-line monitoring (at output of decoder) shows good height resolution.

The advantages of software decoding are the following:

(1) For low data rates, an already available general purpose computer may be used.

(2) Ease of implementation.

The next sections of this paper describe various types of hardware decodes, and present a new design for one which is well-suited for middle atmosphere work.

THE DECODER: A TRANSVERSAL FILTER

A decoder perform a convolution

$$o(t) = i(t) * h_j(t) \tag{1}$$

where $i(t)$ is the input signal
$h_j(t)$ is the impulse response of the decoder programmed for code j

and $o(t)$ is the output signal.

The convolution may be written out as an integral

$$o(t) = \int_0^t i(q)\, h(t-q)dq \tag{2}$$

This integral becomes a summation for discrete samples. Figure 1 shows a block diagram of a decoder for a code N long. The present value and N-1 earlier values are stored in a delay line. In the case of a binary phase code the impulse response consists of +1 or -1 and so the multiplications are replaced with additions of either the signal or the inverted signal. The figure shows a digital implementation. This could be done with an analog delay line and analog inverter and adders.

If we wish to change the code quickly, we must add a shift register and selector switches. This is shown in Figure 2. Note that if the signal values

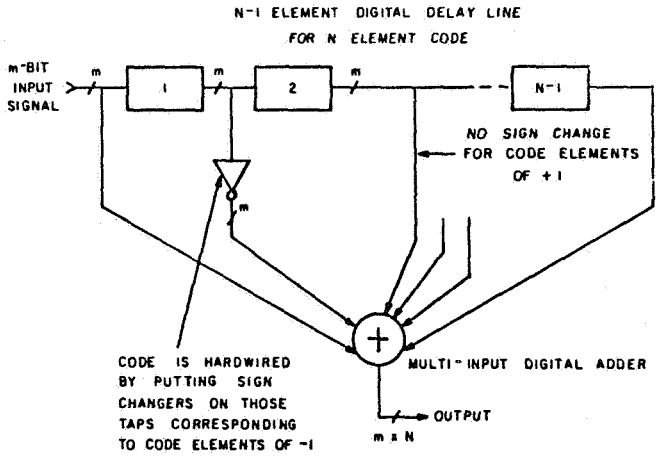


Figure 1. Transversal filter decoder for code length N.

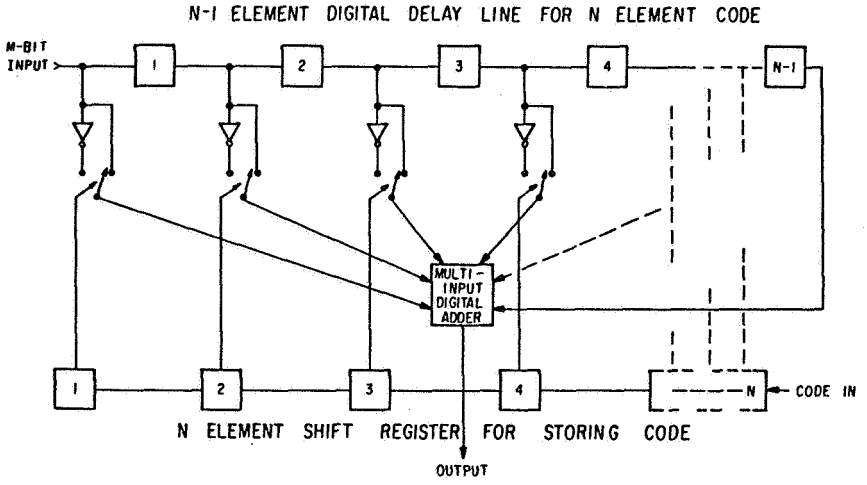


Figure 2. Transversal filter decoder for code length N allowing quick changes of code.

enter from the left, the code must fill the shift register from the right. This accomplishes the folding about the time axis which is indicated by the negative sign in equation 2.

For a practical decoder, we need one more shift register which is parallel to the one storing the phase code. This is the amplitude of the code and is '1' wherever the code is defined and '0' otherwise. It is also attached to the control lines on the switches and when it is '0' neither switch is connected. This makes it possible to use codes with the length less than N.

An example of the type of decoder described in Figure 1 is the Barker Decoder used at Arecibo Observatory. The device is completely analog and works at the 30 MHz IF frequency. The input signal is fed to a surface-wave delay line with taps. The appropriate taps are inverted and added. The device is good only for the 13 bit Barker code with 4 μsec baud.

The transversal filter decoder would probably not be built with a digital implementation because of the complexity of the multi-bit delay line. The next section describes a digital design which is much more efficient.

THE DECODER: A CORRELATOR

The convolution described in the last section can be performed in a very different way which is particularly useful for digital implementation and has the advantage that the number of bits in the code may exceed the number of storage resistors. This implementation has the form of a correlator, where each lag corresponds to a different decoded range. Hence, the number of ranges, rather than the number of bits in the code, is limited.

Referring to Figure 3, we see that the code enters a shift register which is as long as the number of ranges. At the output of each flip flop in the shift register is an adder (or subtractor) which adds into the associated accumulator if the flip flop state is '1' or subtracts if '0'. The signal to be added (or subtracted) is in all cases the present value, the one just digitized. If the code has N bits, then after N cycles, the ith accumulator will contain

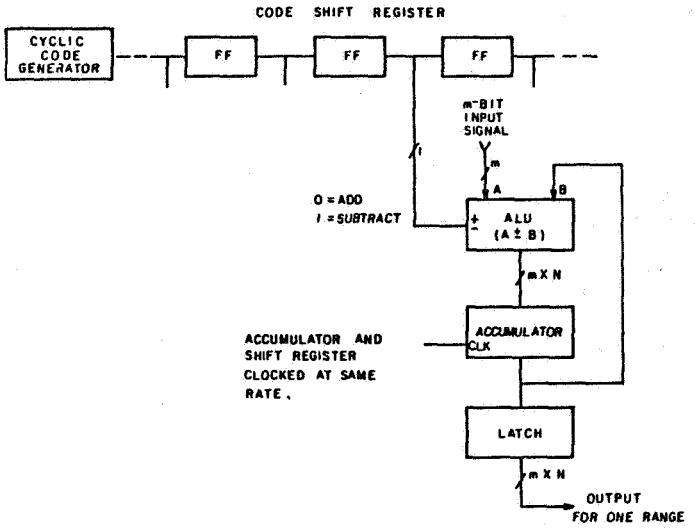


Figure 3. Decoder using correlator architecture uses one channel per range (one channel shown).

$$A_i = \sum_{n=0}^{n-1} (S_{i+1}) \cdot h_n$$

where S_i is the m-bit input signal
h_n is the code value (1 or -1)
A_i is the accumulated signal.

We can see that there is no limitation to the length of the code, in principle, although the accumulator will eventually overflow. On the otherhand, if the code is shorter than this number of registers, there is no range limitation either since each accumulator may be dumped to the computer as it completes, and the code may be started into the shift register again.

Figure 3 is a simplified block diagram of the planetary radar decoder at the Arecibo Observatory; this decoder has been used for MST observations in additon to its intended purpose. The actual circuit does not have an ALU for each range, but rather time shares one ALU among as many as 32 ranges, thus reducing the amount of hardware.

REQUIREMENTS OF AN MST RADAR DECODER

In the last section we have seen several ways to design a decoder. In designing a new decoder for MST work, the most important criterion for the instrument is that it fit into all, or nearly all, existing systems with a minimum of change. Since all, or nearly all such systems already have A/D converters and the ability to coherently add samples either in the general purpose computer for just before it, it is best not to interfere with this part of the system. This means that the model of the decoder as an analog filter is most appropriate. In this case, the decoder is placed after the baseband mixer, and before the A/D converter. Actually, the decoder could be digital inside; the important thing is that it be analog to analog and operate in real time. The requirement may seem unnecessary, but it should be remembered that the cost of any equipment must include the cost of altering the system of which it will be a part.

Other requirements are relatively simple. A length of 64 is adequate; we would like to use baud lengths of 1 μsec and keep errors better than 40 dB down if possible.

THE ANALOG RING DECODER

When designing a filter which will have an analog input and output, we must consider the possibility that using analog circuitry inside may be the best approach. High-speed digital processors are very powerful machines, but they have some disadvantages. The most important disadvantage for our purpose is difficulty in maintenance. These machines are usually one, or few, of a kind and the trouble-shooting procedures are often not well-defined.

We present here a new analog design which is similar to the normal transversal filter design in that the output is the simultaneous sum of N stored samples but is like the correlator design in that only the code, a simple three-level signal, need be shifted.

Figure 4 is a block diagram showing the concept of the analog ring decoder. To understand its operation first look at the rotor of the central rotary switch which is connected to the input signal. This switch deposits samples on the capacitors which are stored until the switch completes a revolution and deposits a new sample. As long as the code is shorter than the number of capacitors, we have enough information stored to decode the signal. All we have to do is add the signals from N adjacent capacitors with the correct polarity. To do this, we place the code in a shift register, and the outputs of the shift register control switches in the amplifiers to give the proper gain (+1, -1, or 0) to the signal. Everytime the switch moves forward one step, so does the shift register, and the correct signs are given to the delayed signals, which are added through the resistors to the output buss.

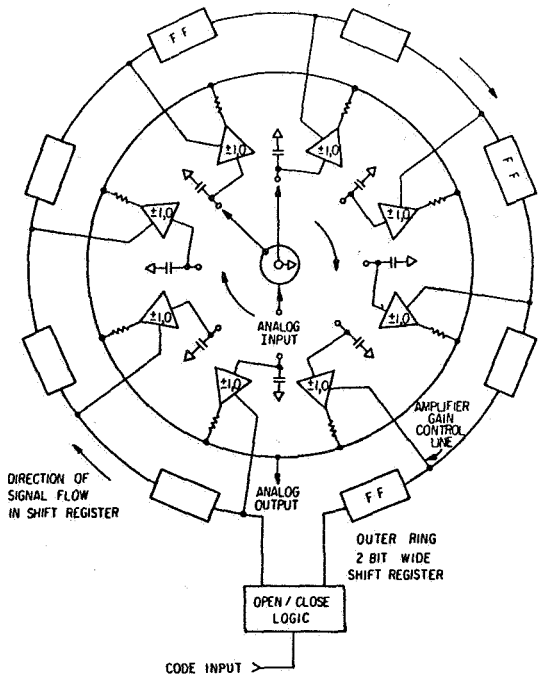
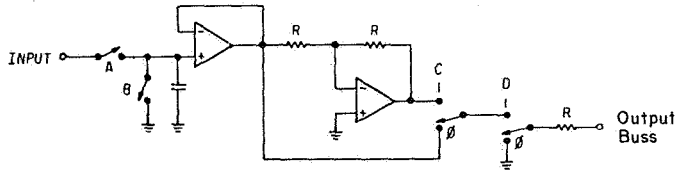


Figure 4. Block diagram of analog ring decoder.

This sample block diagram is very close to the way the actual machine operates. However, we can make a few simple additions which will allow the machine to run much faster with high accuracy. First let us make the number of capacitors and switch positions, M, somewhat larger than N, the length of the largest code we will use. Second, we arrange the phase of the switch rotor so that the code begins two positions after the switch pointer. This means that the present sample and the ones surrounding it will be connected to amplifiers with gain of 0. Third we add a second rotor to the switch which runs one step ahead of the other. This rotor is connected to the ground.

Figure 5 shows one section of the ring. Swintches A and B are each part of a CMOS analog multiplexer which acts like the rotary switch of Figure 4. The first step in taking a sample is to discharge the capacitor; the second step is to connect it to the input line so that a sample may be taken. This latter step also removes the previous signal, but it turns out that it is more important to discharge as nearly completely as possible the previous signal, than it is to fully charge it with the new signal. This is because keeping an old signal will cause a "ghost" and there is no way around this problem except to discharge the capacitor. On the other hand, if all of the capacitors and switch resistances had the same values it would not matter how many time constants of settling time were allowed since the relative values of the samples would be unaffected. Therefore, it is only the differences between these component values which set the necessary settling time for sampling. Allowing 2.5 time constants ($R = R_{switch} + R_{amp}$) and a maximum component value error of 10% give a maximum error of .8% in voltage, which is more than 40 dB down. Using an additional cycle to discharge the capacitor ($R = R_{switch}$ only) puts the ghost down over 80 dB. To achieve the same ghost level without the extra cycle would slow down the machine by more than a factor of two.

The two operational amplifiers and switches C and D in Figure 5 are equivalent to the gain-switched amplifier of Figure 4. After the sample is completed in step two, we wait one more clock cycle for the two amplifiers to settle. Once the outputs of the two amplifiers have settled, they will not change for M-3 clock cycles. The gain charge is "passive" in that it is achieved by switching between two unchanging signals. The addition of the various delayed signals is also passive in that it invokes only resistors and no operational amplifier. The output does contain "glitches" due to the finite switching time (∿50 nsecs) and pick up from the logic circuitry. Since all of the logic is synchronous, all the glitches occur on clock transitions and may be removed by using a sample and hold on the output which samples just before the clock transition.



	SWITCH				
STEP	A	B	C	D	PURPOSE
1	0	C	X	Ø	CLEAR CAP.
2	C	0	X	Ø	SAMPLE
3	0	0	X	Ø	SETTLE
4-N	0	0	Ø or 1	1	ADD / SUB

X = DON'T CARE
Ø or 1 DEPENDS UPON PHASE OF CODE

Figure 5. Sampling and switching.

A complete error analysis is not possible in a paper of this length, but a few comments will give a general idea of the sensitivity to errors. Assume that there will be a relative error in each of the N slices of the decoder given by ΔE_i for i = 1,N, where by relative error we mean the error in the sampled signal divided by the signal. When the N samples are added the N signals add directly but the N error add as random numbers since they are independent and so the relative error in the output signal voltage will be $\sqrt{N}$ times smaller than the typical relative error on a single sample or N times better in power. Furthermore, the error in the output signal is a function of the code; that is, some codes will give a positive error, others negative. Thus by using the Quasi-complementary codes which have a different code for each member of the sequence, we obtain a further reduction in the error in the decoder in the same manner errors in the transmitter are reduced.

REFERENCES

Sulzer, M. P. and R. F. Woodman (1982), Quasi-complementary codes, presented at the Workshop on Middle Atmosphere Measurements and Middle Atmosphere Radars (Cosponsored by SCOSTEP, COSPAR, IAGA, IAMAP, VRSI), May 10-12, Estes Park, Colorado.

Sulzer, M. P. and R. F. Woodman (1983), Quasi-complementary codes: A new technique for MST radar sounding, accepted for publication by Radio Science.

Woodman, R. F. (1980), High-altitude resolution stratospheric measurements with the Arecibo 430-MHz radar, Radio Sci., 15, 417-422.