Contractor Report Number: CR180822

ONBOARD MULTICHANNEL DEMULTIPLEXER/ DEMODULATOR

NASA CONTRACT NAS3-24885

29 JULY 1987

FINAL REPORT

DR. S. JOSEPH CAMPANELLA

DR. SOHEIL SAYEGH

COMSAT LABORATORIES

CLARKSBURG, MARYLAND

(NASA-CR-180822) ONBOARD MULTICHANNEL DEMULTIPLEXER/DEMODULATOR Final Report (Communications Satellite Corp.) 113 p Avail: NTIS HC A06/MF A01 CSCL 17B

N87-28819

Unclas

G3/32 0098642

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. CR180821	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle OnBoard Multichannel Demultiplexer/Demodulator		5. Report Date July 1987
		6. Performing Organization Code 650-20-26
7. Author(s) Dr. S. Joseph Campanella Dr. Soheil Sayegh		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address COMSAT 22300 Comsat Drive Clarksburg, MD 20871		11. Contract or Grant No. NAS3-24885
		13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address NASA-LEWIS Research Center 21000 Brookpark Road Cleveland, OH 44135		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

see attached

17. Key Words (Suggested by Author(s))	18. Distribution Statement General Release

19. Security Classif. (of this report) unclassified	20. Security Classif. (of this page)	21. No of pages	22. Price*

NASA FORM 1626 OCT 86 *For sale by the National Technical Information Service, Springfield, Virginia 22161

ABSTRACT

ONBOARD MULTICHANNEL DEMULTIPLEXER/DEMODULATOR STUDY

NAS3-24885

S. JOSEPH CAMPANELLA
AND
SOHIEL SAYEGH

COMSAT LABORATORIES
CLARKSBURG, MARYLAND

An investigation, performed for NASA LeRC by COMSAT LABS , of a digitally implemented on-board demultiplexer/demodulator able to process a mix of uplink carriers of differing bandwidths and center frequencies and programmable in orbit to accommodate variations in traffic flow is reported. The processor accepts high speed samples of the signal carried in a wideband satellite transponder channel, processes these as a composite to determine the signal spectrum, filters the result into individual channels that carry modulated carriers and demodulates these to recover their digital baseband content. The processor is implemented by using forward and inverse pipeline Fast Fourier Transformation techniques. The recovered carriers are then demodulated using a single digitally implemented demodulator that processes all of the modulated carriers. The effort has determined the feasibility of the concept with multiple TDMA carriers, identified critical path technologies, and assessed the potential of developing these technologies to a level capable of supporting a practical, cost effective on-board implementation. The approach is referred to as a flexible, high speed, digitally implemented Fast Fourier Transform (FFT) bulk demultiplexer/demodulator.

ONBOARD MULTICHANNEL DEMULTIPLEXER/DEMODULATOR STUDY FINAL REPORT

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NASA CONTRACT NAS3-24885
ON-BOARD MULTICHANNEL DEMULTIPLEXER/DEMODULATOR STUDY
FINAL REPORT

1.0 INTRODUCTION

The purpose of this study is to conduct an investigation of an on-board demultiplexer/demodulator concept, determine its feasibility with TDMA in a multifrequency environment, identify critical path technologies, and assess the potential of developing these technologies to a level capable of supporting a practical, cost effective on-board implementation. The approach is to incorporate a flexible, high speed, digitally implemented Fast Fourier Transform (FFT) demultiplexer/demodulator.

A functional diagram of a complete on-board baseband processor is shown in Figure 1.1. The portion of this processor considered for digital implementation by this study is outlined in the dashed box. Such digital implementation provides flexibility that permits the onboard processor to accommodate different types of multichannel FDMA of TDMA/FDMA digital service simply by changing its computation rules and organization. This can be done from the ground by sending to the onboard processor new programing instructions that for example permit one wideband processor to demultiplex and demodulate hundreds of narrow bandwidth digital carrier channels while another is doing the same thing with tens of wide bandwidth digital carrier channels and yet another is doing it with a mix of wide and narrowband carrier channels. Of course the rules and organization can easily be changed to accommodate variations in the service over the lifetime of the satellite or to accommodate different applications of the same type of satellite in different locations around the earth. This flexibility is the central piece of the concept.

The objective of the study is to determine the details of digital implementation of the demultiplexer and the demodulators and to assess the feasibility of constructing such processors in the future. In this respect an important part of the effort is a review of the advances that can be expected to occur in the important digital component areas in terms of size, power, weight, speed and radiation resistivity of the digital logic and memory components from which the processor is to be fabricated. Also critical technology areas into which R and D should be expended to achieve efficient and practical onboard implementation are identified.

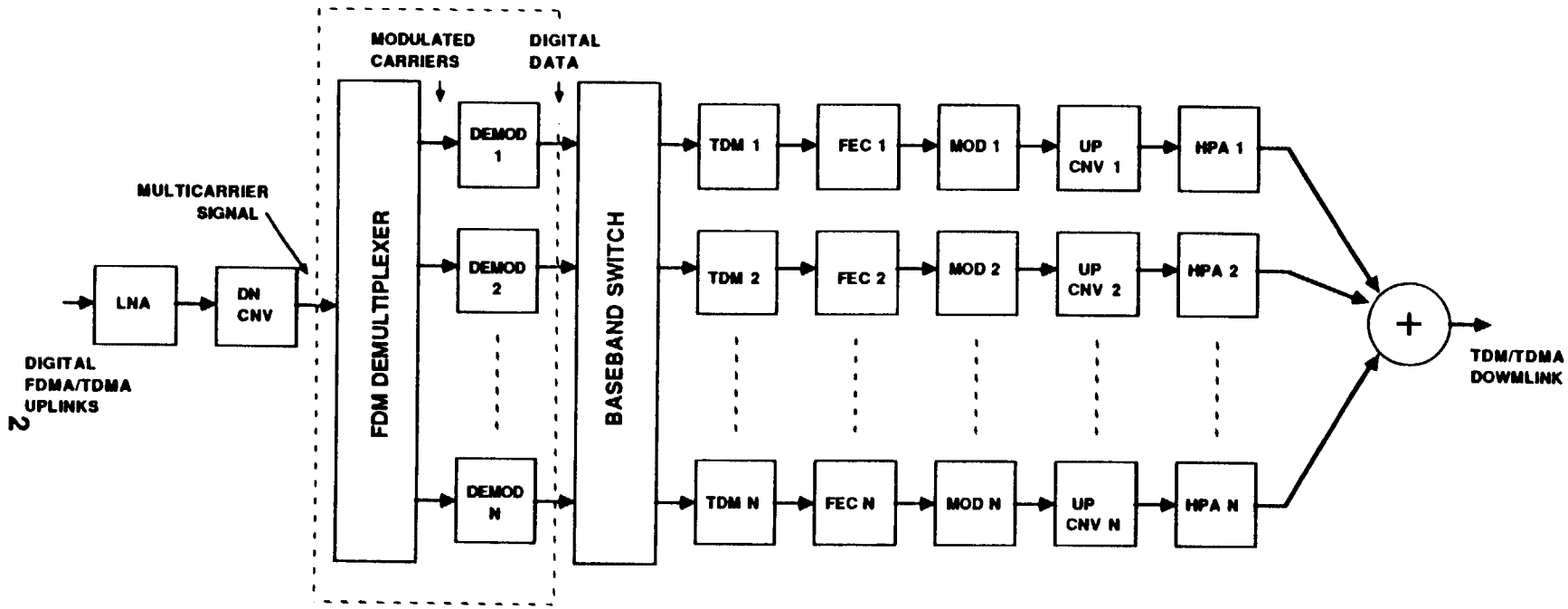


FIGURE 1.1. DIGITAL ONBOARD PROCESSING FUNCTIONAL CONFIGURATION

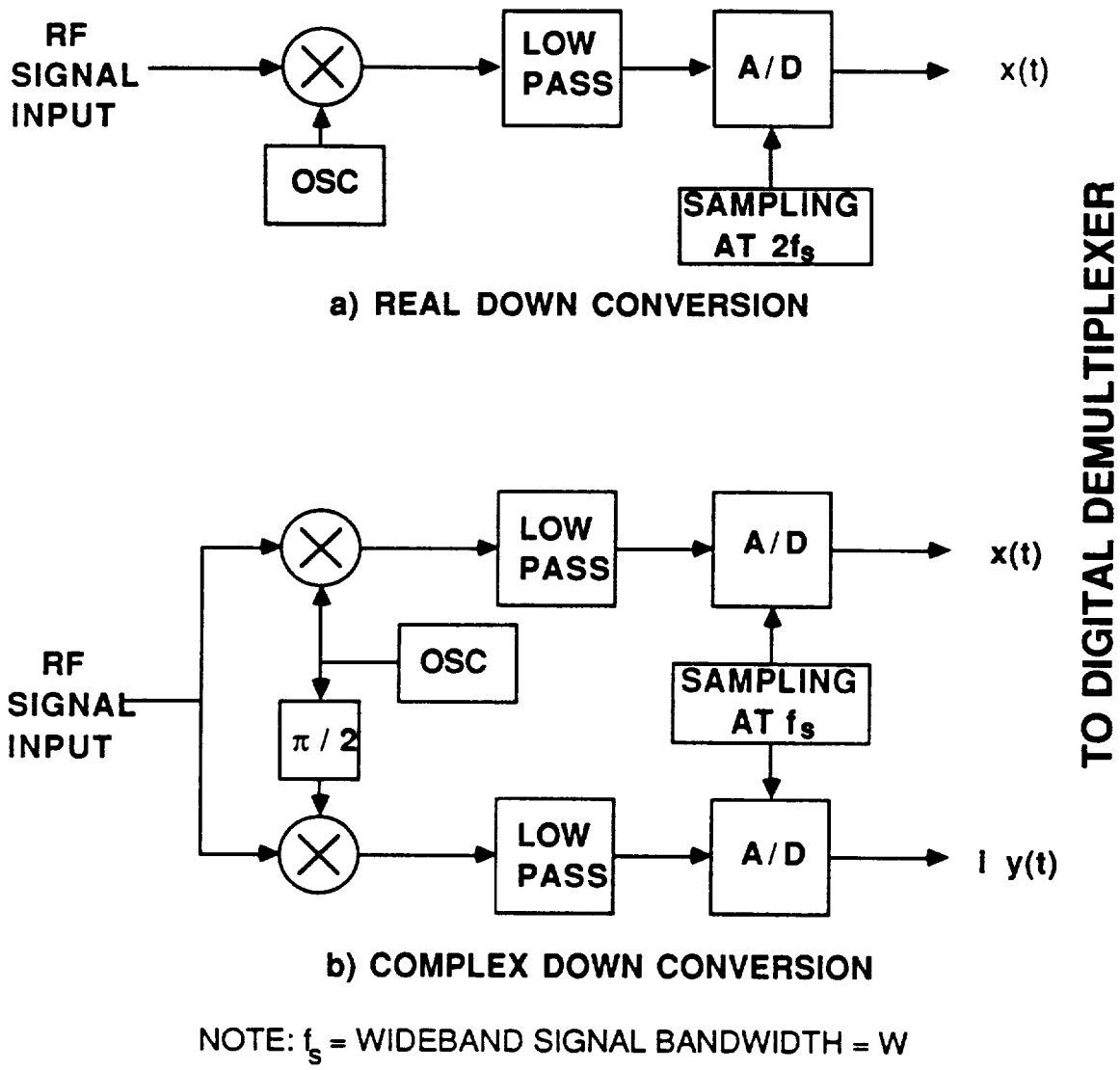
The processor is envisioned as operating in wideband channels of fixed bandwidth similar to that of the transponder channels used in the existing satellites. The wideband channel input signals which occur at their assigned RF carrier frequency at the front end are down converted so that their carrier frequency is at zero Hz at the input to the Demultiplexer. A multiplicity of such wideband channels would occupy the spectrum assigned to the service. For the purpose of this study, a wideband channel bandwidth of 40 MHz has been chosen because it is typical of transponder's used in todays satellite systems. The wideband channel signal can be sampled in either real or complex form as illustrated in Figure 1.2. For real sampling, the channel is sampled at twice the wideband channel bandwidth as shown in part (a) of Figure1.2. For complex sampling, the signal is divided into direct and quadrature paths as shown in part (b) of Figure 1.2. In this case the channel sampling rate is equal to the channel bandwidth, i. e. 40 Msamp/s. Because complex sampling operates at a lower sampling it is easier to implement. Also it is inherently more suited to the FFT processing structures that are used extensively in this investigation. If the technology permits it, extension to higher sampling rates and consequently higher channel bandwidths is obvious. For example, processing using channel bandwidths of 80 MHz can be expected in the future.

The down converted baseband is processed by a forward FFT to determine the spectrum distribution within the wideband channel in terms of discrete Fourier coefficients . Next, these coefficients are processed by a digital filter to select a particular channel and the resulting coefficients processed by a demodulator processor to recover the bits of the digital signal. Details of the arithmetic and its implementation constitute a large portion of the report that follows. The demodulation is described in detail for QPSK modulation and extensions for accommodating other modulation formats such as offset QPSK and 8-PSK are indicated.

The report is divided into five sections each covering a major area of concern. These are:

SECTION 2.0 - DEMULTIPLEXER IMPLEMENTATION.

This section presents the most efficient architecture for the implementation of the FFT algorithm and determines the size of the FFT that will be sufficient to meet the needs of the demultiplexing processing.



NOTE: f_s = WIDEBAND SIGNAL BANDWIDTH = W

FIGURE 1.2. DOWN CONVERSION, REAL AND COMPLEX

SECTION 3.0 - RECOVERY OF THE TIME DOMAIN SAMPLES OF SELECTED CHANNELS

This section develops the rules for realizing the filters that suitably separate the communications carriers into their required bandwidths. These filters must be flexibly programmable to accommodate a wide variation in the number of carriers and their bandwidths. The output must be samples in the time domain that are suitable for the demodulation processing that follows.

SECTION 4.0 - DIGITAL DEMODULATOR .

This section presents the demodulator architecture for extracting the baseband digital information from the filtered carriers. This requires processing to recover the carrier frequency and phase, the clock frequency and phase and the information.

SECTION 5.0 - TECHNOLOGY SURVEY.

Based on the detailed processing architecture and requirements identified, the current technology has been reviewed from the point of view of its ability to meet the need and new technology requiring additional development has been identified. In particular the developments from the VHSIC program are included.

SECTION 6.0 - RECOMMENDED DEVELOPMENT PROGRAM.

Long term development requirements needed to fabricate space flight qualified operational hardware are identified. This identifies areas where future NASA sponsored research and development can be directed to realize a practical cost effective implementation.

2.0 DEMULTIPLEXER IMPLEMENTATION

2.1 DEMULTIPLEXER IMPLEMENTATION WITH A PIPELINE FFT AND AN IDFT

2.1.1 GENERAL

The demultiplexer comprises a forward FFT implemented using a pipeline architecture which decomposes the input wideband spectrum into discrete Fourier coefficients followed by a channel filter that selects those coefficients that are in the wanted channel and an IDFT that reconstructs the time domain samples from the filtered coefficients. This arrangement proves suitable for demultiplexing multiple carriers when they all have the same bandwidth, but it consumes too much power for demultiplexing carriers of mixed bandwidths. In the section that follows this one, it is shown that use of an IFFT followed by an interpolating filter for the reconstruction of the time domain samples greatly reduces the computational intensity and power required by the sample reconstruction process when mixed carrier bandwidths are involved.

A pipeline architecture is selected as the most efficient way to implement the conversion of the wideband input signal into the discrete spectrum coefficients needed to demultiplex individual carrier channels. It can readily be implemented in hardware which can be operated under microprocessor stored program control to make adjustments to change the composition of multiple carrier channels demultiplexed.

The FFT pipeline architecture shown in Figure 2.1 has a number of important advantages for the implementation of the onboard demultiplexer. These are:

1) Its pipeline architecture is suited to high speed operation because it inherently distributes the processing among many separate processing functions.

2) In contrast with a parallel architecture which may also be able to operate at high speed, it requires far less memory (2 to 3 times less).

3) It yields a compact structure, i.e. one that does not have an excessive number of branches and is therefore well suited for hardware implementation.

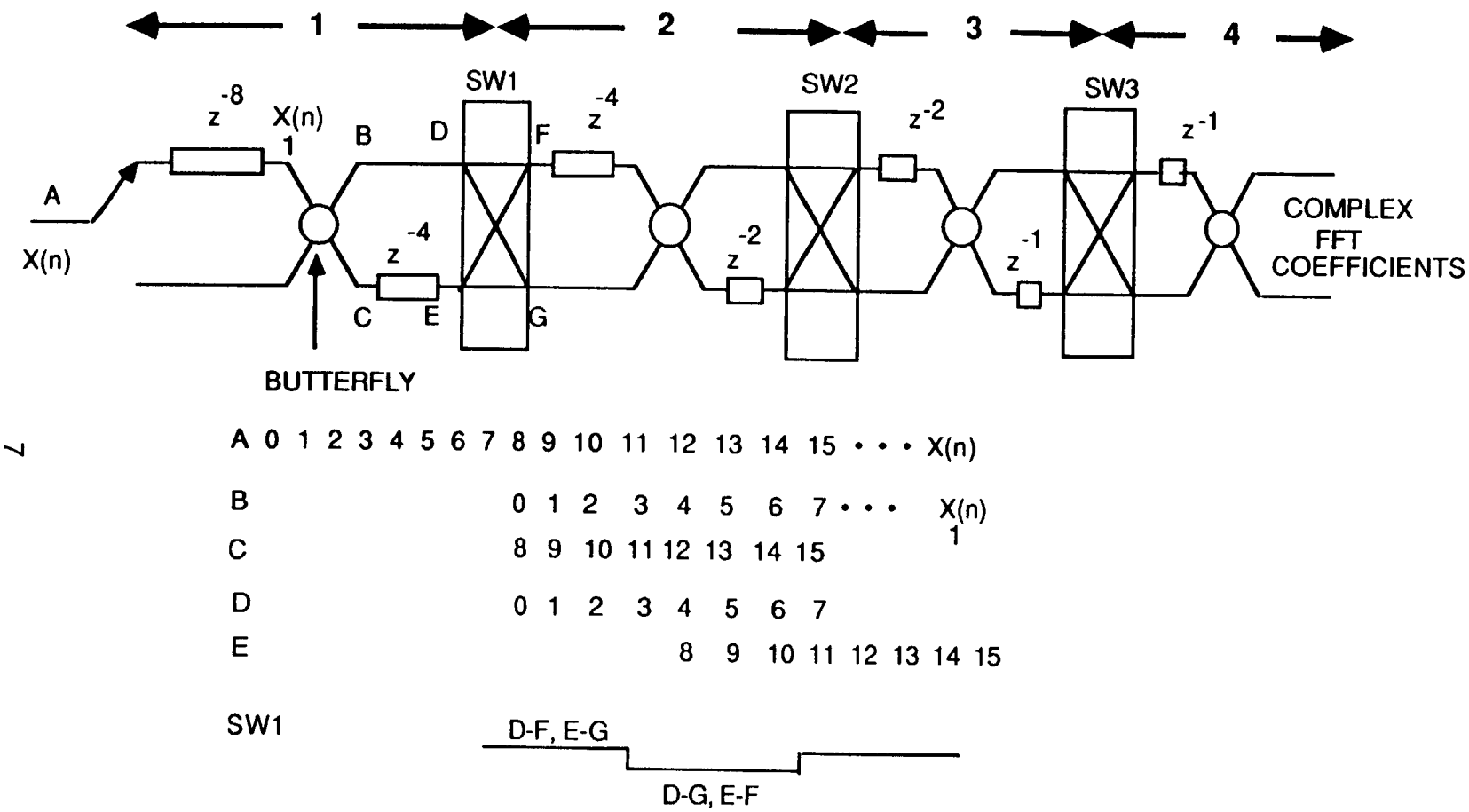


FIGURE 2.1. 16 POINT, RADIX 2 PIPELINE FFT

In the following, examples are given of the pipeline implementation of an FFT processor operating on a 40 MHz wideband multicarrier input signal for three cases involving different choices of multicarrier composition. These are: 1) Demultiplexing 800 narrowband 64kbit/s QPSK carriers, 2) Demultiplexing 24 medium bandwidth 2.048 Mbit/s QPSK carriers and 3) Demultiplexing a mix of 400 narrowband 64 kbit/s and 12 medium bandwidth 2.048 Mbit/s carriers.

2.1.2 EXAMPLE 1, DEMULTIPLEXING OF 800 64KBIT/S CARRIERS

2.1.2.1 BASIC PARAMETER SELECTION

• SAMPLING RATE- 40 MSAMP/S. This rate is established by the Nyquist sampling theorem which for a bandpass of W Hz requires W complex samples per second. It is assumed that each of the 800 64 kbit/s QPSK carriers is assigned to a channel of 45 kHz width. Thus 800 carriers would occupy a bandwidth of 36 MHz. To allow for realization of the anti-aliasing filter needed to select the occupied spectrum, the processed bandwidth must be greater. For this case it is assumed to be 40MHz. Hence the sampling rate is 40 Msamp/s.

• DOWN CONVERSION- Theoretically, it is possible to sample the signal directly at its carrier frequency provided that the carrier has been passed through the anti-aliasing filter and the sampling pulse width is much smaller than a single period of the carrier frequency. At the very high frequencies used for satellite transmission, achieving a sufficiently short sampling pulse width is impractical and it is necessary to down convert the carrier to a lower frequency. Also it is necessary that the relationship between the sampling frequency and the frequency at the center of the original band being sampled be stable and maintained with high accuracy. This requires that the local oscillators for the sampling and the down conversion process have high accuracy. Otherwise it will not be possible to maintain the frequency alignment of the individual channels at baseband. For the narrow band example considered here, the individual channels have a width of 45 kHz and the accuracy should be approximately 1% of the width or 450 Hz. Relative to an uplink band center of 30 GHz this requires individual carrier and frequency conversion accuracy of 6.7×10^{-8}. Accuracy for wider bandwidth or lower carrier frequencies is proportionately less.

In the down conversion process, it is important to select a suitable IF for implementing a practical sampler and associated anti-aliasing filter.In

the present technology, this is in the range up to 100 MHz with 8 bit resolution. The IF can actually be at zero Hz, a choice that eases the sampler design since the highest frequency that occurs is half the channel bandwidth and the sampling rate is equal to the channel bandwidth.

• SAMPLING WINDOW AND INPUT COEFFICIENTS- The Nyquist sampling theorem requires at least one complex sample per time interval B^{-1} where B is the spacing between the individual carriers being demultiplexed. This is one complex sample for each carrier to be demultiplexed in the band being analyzed. These are the input coefficients to the FFT processor. For the example considered here the number is $40 \times 10^6 / 45 \times 10^3 = 888.88$ complex samples per window. However, this results in only one spectrum sample per channel which is insufficient to accurately represent a suitable channel filter. Our simulations indicate that a practical design free of operational constraints requires a sixteen fold increase in the number of samples and consequently in the width of the sampling window. This results in 14222 samples which when rounded up to the nearest power of 2 yields $2^{14} = 16384$. To eliminate undesirable consequences of circular convolution, an "overlap and save" process in which the overlap is 50% of the window width is performed. This is done to eliminate the first half of the samples which suffer aliasing.

2.1.2.2 FORWARD FFT IMPLEMENTATION

The function of the forward FFT in this application is to obtain 16384 complex frequency samples in the 40MHz spectrum occupied by the desired channels. This results in a window of 409.6μs width. To accomplish this a single pipeline processor simultaneously performs an FFT on 50% overlapping sample windows each containing N=16384 complex samples. Hence, the equivalent of 2 pipeline processors are required. These complex samples are processed to translate each of the 800 channels to its baseband (spectrum centered at a carrier frequency of zero Hz).

The processing steps follow:

• BUTTERFLY CALCULATIONS- The pipeline processor will perform $(N/2)\log_2 N = 114688$ butterfly calculations for each FFT sample window. Each sample window has a duration $= 16384 \div 40 \times 10^6 = 409.6\ \mu s$. Each butterfly requires one complex multiply (4 real multiplies and 2 real adds) and two complex adds (4 real adds) for a total of 4 real multiplies and 6 real adds. 16 bit precission is assumed. For processing the 114688 butterfly calculations this yields a total of 458752 real multiplies and

688128 real adds for each 409.6 μs window which corresponds to 1.12 multiplies per ns and 1.68 adds per ns. 50% overlap operation doubles these rates to 2.24 multiplies and 3.36 adds per ns.

• DISTRIBUTION OF THE CALCULATIONS- The pipeline FFT processor for this example will consist of a cascade of 14 butterfly stages. The calculations are equally distributed among these and accordingly the rates will be reduced to 160 multiplies per μs and 240 adds per μs in each stage. These correspond to 6.25 ns per multiply and 4.17 ns per add. Since there are 4 real multiplies and 6 real adds per butterfly and if these are implemented separately , there is a further rate reduction resulting in 25 ns per multiply and 25 ns per add. In this case the pipeline processor would contain 4 x14 = 56 multipliers and 6 x14 = 84 adders.

• MEMORY REQUIREMENT- As shown in Figure 2.1, delay memories are required in each stage to achieve proper time alignment of the samples as they are processed in the butterflies. There is a single delay of N/2 complex samples in the first stage and a pair of delays of $N/2^k$ complex samples in each kth stage for $2 \le k \le \log_2 N$. The total number of real samples in the delay memories of the entire pipeline processor is

$$2[\, N/2 + 2 \sum_{k=2}^{k=\log_2 N} (\, N/2^k)] = 3N - 4$$

The above expression yields 3x16384 - 4 = 49148 real samples. If each sample is 16 bits, the the total memory capacity of the pipeline FFT processor is 98.3 Kbytes. There are 27 memories ranging in size from 4 bytes (one complex sample) to 32768 bytes (8192 complex samples). The propagation time in passing through the FFT processing is N/W which for this case is 409.6 μs. The memories operate at a 40 Msamp/s rate.

2.1.2.3 FREQUENCY DOMAIN PRODUCT.

The purpose of this operation is to select and shape the spectrum of each recovered channel. It is performed by calculating the product of the complex samples from the 16384 complex coefficient FFT and a model of the channel filter expressed in the form of a set of 16 complex coefficients selected to represent the desired filter characteristic (amplitude and phase). An example of such a filter is shown in Figure 2.2. These filter coefficients are selected so that the resulting impulse

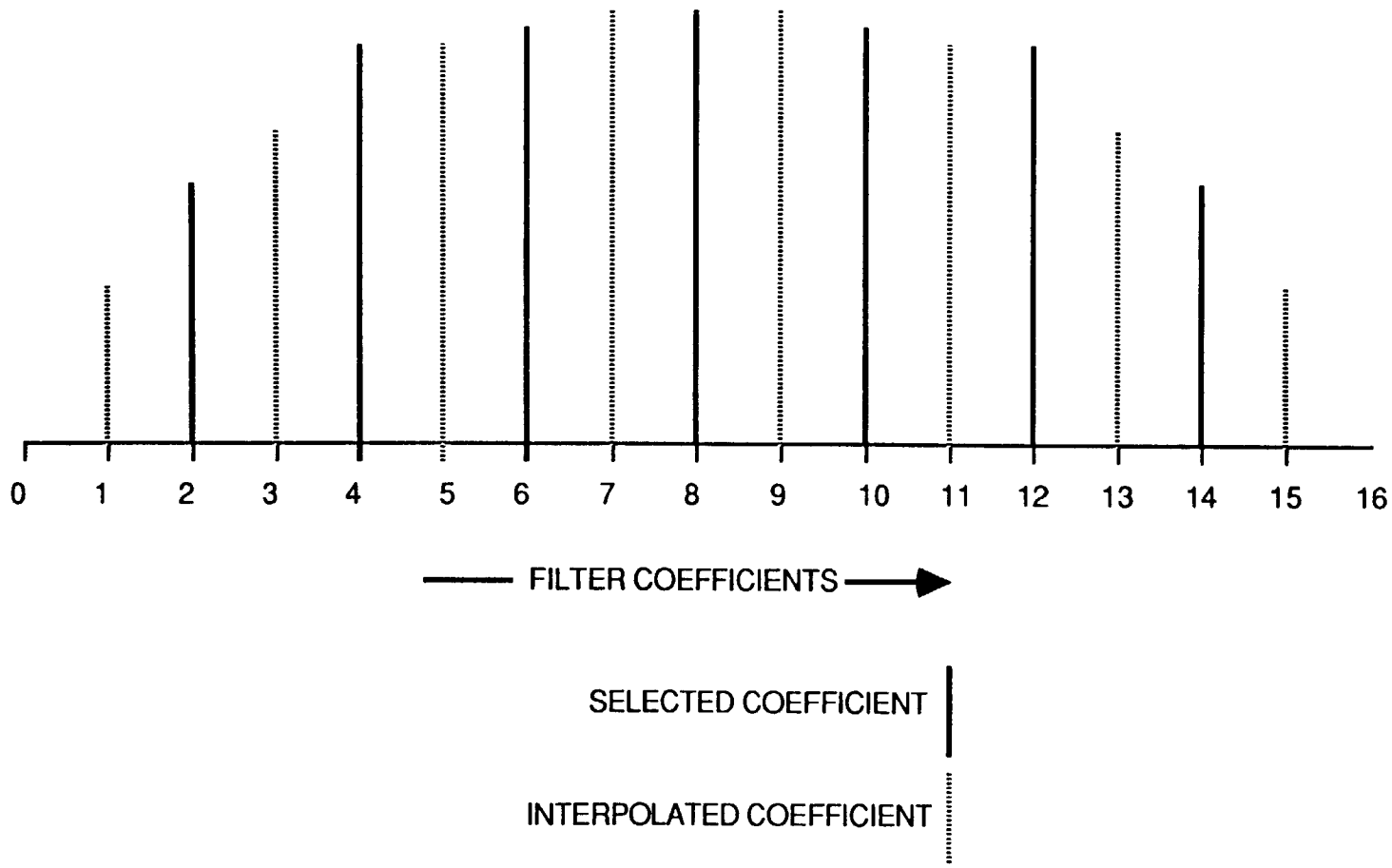


FIGURE 2.2 COEFFICIENTS OF THE INDIVIDUAL CHANNEL FILTER

response is zero in the second half of the 16384 point sampling window. This is done to eliminate unwanted aliasing contributions caused by circular convolution. A more detailed description of how the channel filter coefficients are determined is given in another section. Only those 16 complex FFT coefficients corresponding to the frequency locations of the16 complex filter coefficients need be considered to demultiplex a given channel because all of the other filter coefficients are zero. Hence, for each 45 kHZ channel, the 16 complex coefficient filter function multiplies the 2x16 complex FFT coefficients of the overlapped windows having a width of 409.6 μs as illustrated in Figure 2.3 (Two overlapping windows need to be processed during each sampling window for each channel). Therefore the rate of complex multiplies is 2x16/409.6μs = 0.0781/μs which is equivalent to 12.8 μs per complex multiply or 3.2 μs per real multiply and 6.4 μs per real add. The result of the frequency domain product consists of 16 non-zero complex frequency coefficients out of a total of 16384 coefficients occurring for each window. By interpreting the frequencies represented by the coefficients to be those that are symmetrical about zero Hz for each channel, the channel is automatically converted to the desired baseband form.

2.1.2.4 INVERSE DISCRETE FOURIER TRANSFORM.

An inverse discrete Fourier transform is used to convert the complex frequency domain coefficients for each channel to the sampled data time domain form. An IDFT rather than an IFFT is used because at the input only a small number of non-zero coefficients are presented and at the output only a small fraction of the samples need to be calculated. The IDFT calculation is of the form shown in Figure 2.4 and is performed separately for each window. If the full IDFT were determined for each window, the result would be 16384 time domain samples the first half of which would be discarded because they are aliased and from the second half only a fraction are needed because of decimation. By anticipating this only those coefficients needed will be computed, greatly reducing the computational load. Since for each modulated symbol period only two samples are needed and these are for half a window and since there are 13 32 ksym/s symbol periods per 409.6 μs window for each channel for the example being treated, the number of calculations per window for each channel is 16 x 13 = 208 complex multiplies every 409.6μs. Because this calculation must be performed for each of the overlapping windows the above calculation rate must be doubled to 416 every 409.6 μs. The results of the calculations of both sets of windows taken together constitute the complex sampled data that is to be used for subsequent demodulation of the data signal.

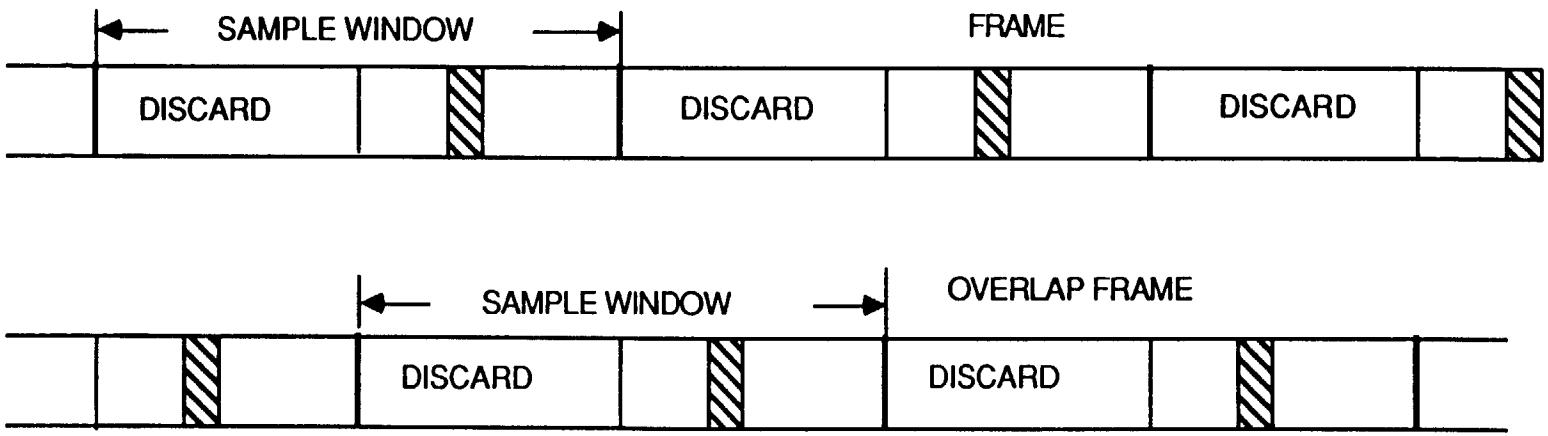


FIGURE 2.3 ILLUSTRATION OF 50% OVERLAPPED IDFT SAMPLE FRAMES

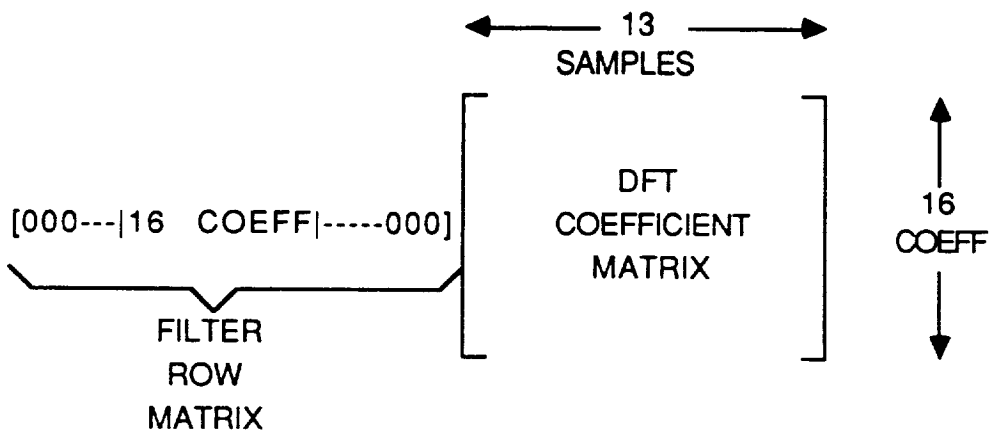


FIGURE 2.4. INVERSE DISCRETE FOURIER TRANSFORM CALCULATION

2.1.2.5 ESTIMATE OF THE IMPLEMENTATION POWER REQUIREMENTS.

The following presents estimates of the power requirements for implementing the multiplications involved in the Foward FFT, Frequency Multiplication and IDFT functions of an onboard processor to accomplish the demultiplexing of the 800 channels of the example being considered. The power requirements of the adders is expected to be quite small compared to that required for the multipliers. The estimates presented are based on present day technology and are expected to be considerably better with devices that will be available in the future.

• FORWARD FFT- This function requires 56 multipliers each operating at a rate of one multiply every 25 ns. Toshiba manufactures a 16 x16 bit CMOS/SOS VLSI multiplier with an operation time of 27 ns and a power dissipation of 150 mw. For a guideline it is assumed that this unit can be improved to a speed of 25 ns without increased power. Consequently, the estimated power needed to implement the FFT multipliers is 56 x 0.15 = 8.4 w.

• FREQUENCY MULTIPLIER- This function requires a rate of 0.3125 real multiplies/µs for each of 800 channels yielding a total of 0.25 multiplies/ns or 4ns/ multiply. This rate can be satisfied by using 6 of the guideline multipliers which would require a total power of 6x0.15 = 0.9 w.

•INVERSE DFT- This function requires the determination of13 time domain samples each requiring 16 complex multiplies for each of two overlapping windows. Since each complex multiply requires 4 real multiplies, the number of real multiplies per channel for each 409.6µs window is 16x13x2x4 = 1664 or 4.0625/µs. For 800 channels this becomes 3.25 multiplies per ns. This can be satisfied by using 81 of the guideline multipliers which yields a power requirement of 12.2 w.

2.1.3 EXAMPLE 2. DEMULTIPLEXING of 24 2.048 CARRIERS.

2.1.3.1 BASIC PARAMETER SELECTION.

• SAMPLING RATE- 40 MSAMP/S. This rate depends only on the 40 MHz spacing of the transponder and the need to accommodate the anti-aliasing filter for realizing an occupied bandwidth of 36 MHz. It is independent of the number of channels to be demultiplexed.

• DOWN CONVERSION- Same as for example1.

• SAMPLING WINDOW- The same argument given for example1 applies to this case with appropriate changes to account for the difference in the carriers. For a 2.048 Mbit/s QPSK carrier the practical spacing between channels should be 1.4 times the symbol rate bandwidth yielding a spacing between carriers of 1.4 MHz. In the 36 MHz bandwidth of the transponder, 24 of these can easily be accommodated. The minimum number of complex samples per window needed to represent such channels is $40x10^6/1.4x10^6 = 28.57$. However practical filter design dictates that this be increased 16 fold to 457 and when rounded up to the nearest power of 2 this becomes $2^9 = 512$.

2.1.3.2 FORWARD FFT IMPLEMENTATION.

In this example the function of the forward FFT is to obtain N = 512 complex frequency samples in the 40 MHz spectrum occupied by the transponder. A pipeline FFT implementation is used to accomplish this.

• BUTTERFLY CALCULATIONS- The pipeline processor will perform $(N/2)\log_2 N = 2304$ butterfly calculations for each FFT sample window. Each sample window has a duration $= 512 \div 40x10^6 = 12.8\ \mu s$. Each butterfly requires one complex multiply comprising 4 real multiplies and 6 real adds. For processing the 2304 butterfly calculations this yields a total of 9216 real multiplies and 13824 real adds for each 12.8 µs window which corresponds to 0.72 multiplies and 1.08 adds per ns. 50% overlap operation doubles these rates to 1.44 multiplies and 2.16 adds per ns.

• DISTRIBUTION OF THE CALCULATIONS- The pipeline FFT processor for this example will consist of a cascade of 9 butterfly stages. The calculations are equally distributed among these and accordingly the rates will be reduced to 160 multiplies and 240 adds per µs in each stage. These correspond to 6.25 ns per multiply and 4.17 ns per add. Since there are 4 real multiplies and 6 real adds per butterfly and if these are implemented separately , there is a further rate reduction resulting in 25 ns per multiply and 25 ns per add. These rates are the same as those calculated in example1. However in this case the pipeline processor would contain 4 x 9 = 36 multipliers and 6 x 9 = 54 adders.

• MEMORY REQUIREMENT- Using the same expression developed for the memory size in example1, with N = 512, the memory requirement is 3x512- 4 = 1532 real samples. If each sample is 16 bits, then the total memory capacity of the pipeline FFT processor is 3 Kbytes. There are 17 memories ranging in size from 4 bytes (one complex sample) to 512 bytes

(128 complex samples). The propagation time in passing through the FFT processing is N/W which for this case is 12.8 μs. The memories operate at a 40 Msamp/s rate.

2.1.3.3 FREQUENCY DOMAIN PRODUCT.

The purpose of this operation is to select and shape the spectrum of each recovered channel. It is performed by calculating the product of the complex samples from the 512 complex coefficient FFT and a 16 complex coefficient model of the channel filter selected to represent the desired filter characteristic (amplitude and phase). The filter coefficients are selected by the method described in example1 which eliminates unwanted aliasing contributions caused by circular convolution. Only those 16 complex FFT coefficients corresponding to the frequency locations of the16 complex filter coefficients need be considered to demultiplex a given channel because all of the other filter coefficients are zero. Overlap and save operation requires that two sets of16 complex frequency coefficients be processed for each window. Hence, for each 1.4 MHZ channel, the 16 complex coefficient filter function multiplies the 2x16 complex FFT coefficients of the overlapped windows having a width of 12.8 μs. Therefore the rate of complex multiplies is 2x16/12.8μs = 2.5/μs which is equivalent to 0.4 μs per complex multiply or 0.1 μs per real multiply and 0.2 μs per real add. The result of the frequency domain product consists of 16 non-zero complex frequency coefficients out of a total of 512 coefficients occurring for each window. By interpreting the frequencies represented by the coefficients to be those that are symmetrical about zero Hz for each channel, the channel is automatically converted to the desired baseband form.

2.1.3.4 INVERSE DISCRETE FOURIER TRANSFORM

The procedure used is the same as that described for example1 with the number of samples per window being 512. Based on the observation that for each symbol period only two samples are needed and these are for half a window and since there are 13 symbol periods per window for the example being treated, the number of calculations per window for each channel is 16 x 13 = 208 complex multiplies every 12.8μs. Because this calculation must be performed for each of the overlapping windows the above calculation rate must be doubled to 416 every 12.8 μs . The results of the calculations of both sets of windows taken together constitute the complex sampled data that is to be used for subsequent demodulation of the data signal.

2.1.3.5 ESTIMATE OF THE IMPLEMENTATION POWER REQUIREMENTS

The following presents estimates of the power requirements for implementing the multipliers in the Forward FFT, Frequency Multiplication and IDFT functions of an onboard processor to accomplish the demultiplexing of the 24 2.048Mbit/s channels of the example being considered.

• FORWARD FFT- This function requires 36 multipliers each operating at a rate of one multiply every 25 ns. Using the Toshiba 16 x16 bit CMOS/SOS VLSI multiplier with an operate time of 27 ns with a power dissipation of 150 mw as a guideline, the estimated power needed to implement the FFT multipliers is 36 x .15 = 5.4 w.

• FREQUENCY MULTIPLIER- This function requires a rate of 10 real multiplies/µs for each of 24 channels yielding a total of 0.24 multiplies/ns or 4.25 ns/ multiply. This rate can be satisfied by using 6 of the guideline multipliers which would require a total power of 6x0.15 = 0.9 w.

•INVERSE DFT- This function requires the determination of13 time domain samples each requiring 16 complex multiplies for each of two overlapping windows. Since each complex multiply requires 4 real multiplies, the number of real multiplies per channel for each 12.8µs window is 16x13x2x4 = 1664 or 130/µs. For 24 channels this becomes 3.12 multiplies per ns. This can be satisfied by using 78 of the guideline multipliers which yields a power requirement of 11.7 w.

2.1.4 EXAMPLE 3, DEMULTIPLEXING OF 400 64KBIT/S AND 12 2.048 MBIT/S CARRIERS

2.1.4.1 BASIC PARAMETER SELECTION

• SAMPLING RATE- 40 MSAMP/S. This rate depends only on the 40 MHz spacing of the transponder and the need to accommodate the anti-aliasing filter for realizing an occupied bandwidth of 36 MHz. It is independent of the number, bandwidth and distribution of channels to be demultiplexed. Each of the 400 64 kbit/s QPSK carriers is assigned to a channel of 45 kHz width in one half of the wideband and each of the 12 2.048 Mbit/s QPSK carriers to a channel of 1.4 MHz width in the other half. However, channels of a given bandwidth need not be grouped together because the spectrum coefficients of any channel are independently selected.

• DOWN CONVERSION- Same as for example1.

• SAMPLING WINDOW AND INPUT COEFFICIENTS- It is assumed that the same FFT processor processes carriers of both carrier channel widths. The narrowband carriers drive the resolution requirement. This being the case, the sampling window and number of FFT coefficients processed are the same as for example 1. Therefore the number of samples will be 16384 and the window width 409.6 µs. Processing of narrow bandwidth carriers in a given broadband is more computationally intense than wider bandwidth carriers. Alternatives for minimizing the computational intensity needed for mixed bandwidth situations are treated later.

2.1.4.2 FORWARD FFT IMPLEMENTATION

Since it is assumed that a common forward FFT pipeline processor will be used to process channels of different bandwidths, its frequency resolution is determined by the narrowest bandwidth channel which is 64 kHz. Thus, its implementation is the same as that described in example 1.

2.1.4.3 FREQUENCY DOMAIN PRODUCT

As in example 1, for each of the 64 Kbit/sec carriers, a 16 complex coefficient filter function multiplies the 2 x 16 complex FFT coefficients of the over lapped windows. Each 2.048 Mbit/sec carrier, on the other hand, occupies a bandwidth 32 times larger than the 64 Kbit/sec carriers and therefore a 512, (32 x 16), complex coefficient filter function is used to multiply the 2 x 512 complex FFT coefficients of the overlapped windows.

2.1.4.4 INVERSE DISCRETE FOURIER TRANSFORM

As in example 1, for each of the 64 kbit/sec carriers, the number of calculations per window for each channel is 208 complex multiplies every 409.6µs. Because this calculation must be performed for each of the overlapping windows, the above calculation rate must be doubled to 416 every 409.6 µs for each narrowband carrier. For the 2.048 Mbit/sec carriers, the number of frequency coefficients in each window is 32 times larger than for the 64 Kbit/sec carrier. The number of resulting time domain samples are also 32 times larger. Thus, 416x32x32 complex multiplies are required every 409.6 µs for each wideband carrier. This high computationally intensity for the 2.048 Mbit/sec carriers is the consequence of mixed bandwidth operation and use of the IDFT. A much

more efficient IFFT method is discussed in the next section.

2.1.4.5 ESTIMATE OF THE IMPLEMENTATION POWER REQUIREMENTS

The following presents estimates of the power requirements for implementing the multipliers in the Forward FFT, Frequency Multiplication and IDFT functions of an onboard processor to accomplish the demultiplexing of 400 64 Kbit/sec carriers and 12 2.048 Mbit/sec carriers in a 40 Mhz bandwidth.

• FORWARD FFT- This function requires 56 multipliers each operating at a rate of one multiply every 25 ns. Using the Toshiba 16 x 16 bit CMOS/SOS multiplier as a guideline, the estimated power needed to implement the FFT multipliers is 56 x 15 = 8.4 w.

• FREQUENCY MULTIPLIER- This function requires a rate of 0.3l25 real multiplies/μs for each of the 64 Kbit/s carriers and a rate 32 times larger for each of the 2.048 Mbit/sec carriers. This yields a total of 0.25 multiplies/ns or 4ns/multiply. This rate can be satisfied by using 6 of the guideline multipliers which would require a total power of 6 x 0.15 = 0.9w.

• INVERSE DFT- For each of the 64 Kbit/s carriers, the number of real multiplies per channel for each 409.6 μs window is 416 x 4 = 1664 or 4.0625/μs. For each of the 2.048 Mbit/sec carriers, the number of real multiplies per channel for each 409.6 μs window is 416 x 1024 x 4 = 1,703,936 or 4160/μs. For the 400 narrow bandwidth carriers and the 12 wide bandwidth carriers, this becomes 51.545 multiplies per ns. This can be satisfied by using 1289 of the guideline multipliers which yields a power requirement of 194w.

2.1.5 SUMMARY OF SPEED AND POWER

The results of the demultiplexer implementations for the three examples considered in the foregoing are tabulated in Table 2.1. Clearly in the case of mixed size carriers where the ratio of the widest to narrowest carrier bit rate and bandwidth is high (32 in our example), the use of the IDFT to recover the time samples of the individual carriers from the frequency coefficients of the forward FFT is very computationally intensive and power consuming. The use of an IFFT followed by an interpolating filter is therefore perferred to the use of the IDFT. This will be discussed in detail in the next section where the IFFT approach will be shown to be much more efficient.

TABLE 2.1
SUMMARY OF MULTIPLIER SPEED AND POWER REQUIREMENTS
FOR THREE EXAMPLES CONSIDERED

EXAMPLE	MULT/ns	POWER, w
800 64KBIT/S CHANNELS	5.74	21.5
24 2.048 MBIT/S CHANNELS	4.85	18.0
400 64KBIT/S + 12 2.048 MBIT/S CHANNELS	54.03	203.3

THE RESULTS GIVEN ABOVE ARE BASED ON A WIDEBAND MULTICARRIER INPUT SIGNAL OF 40 MHz BANDWIDTH A 16X16 BIT MULTIPLER WITH A 25 ns OPERATE TIME AND A POWER DISSIPATION OF150mw

2.2 DEMULTIPLEXER IMPLEMENTATION WITH A PIPELINE FFT AND IFFT FOR MULTIPLE BANDWIDTH CARRIER OPERATION

2.2.1 GENERAL

For the onboard demultiplexer/demodulator to be fully flexible and useful, it must be able to demultiplex multiple carriers of different bandwidths. The discussion of the previous section revealed that although the use of the IDFT is suitable for recovering multiple carriers all of the same bandwidth, it is excessively computationally intensive and power consuming for use with multiple carriers of mixed bandwidths in the wideband signal being processed. To overcome this difficulty, it is best to use an IFFT, implemented using the pipeline approach, in place of the IDFT. The resulting computational intensity is significantly reduced. It is also influenced by the choice of implementation of the forward FFT.

This section addresses three ways to accomplish the multiple bandwidth operation which vary with regard to the forward FFT: a Single Large FFT Processor, a Cascade FFT Processor and a Parallel FFT Processor. These implementations are described in the following and a comparison is made of their relative performance in terms of the number of complex multiplications needed to process a block of 16384 (2^{14}) complex time domain input signal samples. This number is determined by the narrowest bandwidth to be processed. It is assumed that the processor is to demutiplex an input signal spectrum containing 512 narrowband carriers in one half of the spectrum space and 16 wideband carriers in the the other half of the spectrum space, where each wideband carrier has a width equal to 32 narrowband carriers. The extension to accommodating more bandwidths is obvious. Carriers of a certain bandwidth may be grouped together in each half of the spectrum or they may be in disconnected groups distributed arbitrarily in the spectrum. The comparison is based on the number of multiplications required for each FFT and IFFT and must be doubled to account for "overlap and save"

The results show that the Single Large FFT Processor transforms all carriers to baseband with the least number of multiplications.

2.2.2 SINGLE LARGE FFT PROCESSOR

The single large FFT processor is illustrated in Figure 2.5 and the number of complex multiplications required in the various steps of

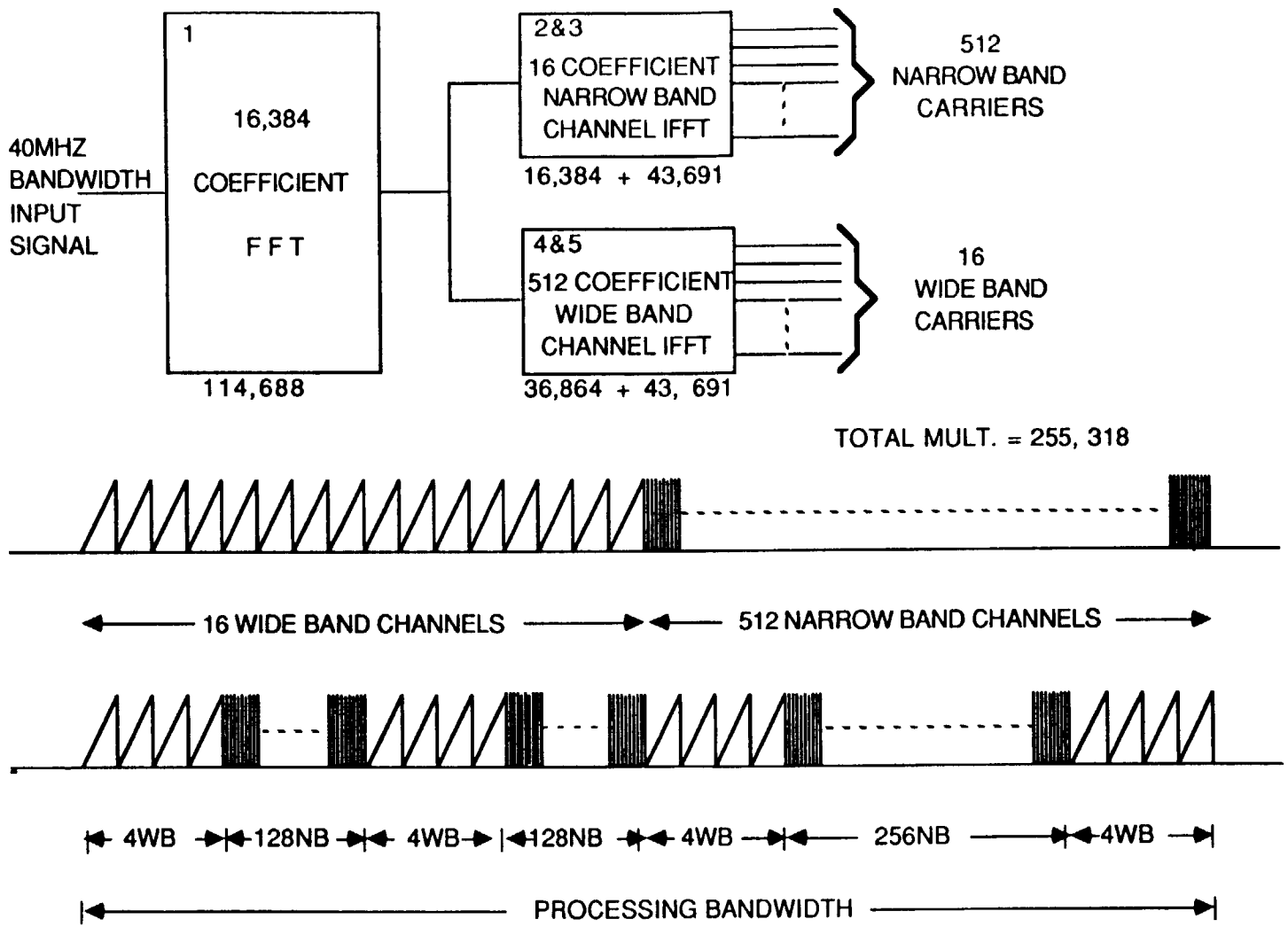


FIGURE 2.5. SINGLE LARGE FFT PROCESSOR

processing are given in Table 2.2. Each step is numbered in the figure and the table. The first processing step is to calculate an FFT that is sufficient to provide a resolution that supports the narrowest bandwidth carriers expected. In the example considered, this narrowest bandwidth is determined by allocating 1024 carriers in the input signal band and for each it is assumed that the narrowband processing channel filter can be suitably realized using 16 frequency coefficients, thus yielding 16 x 1024 = 16384 frequency coefficients. The input signal is sampled in complex form at a rate of W where W is the width of the spectrum assigned to the composite of carriers to be demultiplexed. The complex samples are presented to the FFT processor in blocks of 16384 and the duration of a block is 16384/W. The number of multiplies needed to perform this FFT is approximately $(N/2)\log_2 N$ where N is the number of coefficients. The resulting number of complex multiplies for step 1 is 114,688 for N = 16384.

The processing represented by steps 2 and 3 in Table 2.2 and Figure 2.5 convert selected subsets of FFT coefficients corresponding to the frequency locations of the narrowband channels into the complex signal basebands of 512 narrowband carriers. This is done by multiplying the FFT coefficients by the 16 frequency coefficients of the channel filter and performing an IFFT for each of 512 narrowband carriers. This requires 8 x 4 ($(N/2/)\log_2 N$, N =16) multiplications for each filter, yielding a total of 16,384 complex multiplies. Next, 8 time domain samples resulting from each of the 50 % overlapping sample blocks must be interpolated to derive samples aligned with the symbols of the digitally modulated carrier. This interpolation requires 8 multiplies for each complex sample and the number of samples is the product of the number of IFFT samples and the ratio of the bandwidth W to the symbol rate R. This latter ratio is assumed to be 4/3 for typical QPSK modulated carriers. Thus, the interpolation of the samples requires 8 x 8 x 4/3 complex multiplications for each of the 512 narrowband channels. This yields a total of 43, 691 multiplications for each block of 16384 samples.

In a similar manner the wideband processor recovers the basebands of the 16 wideband carriers in steps 4 and 5. Since these filters are 32 times wider than the narrowband filters, they will contain 16 x 32 = 512 FFT coefficients for each wideband channel. The FFT coefficients corresponding to each wanted channel location are multiplied by the 512 coefficients of the channel filter representing the wideband filter. The resulting frequency coefficients are converted to time domain samples by

an IFFT which requires 256 x 9 ((N/2)$\log_2$N with N=512) complex multiplications for each of the 16 channels processed yielding 16 x 256 x 9 = 38,864 multiplications. This is followed by interpolation processing of the 256 time domain samples produced by the IFFT to generate 4/3 x 256 samples properly aligned with the symbols of the digitally modulated carrier. Since each interpolated sample requires 8 complex multiplications, a total of 16 x 8 x 256 x 4/3 = 43,691 complex multiplications are required for each block of 16384 samples.

The net total of complex multiplications required to process each block of 16384 input complex samples to recover the basebands of 512 narrowband and 16 wideband channels assumed in the model analyzed is 255,318 as given in Table 2.2. The wideband and narrowband carriers can be located anywhere in the input signal band. Two possible arrangements are illustrated in Figure 2.5.

2.2.3 CASCADE FFT PROCESSOR

The configuration of the cascade FFT processor for accomplishing demultiplexing of carriers of two different bandwidths is shown in Figure 2.6. The concept is to first process the input signal into the wide bands in step 1. Those carriers having the narrow bandwidth are processed by a 256 coefficient IFFT in step 2 to convert them back to time domain sampled signal form. These time domain samples are selected from 32 blocks each of two streams of 50% overlapping blocks yielding a block of 8192 time domain samples which are converted to 8192 frequency domain coefficients by the FFT processor of step 3. The latter are multiplied by the 16 coefficients of each of the 512 narrowband filters and these are converted to the 512 basebands by the 16 coefficient IFFT and the sample interpolation processing performed in steps 4 and 5. Those carriers having the wide bands are processed directly to their basebands using the IFFT and associated sample interpolator represented by processing steps 6 and 7.

The number of complex multiplications needed to accomplish each step are tabulated in Table 2.3. Note that the input wideband FFT has only 512 coefficients as determined by the bandwidth requirement compared to the 16384 coefficients for the narrow bands. This is a ratio of 32:1. Thus when converting to the FFT needed for the narrowband filters, 32 blocks of the input FFT processor output are aggregated to form one block for the narrowband processor. In Table 2.3 this fact is indicated in the column titled "replications per 16384 samples". The number of complex

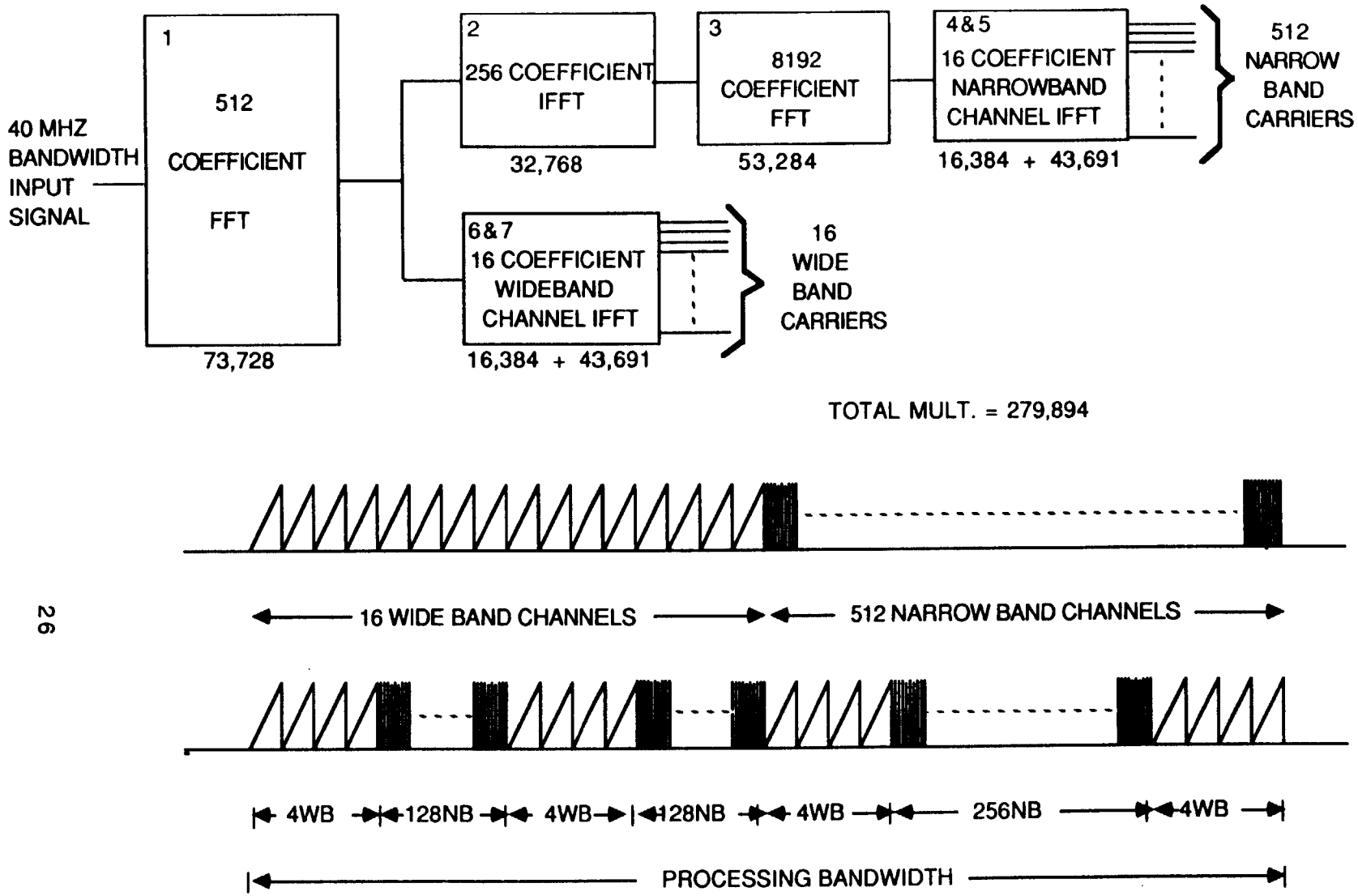


FIGURE 2.6. CASCADE FFT PROCESSOR

multiplications required for each step are tabulated in the rightmost column of the table. The logic used to arrive at these numbers is the same as that previously described for the single large FFT processor and is not repeated here. The total number of complex multiplies needed to convert 512 narrowband and 16 wideband channels for the cascade FFT processor is 279,894 which is greater than that needed for the single large FFT processor.

Because the narrowband carriers are processed in bundles of 32 which equal the width of the wideband carrier channel, the flexibility to adjust their locations in the input signal spectrum is limited to bundles of 32.

2.2.4 PARALLEL FFT PROCESSOR

The configuration of the parallel FFT processor for demultiplexing carriers of two different bandwidths is shown in Figure 2.7. The concept provides a separate processor for each bandwidth accommodated. For the narrowband carriers a 16384 coefficient FFT is used in step 1, followed by a 16 coefficient IFFT and sample interpolator in steps 2 and 3. For the wideband carriers a 512 coefficient FFT is used in step 4, followed by a 16 coefficient IFFT and sample interpolator. The number of complex multiplications required for each step is given in Table 2.4. The total number required for processing 512 narrowband and 16 wideband carriers is 308, 576 which is greater than either of the other methods described above. This result is not surprising since the other methods share a common input FFT processor while the parallel method requires a separate input processor for each bandwidth accommodated.

2.2.5 GENERIC PROCESSOR

Each figure appearing in the text illustrates two example distributions of the wide and narrowband channels. Virtually any arrangement of the channels can be accommodated with only minor additional calculations required to perform frequency translations between the output of the input FFT and the inputs to the narrowband and wideband IFFT processors respectively. In this discussion, only two bandwidths have been considered. In an actual processor many more bandwidths can be accommodated with very little change in the number of multiplications required since the same number of input samples are shared among all processors and each operates at a rate dictated by its share of the total signal spectrum. Furthermore, each processor can be given an amount of processing power sufficient to accomplish its most

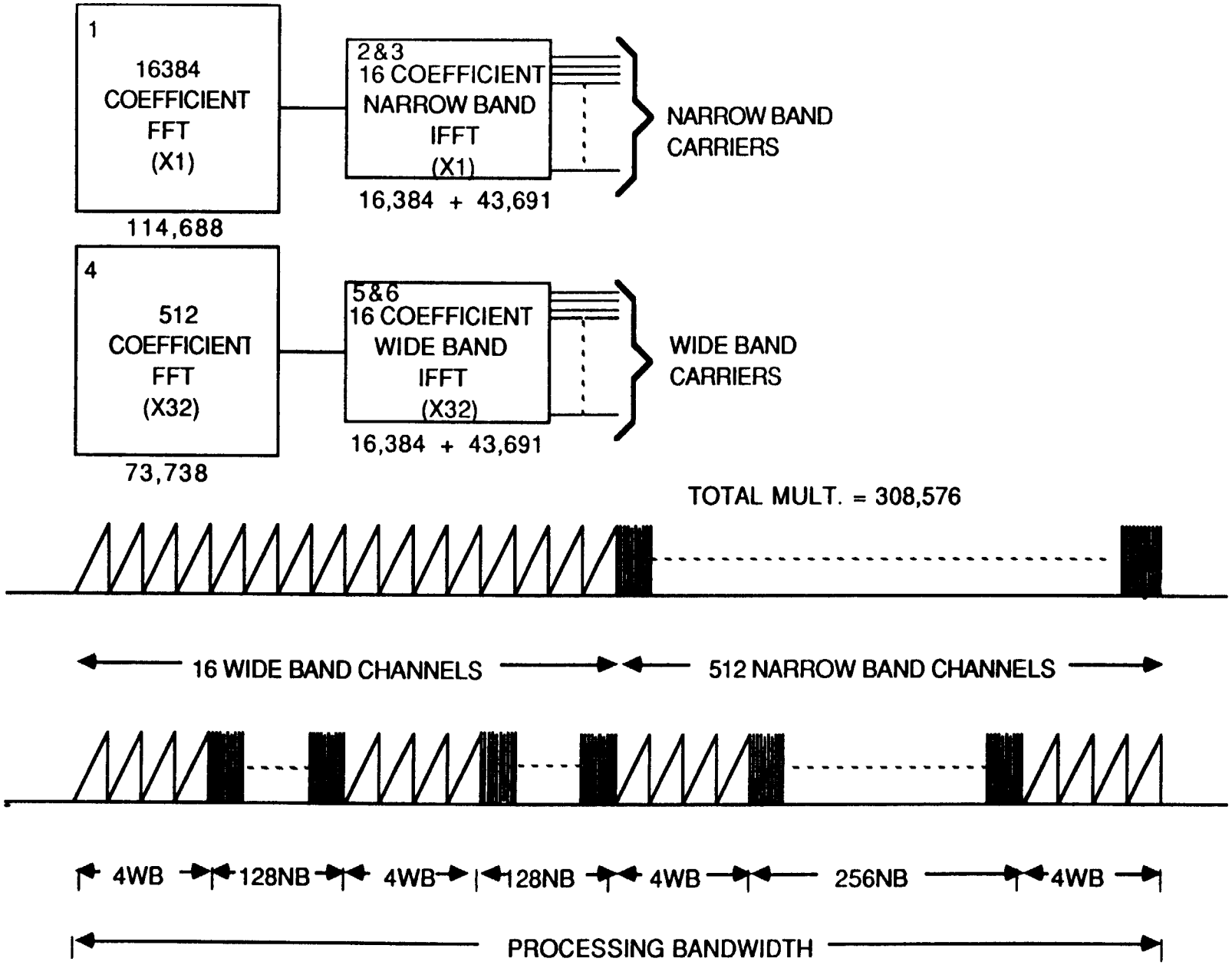


FIGURE 2.7. PARALLEL FFT PROCESSOR

difficult task and be reprogrammed to perform any lesser task. Thus the unit may contain a number of generic processors that can be programmed after launch and reprogrammed during their life to accommodate differing demands. An example of this is seen in the single large FFT processor for which 16,384 + 43691 = 60,075 multiplications are required for narrowband channels and 38,864 + 43,691 = 82,555 multiplications are required for the wideband channels. A generic processor having the greater capability can do either job. For instance, if a 14 stage pipeline processor is available and only a 9 stage FFT is needed, then the last 5 stages can be inhibited by microprocessor control.

TABLE 2.2

NUMBER OF MULTIPLICATIONS FOR
A SINGLE LARGE FFT PROCESSOR
PER 16384 COMPLEX TIME DOMAIN SAMPLES
(DOUBLE VALUES FOR OVERLAP AND SAVE)

PROCESSOR TYPE	REPLICATIONS PER 16384 SAMP	COMPLEX MULTIPLIERS	TOTAL
COMMON FFT:			
1) 16384 COEFF. FFT	1	8192 x 14	114,688
512 NARROWBAND CHANNELS:			
2) 512 x 16 COEFF. IFFT	1	512 x 8 x 4	16,384
3) 512 X 8 x 8 x 4/3 INTERP.	1	512 x 8 x 8 X 4/3	43,691
16 WIDEBAND CHANNELS:			
4) 16 x 512 COEFF. IFFT	1	16 X 256 X 9	36,864
5) 16 X 8 X 256 X 4/3 INTERP.	1	16 x 8 x 256 x 4/3	43,691
		GRAND TOTAL	255,318

TABLE 2.3

NUMBER OF MULTIPLICATIONS FOR
A CASCADE FFT PROCESSOR
PER 16384 COMPLEX TIME DOMAIN SAMPLES
(DOUBLE VALUES FOR OVERLAP AND SAVE)

PROCESSOR TYPE	REPLICATIONS PER 16384 SAMP	COMPLEX MULTIPLIERS	TOTAL
COMMON FFT:			
1) 512 COEFF. FFT	32	32 x 256 x 9	73,728
512 NARROWBAND CHANNELS:			
2) 256 COEFF. IFFT	32	32 x 128 x 8	32,768
3) 8192 COEFF. FFT	1	4096 X 13	53,248
4) 512 x 16 COEFF. IFFT	1	512 x 8 x 4	16,384
5) 512 X 8 x 8 x 4/3 INTERP.	1	512 x 8 x 8 X 4/3	43,691
16 WIDEBAND CHANNELS:			
6) 16 x 16 COEFF. IFFT	32	32 x 16 X 8 X 4	16,384
7) 16 X 8 X 8 X 4/3 INTERP.	32	32 x 16 x 8 x 8 x 4/3	43,691
		GRAND TOTAL	279,894

TABLE 2.4

NUMBER OF MULTIPLICATIONS FOR
PARALLEL FFT PROCESSOR
PER 16384 COMPLEX TIME DOMAIN SAMPLES
(DOUBLE VALUES FOR OVERLAP AND SAVE)

PROCESSOR TYPE	REPLICATIONS PER 16384 SAMP	COMPLEX MULTIPLIERS	TOTAL
512 NARROWBAND CHANNELS:			
1) 16384 COEFF. FFT	1	8192 x 14	114,688
2) 512 x 16 COEFF. IFFT	1	1 X 512 X 8 X 4	16,384
3) 512 X 8 X 8 X 4/3 INTERP.	1	512 X 8 X 8 X 4/3	43,691
16 WIDEBAND CHANNELS:			
4) 512 COEFF. FFT	32	32 x 256 x 9	73,738
5) 16 x 16 COEFF. IFFT	32	32 x 16 x 8 x 4	16,384
6) 16 X 8 X 8 X 4/3 INTERP	32	512 x 8 x 8 x 4/3	43,691
		GRAND TOTAL	308,576

2.2.6 POWER ESTIMATES FOR THE FFT-IFFT IMPLEMENTATION

Tables 2.2, 2.3 and 2.4 present the number of complex multiplications required in processing a window of 16, 384 samples through a FFT, an IFFT and an interpolating filter. As mentioned in a previous section the use of an IFFT followed by an interpolating filter is more efficient than using an IDFT when carriers of widely varying bandwidths are to be demultiplexed. To obtain power estimates from Tables 2.2-2.4 proceed as follows. Assuming a 40 MHz bandwidth (including the guardbands at the edges) and a 40 MHz sampling rate, a window of 16384 time samples has a duration $16384 \div (40 \times 10^6) = 409.6\ \mu s$. During 409.6 µs, two windows must be processed because of the overlap operation. Thus the grand totals shown in Tables 1-3 represent the number of complex multiplications in 409.6/2 = 204.8 µs. With 4 real multiplications per complex multiplication and using the guideline multiplier of 25 ns and 150 mw, we obtain the following estimates.

For the single large FFT processor (Table 2.2), the number of multipliers required to perform the demultiplexing and interpolation functions become:

$$255318 \times 4 \times 25 \div (204.8 \times 10^3) = 125 \text{ multipliers}$$

Using150 mw/multiplier, the net power dissipation is:

- Large FFT Processor Power = 125x0.150 =18.8w.

The above number represents the estimated power required to perform the necessary multiplications in the demultiplexing and interpolation processes. As we mentioned earlier, the power required for the additions is a small fraction of the power required for multiplications. Therefore the above figure is representative of the total computational power required in the demultiplexing and interpolation functions.

A similar calculation for the cascade FFT processor (Table 2.3) leads to:

- Cascade FFT Processor Power = 137x0.15 = 20.5w

and for the parallel FFT processor (Table 2.3) to :

- Parallel FFT Processor Power = 151x0.15 = 22.6w.

2.2.7 SUMMARY

Three methods for implementing the demultiplexer to accommodate carriers of different bandwidths have been studied. The method which uses
a single large forward FFT processor followed by an IFFT for individual channel selection results in the least computational intensity and power consumption to perform the overall processing for all carriers. This method also has unlimited flexibility for accommodating various arrangements of carrier locations and bandwidths in the input signal band. Because of these desirable properties, it is the preferred method chosen for further consideration. Compared to the power estimate for the FFT-IDFT implementation given in the previous section which required over 200 w any of the three methods using the IFFT discussed in this section consume far less power.

2.3 COMPARISON OF RADIX 2 AND RADIX 4 FFT IMPLEMENTATIONS

2.3.1 GENERAL

This section presents a comparison of the radix 2 and radix 4 pipeline implementations of the FFT. It is concluded that the radix 4 implementation causes an increase in the number of multipliers and adders by factors of 1.5 and 1.83 compared to the radix 2 implementation while reducing the speeds of the individual multiplies and adds by factors of 0.75 and 0.9167. The radix 4 implementation would therefore be of interest only if the speed of multiplication becomes a limiting factor. Otherwise, the radix 2 design would be preferred.

2.3.2 NUMBER OF STAGES

In the implemention of the FFT, the pipeline architecture can be expressed as a cascade of Discrete Fourier Transforms (DFTs) and the lowest order transform that is conceivable is the 2x2 or radix 2 DFT. When the pipeline FFT is implemented using the 2x2 DFT, it is referred to as a Radix 2 FFT. This implementation was previously described in an earlier section. For a sample window containing N samples, the number of radix 2 pipeline stages is given by the expression

$$K_{RADIX\ 2} = \log_2 N$$

In a radix 4 implementation, the DFT processes a 4x4 subset of samples and consequently for a sample window of N samples, the number of radix 4 pipeline stages is given by the expression

$$K_{RADIX\ 4} = \log_4 N = (1/2)\log_2 N$$

Thus the radix 4 implementation halves the number of pipeline stages needed relative to the radix 2 to perform the FFT. A block diagram of a radix 4 pipeline implementation for a 64 sample window is shown in Figure 2.8.

2.3.3 COMPUTATION SPEED

With regard to the speed of computation, each radix 4 stage has twice as long to perform its processing and consequently operates at half the

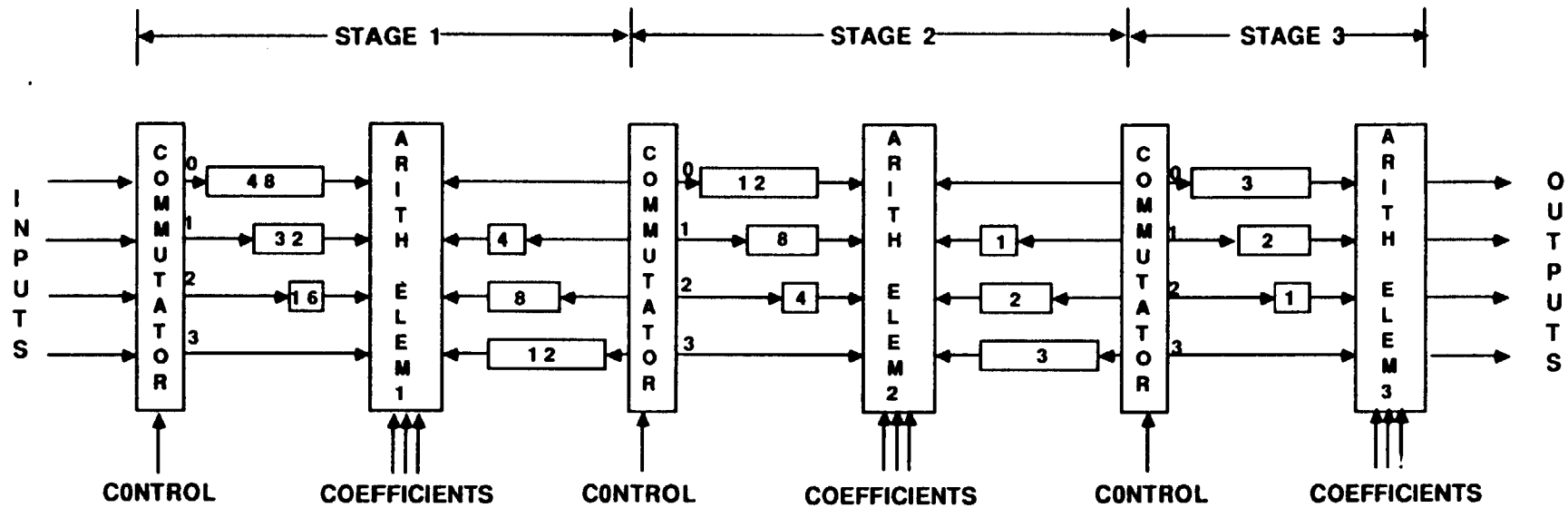


FIGURE 2.8 RADIX 4, 64 POINT, PIPELINE FFT

rate of the radix 2 stage. Since there are one half the number of stages and each operates at one half the rate, the speed of computation is cut to one fourth that of the radix 2 implementation.

2.3.4 NUMBER OF COMPUTATIONS PER STAGE

Diagrams of the radix 2 and radix 4 computational elements (also called butterflies) of each stage are shown in Figure 2.9 for comparison. Each radix 2 butterfly comprises 1 complex multiply and 2 complex adds which in turn require 4 real multipliers and 6 real adders, whereas each radix 4 butterfly comprises 3 complex multipliers and 8 complex adders which in turn require 12 real multipliers and 22 real adders. Thus, the total number of real multipliers and real adders for each radix are

$$\text{No. of RADIX 2 Adders} = 6 \log_2 N$$

$$\text{No. of RADIX 4 Adders} = 11 \log_2 N$$

$$\text{No. of RADIX 2 Multipliers} = 4 \log_2 N$$

$$\text{No. of RADIX 4 Multipliers} = 6 \log_2 N$$

From the above it is seen that the number of adders and multipliers needed for the radix 4 implementation exceed those needed for the radix 2 implementation; however, the influence of speed has yet to be accounted for. The clock speed of the radix 2 design which processes 2 samples at a time is thus 1/2 the sample rate while that of the radix 4 design which processes 4 samples at a time is 1/4 the sampling rate. Consequently the rates of adds and multiplies for the radix 2 and radix 4 implementations assuming a sampling rate of R per second are respectively,

$$\text{RADIX 2 add speed} = 3.0\ R \log_2 N$$

$$\text{RADIX 4 add speed} = 2.75\ R \log_2 N$$

$$\text{RADIX 2 mult speed} = 2.0\ R \log_2 N$$

$$\text{RADIX 4 mult speed} = 1.5\ R \log_2 N$$

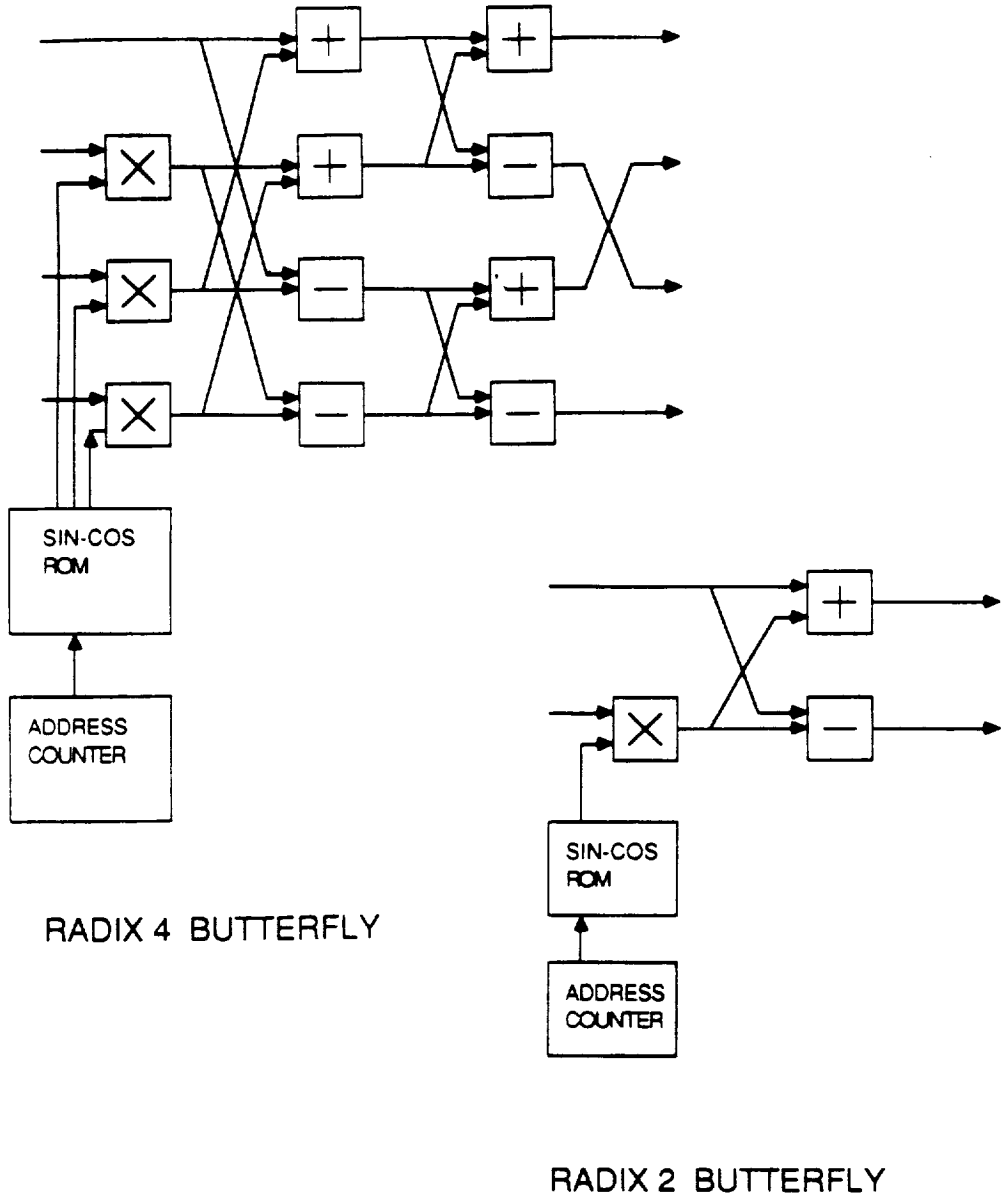


Figure 2.9 Radix 2 and radix 4 butterflies

2.3.5 SUMMARY

From the above discussion comparing the radix 2 and radix 4 pipeline FFT implementations, it can be concluded that:

1. The radix 4 compared to the radix 2 implementation increases the number of multipliers by a factor of 1.5 and the number of adders by a factor of 1.833. This increases the size of the overall processor accordingly.

2. The radix 4 compared to the radix 2 implementation decreases the speed of the multipliers by a factor of 0.75 and that of the adders by a factor of 0.9167.

Use of a radix 4 implementation is of interest if the speed of the multipliers becomes the limiting factor. Otherwise the radix 2 design is preferred.

3.0 RECOVERY OF THE TIME DOMAIN SAMPLES OF SELECTED CHANNELS

3.1 GENERAL

To recover a given carrier from the 40MHz band processed by the input FFT, it is necessary to calculate the product of the FFT coefficients and the coefficients of a channel filter defining the bounds of the wanted channel that are stored in onboard memory. The FFT processing required to obtain the spectrum coefficients of the input multicarrier signal has been discussed in the previous sections. The method used to obtain the coefficients of the channel filter is now discussed and this is followed by a discussion of the processing used to recover the time domain samples needed at the input to the demodulator. The discussion is presented in terms of the recovery of multiple 1.024 Msym/s rate carriers each carrying 2.048 Mbit/s which from the previous discussion requires a 512 point FFT over a 40 MHz spectrum allocation.

To accommplish recovery of the samples, first the forward FFT coefficients must be filtered by a channel filter to select the wanted components and next an interpolation filter must be applied to calculate the properly phased time domain samples needed at the demodulator input. The time domain samples delivered at the output of the IFFT processor are timed relative to the clock that controls the demultiplexer and this clock is established by the wideband signal sampler located at the input to the forward FFT. The time domain samples that are used in the demodulator are established by the need to sample the carrier signal appearing at the input to the demodulator at twice the symbol rate. Furthermore, the phase of the samples must be adjusted according to a phase control signal from the demodulator to align the samples at the proper positions in each symbol. These points will become clear in the discussion of the demodulator which comes in a later section. To accomplish this, a sample interpolator is needed between the demultiplexer and the demodulator.

The discussion concludes with the description of an IFFT method recently identified by Comsat Labs that is still in the process of being developed more fully. This method promises to provide a means for simultaneously performing the IFFTs of a multiplicity of carriers of different sizes in the same pipeline processor. As it is currently, the pipeline processor must be reprogrammed for each different bandwidth processed and simultaneous processing of different bandwidths requires parallel pipeline IFFTs.

3.2 CHANNEL FILTER FREQUENCY COEFFICIENTS

First, a frequency domain transfer function of the baseband channel filter is selected. Typically, this may be a 40% square root Nyquist for the symbol rate selected. The transfer function is sampled in such a way that the 40 MHz band is covered in 256, (1/2 x 512), equally spaced frequency domain points. Most of the samples will be zero because the wanted channel only covers a small fraction of the total spectrum. Next, the inverse transform is performed over the 256 frequency domain points to produce a 256 sample time domain impulse response of the filter. The next step is to add 256 zeros to extend the impulse response to a length of 512 time domain samples and perform a 512 point Fourier transform which results in a 512 sample frequency domain transfer function. This is the frequency domain function which performs the interpolation among the samples needed to satisfy the conditions of the overlap and save method for removing the unwanted aliasing samples of circular convolution. The interpolation process leads to non-zero frequency coefficients outside the desired bandwidth which are small (< - 40 dB) and may be set to zero without introducing significant error. The resulting channel filter function is stored in memory and used to multiply the 512 coefficients of each signal spectrum to recover the frequency domain samples of each carrier channel.

3.3 INVERSE FOURIER TRANSFORM

Following multiplication of the output of the FFT by the channel filter's frequency coefficients (which are stored in RAM) there will be 512 frequency points (only a few of which are non zeros) representing a particular carrier. This process is repeated for all carriers by choosing the part of the FFT spectrum where each carrier is located and multiplying it by the corresponding filter's coefficients. What remains then is to invert those frequency coefficients on a carrier by carrier basis. There are several methods to perform this inverse operation.

a) IDFT Method

The first method consists of computing the desired time samples one at a time using the inverse DFT relationship

$$x(t_i) = \sum x_k e^{-jckt_i}$$

where $x(t_i)$ is the desired time samples at t_i, x_k are the frequency coefficients at $k = 0,1,2, \ldots$ and c is a constant. Only the non zero frequency coefficients need be included in the above sum.The time instants t_i at which samples need to be computed are obtained from the clock synchronizer.

Two samples per symbol are adequate for detection and synchronization. For detection, the samples should be at the middle of the symbols (maximum eye openings), these are assumed to be the even samples. To maintain synchronization an additional set of samples is needed at the zero crossings when symbol transitions occur (minimum eye openings), these are assumed to be the odd samples. Therefore, the time instants at which samples should be computed are separated by half a symbol duration. Clock adjustment is performed by an acquisition and tracking procedure described in the section on demodulation. Within each block, the time domain samples in the first half of the block should be discarded as dictated by the overlap and save technique. This is because this first half suffers from the aliasing arising from the circular convolution.

The advantage of doing the inversion one sample at a time as described above is that only the samples that are needed are computed. Thus the aliased samples are not computed at all. The number of multiplications per output sample increases linearly however as the carrier size (number of non zero frequency coefficients) increases.

b) IFFT Method (Non Power Of Two)

In contrast, if an inverse FFT (IFFT) operation is performed on the set of non zero frequency coefficients, the increase would be logarithmic, which is slower. This leads to a second approach for inverting the frequency coefficients. As we mentioned above the time samples required are separated at half a symbol intervals. To obtain precisely these samples at the output of the IFFT would require that the frequency coefficients used in the transform span a frequency range exactly equal to twice the inverse of a symbol duration. In general, this will imply a noninteger number of frequency points since the frequency resolution and the inverse of a symbol duration are not simply related. Although the error resulting from rounding to the nearest integer may be acceptable, the size of the resulting IFFT will not in general be a power of two. Algorithms for non power of two Fourier transforms exist and could be used. Powers of two Fourier transforms are preferred; however, because

they have a simpler control structure.

c) IFFT Method (Power Of Two)

The third approach that is now presented uses powers of two Fourier transforms. In this third approach of inverting the frequency coefficients to recover the time domain samples, an IFFT whose size is a power of two is used. The chosen power of two is the smallest power of two that is larger than the number of non zero frequency coefficients. (Later in this section, we shall discuss how several IFFTs of different sizes can be implemented simultaneously in a single pipeline.) As shown in Figure 3.1, the samples at the IFFT output will not correspond to the desired even and odd samples and therefore an interpolation process will be required. The interpolation filter must be chosen such that the combined filtering of the demultiplexing filter and the interpolation filter approximate the desired square root Nyquist response.

3.4 CHOICE OF THE SAMPLE INTERPOLATION FILTER.

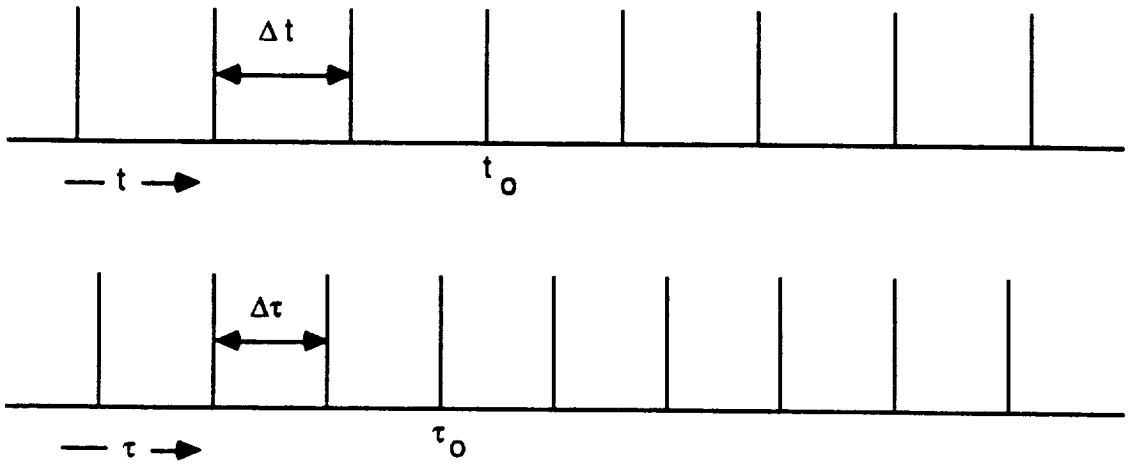
The interpolation filter is used to weigh the samples generated at the IFFT output to determine the properly phased samples needed at the input of the demodulator. Its coefficients must be chosen jointly with those of the channel filter

Three different ways for choosing these filters are shown in Figure 3.2 and discussed below.

Case A represents use of the desired square root Nyquist at the demultiplexer output and a brick wall filter for the interpolation. This is not a good choice due to the difficulty (large computational requirements) in implementing a brick wall filter which theoretically has an infinte impulse and is hence impractical.

Case B shows the square root characteristics equally divided among the demultiplexing and the interpolating filter. This approach is preferred to A but is still not very attractive because of the sharp characteristics of the fourth root Nyquist function which results in a very long impulse response.

In Case C, a square root Nyquist filter is used at the demultiplexer as



$\Delta\tau$, Δt : symbol interval
16 intervals typical

x : SAMPLES AT OUTPUT OF IFFT

y : DESIRED EVEN & ODD SAMPLES

$$y(\tau_o) = x(t_o)h(\tau_o - t_o) + x(t_o + \Delta t)h(\tau_o - t_o - \Delta t) \cdots\cdots\cdots$$
$$x(t_o - \Delta t)h(\tau_o - t_o + \Delta t) \cdots\cdots\cdots\cdots$$

256 VALUES ARE STORED FOR $h(t_i)$

FIGURE 3.1 INTERPOLATOR

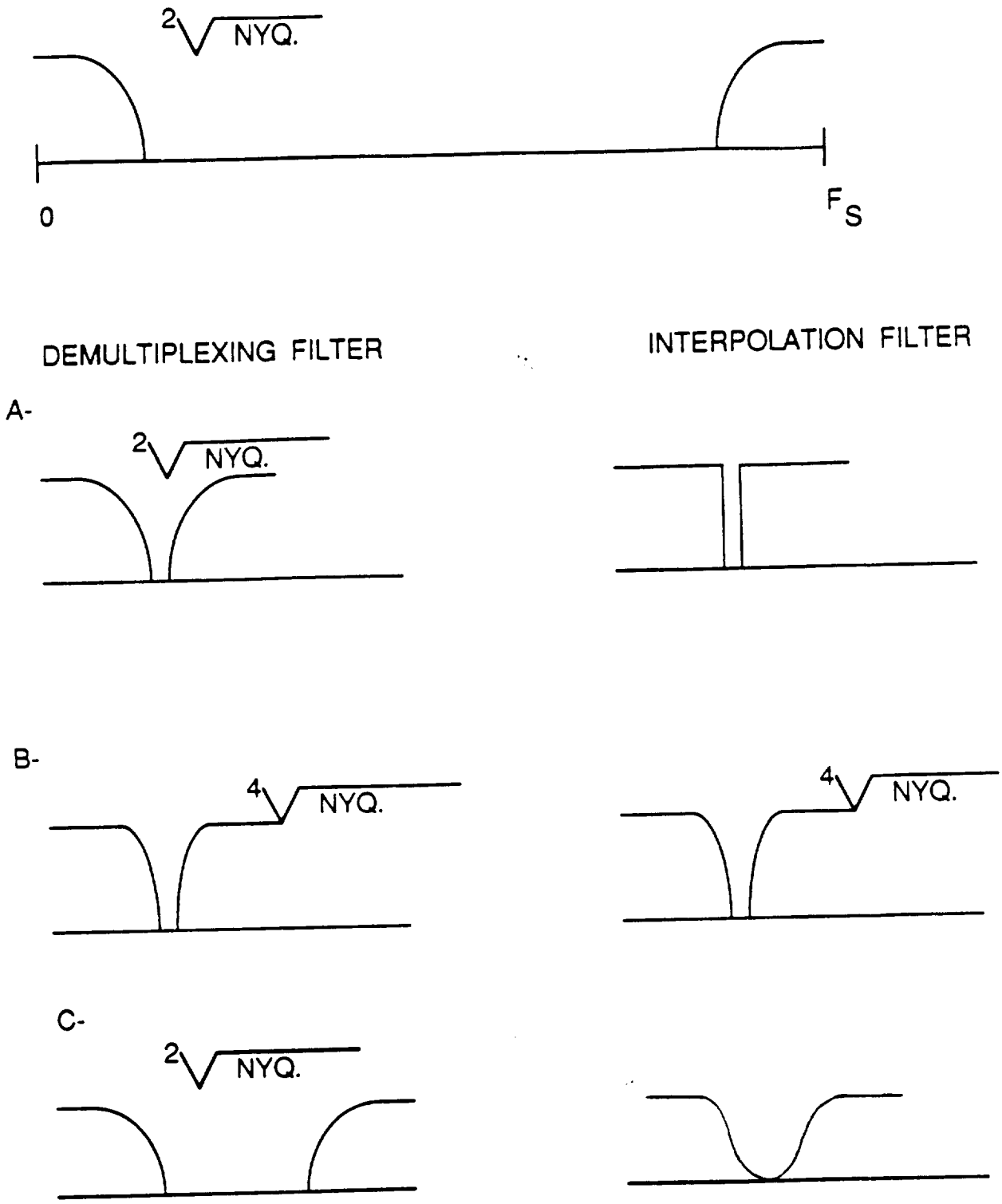


FIGURE 3.2 CHOICE OF INTERPOLATION FILTERS

in Case A. However, a larger size inverse FFT is used. The effect of this, as shown in Figure 3.2, is that the interpolation filter characteristics can now be flat over the range of frequencies where the demultiplexer filter response is non zero and have a smooth transition to zero over the range of frequencies where the demultiplexer filter response is zero. Doing so simplifies the interpolation process considerably. Indeed, simulation results show that only a few (at most 16) samples need be used in the computation of any desired interpolated sample. The impulse response coefficients of the interpolating filter would be stored in memory. The number of coefficients to be stored depends on two factors. The first one is the number of symbols over which the impulse response is non zero. As mentioned above, this number is minimized by choosing a smooth frequency characteristic. The second factor is the accuracy needed in subdividing a symbol interval. Simulation results show that having 32 samples per symbol interval, i.e., being able to compute the sample value at any of 32 equally spaced locations within a symbol interval, is quite adequate. This would correspond to storing no more that 256 coefficients of the impulse response.

3.5 LINEAR AND CIRCULAR INTERPOLATION

3.5.1 LINEAR INTERPOLATION.

The first option which is called linear interpolation consists of the following steps illustrated in Figure 3.3:

1. At the output of each IFFT frame, select the time domain samples corresponding to the carrier under consideration.

2. For each IFFT frame, discard the first half of the samples corresponding to the carrier under consideration.

3. Juxtapose the second half from frame N to the second half from frame N-1 and so on to form a continuous stream of samples.

4. Use this stream as the input to the interpolating filter and compute the output interpolated samples at the time instants indicated by the clock synchronize output.

3.5.2 CIRCULAR INTERPOLATION.

The second option which is called circular interpolation consists of

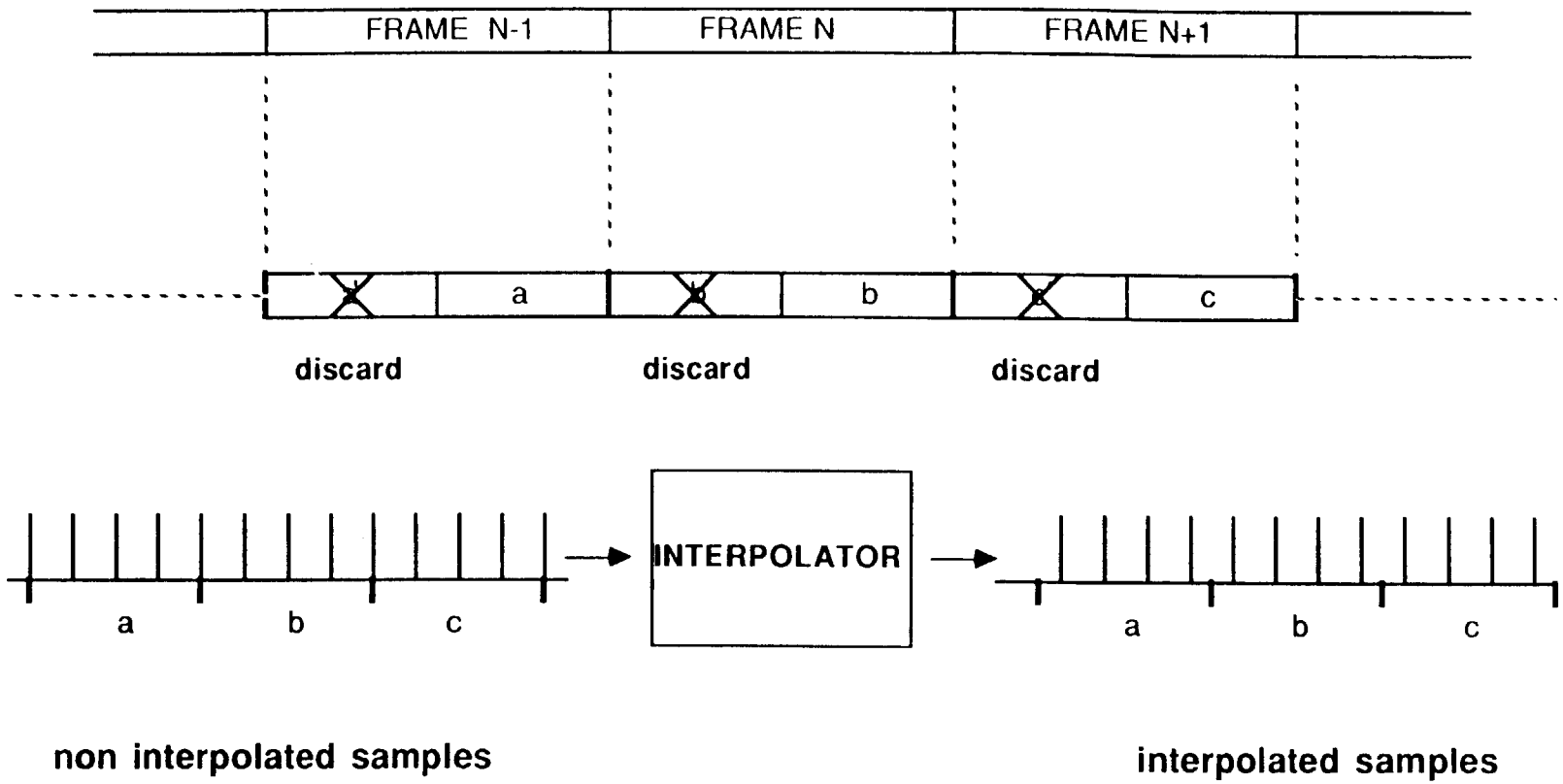


FIGURE 3.3. LINEAR INTERPOLATION

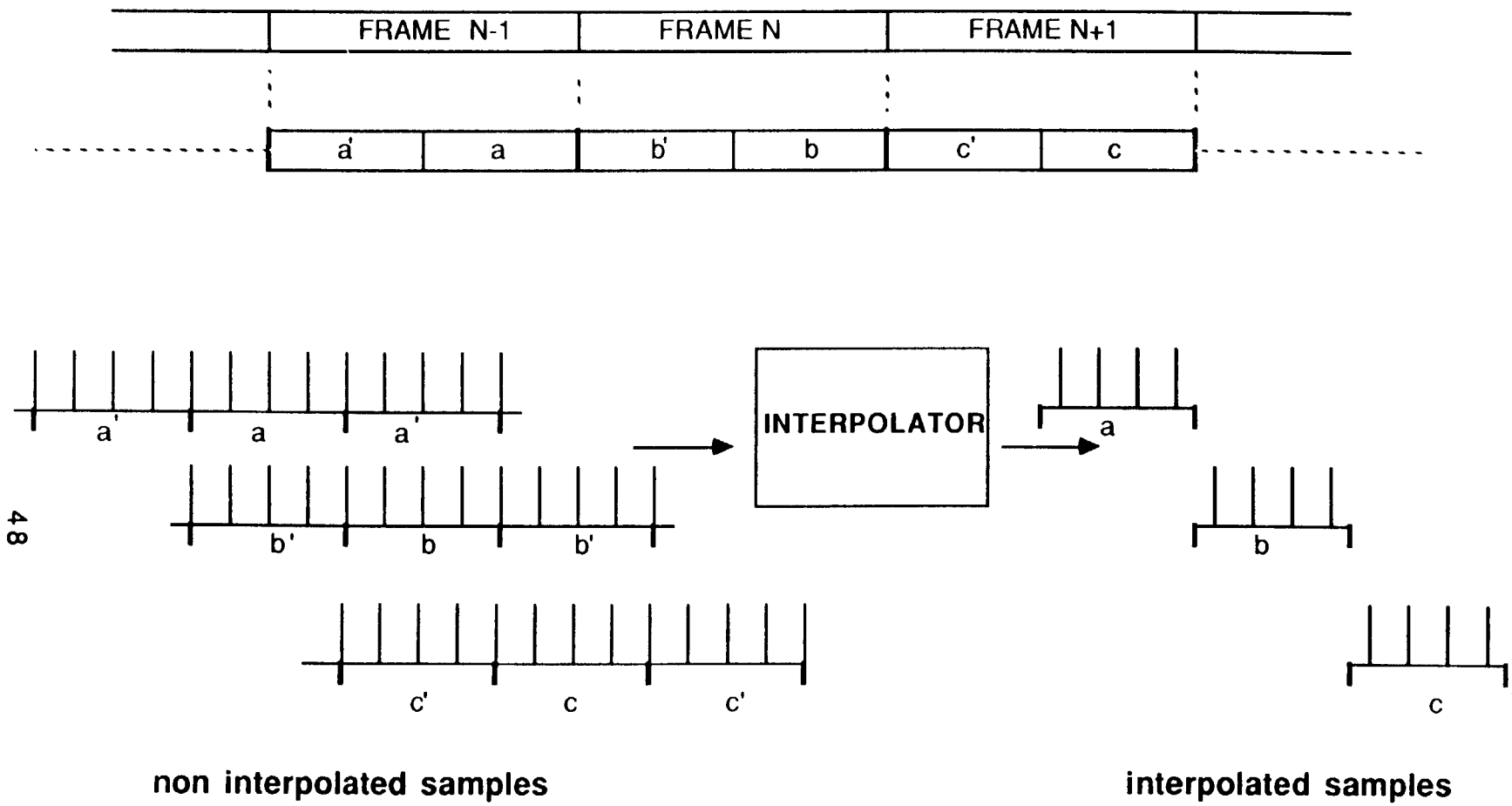


FIGURE 3.4. CIRCULAR INTERPOLATION

the following steps which are illustrated in Figure 3.4:

1. At the output of each IFFT frame, select the samples corresponding to the carrier under consideration.

2. Arrange the samples corresponding to the carrier of interest at the output of each IFFT frame in a circular manner (i.e. as if they constituted one period of a periodic signal). This is simply implemented by numbering the samples 0,l,2,....N-1 and using a module N operation. Thus sample N would be sample 0, sample N+1would be 1 and so on.

3, Use these samples as the input to the interpolating filter and sample the output samples at the indicated time instants.

The circular approach to interpolation is preferred to the linear approach because each frame is processed independently of the previous ones leading to a simpler implementation with less storage requirements.
IFFTs Of Different Sizes In The Same Pipeline Processor

3.6 IFFTs OF DIFFERENT SIZES IN THE SAME PIPELINE PROCESSOR

Several IFFTs of different sizes can be implemented simultaneously in a single pipeline. At every clock pulse, r samples are presented to the butterfly computational elements. The twiddle factors (phase shifts) used with the butterfly operations will depend on the FFT size, the stage within the pipe, and the index of the input samples. Those twiddle factors are precomputed and stored in memory. At every clock pulse, a new factor may be used, thus accommodating a variety of FFT sizes. Of course, the interstage reordering will have to properly match the samples before presenting them to the next butterfly element. These interstage reordering modules consist of delays and commutators as mentioned previously. The amount of delay at a given stage in the pipeline is determined by the stage number. However, more flexibility in the commutator action is needed to implement different output/input matching at every clock pulse. The commutator action can be greatly simplified by properly sequencing the different frequency coefficients of the different carriers. Detailed circuit designs and timing diagrams for such an implementation are being developed under corporate sponsorship at COMSAT LABS.

4.0 DIGITAL DEMODULATION

4.1 OVERVIEW

This section describes a digital signal processing method for demodulating the individual carrier signals that are demultiplexed by a combination of FFT and IFFT processing. The signals are presented to the demodulator in the form of discrete time domain samples at a rate of two samples on each of two quadrature channels for each symbol interval. These samples are processed to recover the modulated data bits. To accomplish this, it is necessary to acquire and maintain both symbol timing and carrier frequency synchronization. A single processor is shared to demodulate all of the carriers.

A block diagram of the demodulation processor is shown in Figure 4.1. Two samples per symbol, X_k and Y_k are derived at the output of a sample interpolator at a rate of two per symbol and controlled in phase by a timing estimate $s\hat{}_n$ that maintains an alignment such that one sample occurs at the center of each symbol and the other at each symbol boundary. This process also compensates for the slip between the FFT/IFFT and demodulator clocks. The sampling interpolator establishes the proper sample phase as the final step in the IFFT processing.

Symbol timing acquisition and synchronization are performed by the processors contained in the loop shown at the top of Figure 4.1. The acquisition process calculates an initial estimate of the timing phase error, $\varepsilon\hat{}_0$, during the preamble segment of each received TDMA burst. This is used to initialize an accumulator in the symbol synchronizer at the start of the traffic portion of the burst. The symbol synchronizer maintains the timing error to a value near zero during the traffic portion of the burst by appropriately adjusting the value of $s\hat{}_n$. For continuous carriers the acquisition function may be replaced by a timing search procedure.

Carrier acquisition and synchronization are performed by the processors shown in the lower half of the Figure 4.1. The acquisition processor calculates initial estimates of carrier phase, $\theta\hat{}_0$, and phase rate, $\dot{\theta}\hat{}_0$, (carrier frequency offset) during the preamble. These are used to initialize the carrier synchronizer which maintains the

QPSK DEMODULATOR

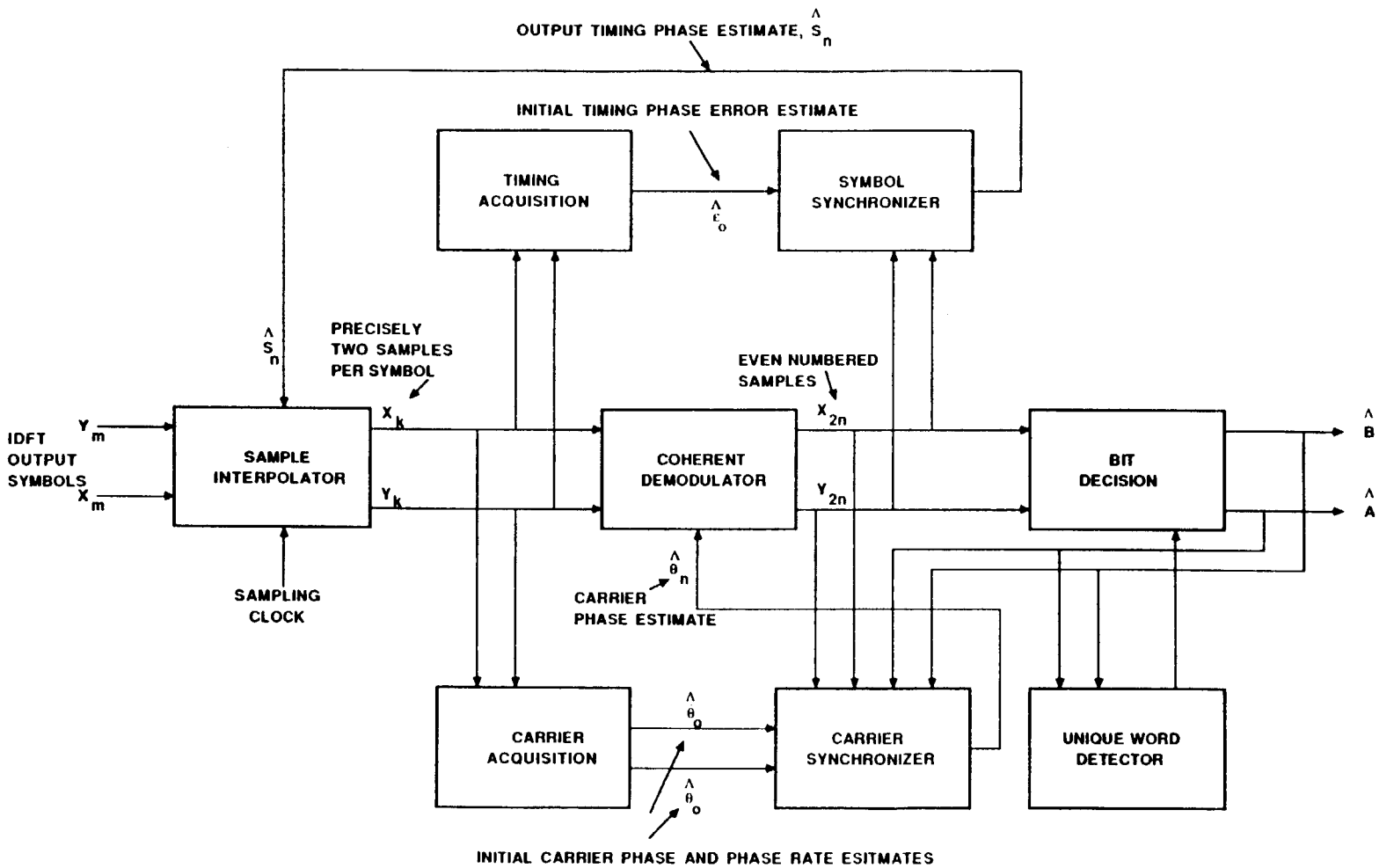


FIGURE 4.1. QPSK DEMODULATOR IMPLEMENTATION

synchronization during the traffic portion of the burst. The output of the coherent demodulator consists of the samples taken at the center of each symbol which are designated as the even numbered samples and those taken at symbol boundaries which are designated as the odd numbered samples. When symbol timing and carrier synchronization are properly maintained, the even numbered samples are taken at the optimum time (at mid symbol) to cancel out intersymbol interference an consequently provide the best possible sample values for making the bit decisions. The odd numbered samples occur at the boundaries between symbols and are consequently nearly zero when symbol transitions occur and at an absolute maximum when no transitions occur. Only the even numbered samples are used by bit decision processor to determine the estimated bit outputs A^ and B^. Decision directed feedback from the output of the bit decision processor is used in the carrier synchronization processing to aid in the calculation of the carrier phase error.

Detailed descriptions of the various processing steps for the acquisition and synchronization phases for symbol timing and carrier recovery are given in the following.

4.2 ACQUISITION PROCESSING

4.2.1 PREAMBLE STRUCTURE.

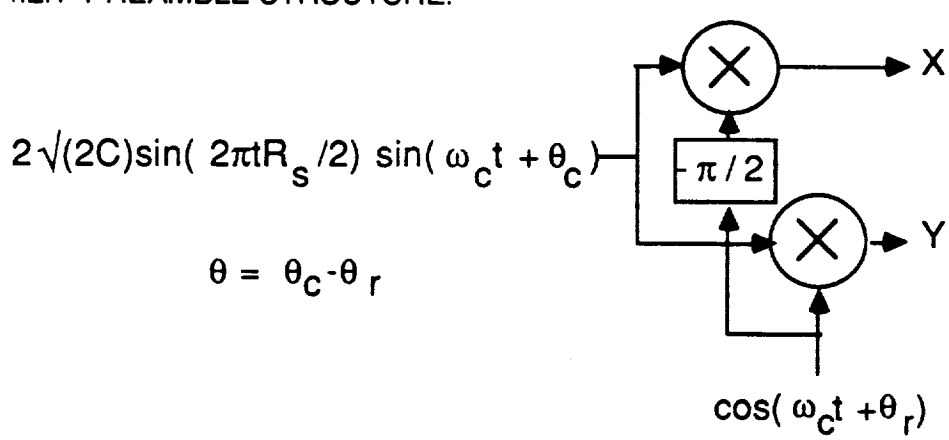


FIGURE 4.2. CONVERSION TO BASEBAND

Let the preamble be represented by a BPSK modulated carrier of power C having quadrature baseband components:

$$X = \sqrt{(2C)}\ \sin(\ 2\pi t R_s/2\ )\ \cos\theta \tag{1a}$$

$$Y = \sqrt{(2C)}\ \sin(\ 2\pi t R_s/2\ )\ \sin\theta \tag{1b}$$

where R_s is the symbol rate, t is time, θ is the phase offset between the signal carrier and the recovered carrier. For digital signal processing, the continuous function must be represented in discrete sampled data form. The Nyquist sampling theory shows that each symbol of each quadrature phase must be sampled twice. Samples are taken at times $t_k+\Delta t$ where Δt is the time error in locating the desired sampling instant. Hence, the sampled data form can be expressed as:

$$X_k = \sqrt{(2C)}\ \sin(\ 2\pi t_k R_s/2 + \varepsilon/2\ )\ \cos\theta \tag{2a}$$

$$Y_k = \sqrt{(2C)}\ \sin(\ 2\pi t_k R_s/2 + \varepsilon/2\ )\ \sin\theta \tag{2b}$$

where $\varepsilon/2$ is the phase displacement between the signal symbol period and the sampling clock period. If the time error is Δt, then $\varepsilon = 2\pi R_s \Delta t$. The 2 samples per symbol are classified into those taken at even and those at odd numbered sampling times. If the timing error $\Delta t = 0$, the even numbered sampling times are at mid symbol and the odd numbered at symbol boundaries. For the nth symbol, the even numbered samples are taken at times t_{2n} and the odd at times t_{2n-1},where

$$t_{2n} = (n + 1/2)T_s \tag{3a}$$

$$t_{2n-1} = nT_s \tag{3b}$$

$$T_s = 1/R_s \tag{3c}$$

Consequently, the sampled values for odd and even numbered sampling times during the preamble are:

$$Y_{odd} = Y_o = \sqrt{(2C)}\,(-1)^n \sin(\varepsilon/2)\,\sin\theta \tag{4a}$$

$$X_{odd} = X_o = \sqrt{(2C)}\,(-1)^n \sin(\varepsilon/2)\,\cos\theta \tag{4b}$$

$$Y_{even} = Y_e = \sqrt{(2C)}\,(-1)^n \cos(\varepsilon/2)\,\sin\theta \tag{4c}$$

$$X_{even} = X_e = \sqrt{(2C)}\,(-1)^n \cos(\varepsilon/2)\,\cos\theta \tag{4d}$$

The values of X_k and Y_k taken at times $t_k = k/2R_s$ are:

For X_k

$$\begin{array}{ll} -\sqrt{2C}\ \cos\theta\,\sin(\varepsilon/2) + \text{noise} & k=1,n=1 \\ -\sqrt{2C}\ \cos\theta\,\cos(\varepsilon/2) + \text{noise} & k=2,n=1 \\ \sqrt{2C}\ \cos\theta\,\sin(\varepsilon/2) + \text{noise} & k=3,n=2 \\ \sqrt{2C}\ \cos\theta\,\cos(\varepsilon/2) + \text{noise} & k=4,n=2 \\ \vdots & \\ \text{repeats every 4 samples} & \end{array} \tag{5}$$

For Y_k

$$\begin{array}{ll} -\sqrt{2C}\ \sin\theta\,\sin(\varepsilon/2) + \text{noise} & k=1,n=1 \\ -\sqrt{2C}\ \sin\theta\,\cos(\varepsilon/2) + \text{noise} & k=2,n=1 \\ \sqrt{2C}\ \sin\theta\,\sin(\varepsilon/2) + \text{noise} & k=3,n=2 \\ \sqrt{2C}\ \sin\theta\,\cos(\varepsilon/2) + \text{noise} & k=4,n=2 \\ \vdots & \\ \text{repeats every 4 samples} & \end{array} \tag{6}$$

Thus only 8 different values actually occur which are repeated every two symbols. This result is evident in the sampled preamble shown in Figure 4.5. From these samples we wish to estimate values of θ and ε.

The sampled values of the preamble signal given above can be combined into the following relationships among the even and odd numbered samples:

$$\text{SUM } X_{odd} = \Sigma_{Xo} = X_1 - X_3 + X_5 - X_7 \cdots \tag{7a}$$

$$\text{SUM } X_{even} = \Sigma_{Xe} = X_2 - X_4 + X_6 - X_8 \cdots \tag{7b}$$

$$\text{SUM } Y_{odd} = \Sigma_{Yo} = Y_1 - Y_3 + Y_5 - Y_7 \cdots \tag{7c}$$

$$\text{SUM } Y_{even} = \Sigma_{Ye} = Y_2 - Y_4 + Y_6 - Y_8 \cdots \tag{7d}$$

substituting the sample values given in equations (1) and (2), recognizing that they repeat in sets of four

$$\Sigma_{Xo} = -\sqrt{(2C)}\, N \cos\theta \sin(\varepsilon/2) \tag{8a}$$

$$\Sigma_{Xe} = -\sqrt{(2C)}\, N \cos\theta \cos(\varepsilon/2) \tag{8b}$$

$$\Sigma_{Yo} = -\sqrt{(2C)}\, N \sin\theta \sin(\varepsilon/2) \tag{8c}$$

$$\Sigma_{Ye} = -\sqrt{(2C)}\, N \sin\theta \cos(\varepsilon/2) \tag{8d}$$

From these relationships, expressions can be written for carrier and symbol timing acquisition.

4.2.2 CARRIER ACQUISITION (DETERMINATION OF θ AND dθ/dt)

4.2.2.1 Determination of θ.

From the expressions previously given for Σ_{Yo} and Σ_{Ye}, the following can be written:

$$\Sigma_{Yo}^2 + \Sigma_{Ye}^2 = C N^2 \sin^2\theta \tag{9a}$$

$$\Sigma_{Xo}^2 + \Sigma_{Xe}^2 = C N^2 \cos^2\theta \tag{9b}$$

$$\therefore \tan^2\theta = (\Sigma_{Yo}^2 + \Sigma_{Ye}^2)/(\Sigma_{Xo}^2 + \Sigma_{Xe}^2) \tag{9c}$$

The value of θ determined from the above expression is limited to the first quadrant consequently resulting in a four fold ambiguity. This can be reduced to a two fold ambiguity by examining the sign of the expression

$$\Sigma_{XY1} = \Sigma_{Xe}\,\Sigma_{Ye} + \Sigma_{Xo}\,\Sigma_{Yo} \tag{10}$$

For noiseless conditions, equation (8) shows that $\Sigma_{Ye} = K\Sigma_{Xe}$ and $\Sigma_{Yo} = K\Sigma_{Xo}$. Substituting these last expressions into equation (10) the result is

$$\Sigma_{XY1} = K[(\Sigma_{Xe})^2 + (\Sigma_{Xo})^2] \tag{11}$$

Thus the sign of the function Σ_{XY1} is the same as that of K and determines the value of K as being either greater than or less than zero. If K is greater than zero the angle is in the first or third quadrant and if it is negative the angle is in the second or fourth quadrant.

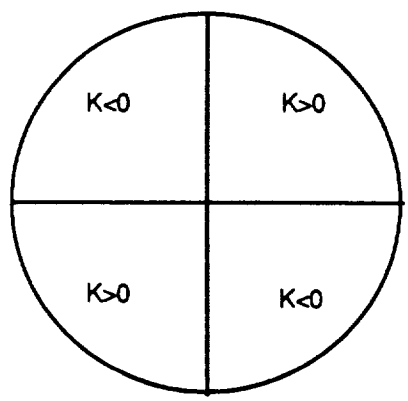


FIGURE 4.3. AMBIGUITY REMOVAL DIAGRAM FOR CARRIER PHASE.

4.2.2.2 Determination of $\dot{\theta}^{\wedge} = E(d\theta/dt)$.

To determine the rate of change of carrier phase (this corresponds to determination of the frequency offset between the carrier and the local reference applied during the acquisition phase), the preamble is divided into halves and the value of θ calculated separately in each half. Let the

value in the first half be θ_1 and in the second half θ_2. The calculation

process inherently determines each estimated value at the center of each half. Hence the estimated value of the derivative is

$$\dot{\theta}^{\wedge} = E(d\theta/dt) = 4(\theta_2 - \theta_1)R_s/N \tag{12}$$

and the estimated value of the phase angle at the end of the preamble is

$$\theta^{\wedge} = E(\theta) = (\theta_2 + \theta_1)/2 + (\theta_2 - \theta_1) = (3\theta_2 - \theta_1)/2 \tag{13}$$

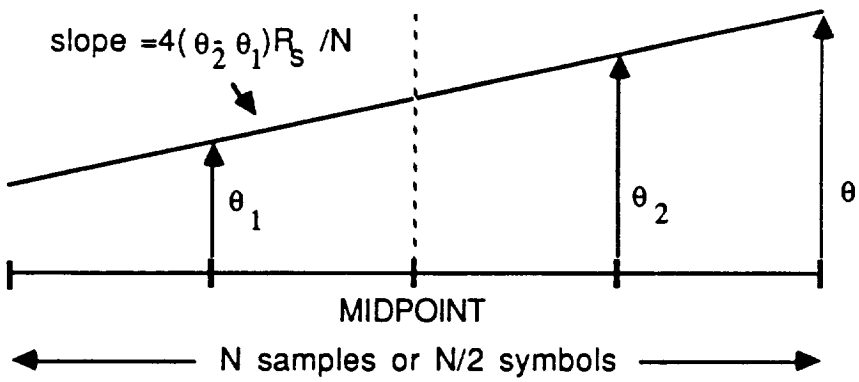


FIGURE 4.4. DETERMINATION OF INITIAL CARRIER PHASE AND PHASE RATE

4.2.3 CLOCK ACQUISITION (DETERMINATION OF $\varepsilon^{\wedge} = E(\varepsilon)$ and $\dot{\varepsilon}^{\wedge} = E(d\varepsilon/dt)$.

4.2.3.1 Determination of ε.

$$\Sigma_{Xo}^2 + \Sigma_{Yo}^2 = (CN^2/4)\sin^2\varepsilon/2$$

$$\Sigma_{Xe}^2 + \Sigma_{Ye}^2 = (CN^2/4)\cos^2\varepsilon/2$$

$$\therefore \quad \tan^2\varepsilon^{\wedge}/2 = (\Sigma_{Xo}^2 + \Sigma_{Yo}^2)/(\Sigma_{Xe}^2 + \Sigma_{Ye}^2) \tag{14}$$

The value of ε determined from the above expression is limited to the first quadrant. The symbol phase has a two fold ambiguity, lying either in the range $(0 < \varepsilon/2 < \pi/2)$ or $(-\pi/2 < \varepsilon/2 < 0)$ which correspond to the ranges $(0 < \varepsilon < \pi)$ or $(-\pi < \varepsilon < 0)$ respectively. This ambiguity can be eliminated by

examining the sign of the expression:

$$\Sigma_{XY2} = \Sigma_{Xe}\,\Sigma_{Ye} + \Sigma_{Xo}\,\Sigma_{Yo} \tag{11}$$

If Σ_{XY2} is greater than zero, the angle ε is in the interval $(0 < \varepsilon < \pi)$ and if less than zero it is in the interval $(-\pi < \varepsilon < 0)$.

An estimate of the derivative $\varepsilon^{\bullet\wedge}$ is obtained by the same method used previously to estimate $\theta^{\bullet\wedge}$, i.e., by dividing the preamble into halves, calculating an estimate for ε in each half and dividing the difference by half the duration of the preamble. This leads to the following expressions for the expected values of ε and $d\varepsilon/dt$ extrapolated to the end of the preamble.

$$\varepsilon^{\bullet\wedge} = E(d\varepsilon/dt) = 4(\varepsilon_2 - \varepsilon_1)R_s/N \tag{15}$$

and the estimated value of the phase angle at the end of the preamble is

$$\varepsilon^{\wedge} = E(\varepsilon) = (\varepsilon_2 + \varepsilon_1)/2 + \varepsilon_2 - \varepsilon_1 = (3\varepsilon_2 - \varepsilon_1)/2 \tag{16}$$

4.2.4 INITIALIZATION OF THE TRACKING PROCESSING.

As a result of the acquisition processing just described, the estimated values of the recovered carrier phase offset $\theta^{\wedge}$, carrier frequency offset $\theta^{\bullet\wedge}$, symbol timing phase offset $\varepsilon^{\wedge}$ and the symbol frequency offset $\varepsilon^{\bullet\wedge}$ have been determined at the instant marking the end of the preamble and the beginning of the traffic. Although an expression has been derived above for $\varepsilon^{\bullet\wedge}$, it is not used in the symbol tracking processing. These values are installed as the initial values in the tracking process. This causes the carrier phase to be established with a two fold ambiguity still to be resolved by examination of the polarities of quadrature modulated UWs*, and the carrier frequency, symbol phase and symbol frequency to be established within the margin of error determined by the noise conditions. The resulting symbol phase adjustment will be

* The UW actually can resolve four phase ambiguity if needed.

such as to cause the even numbered samples to occur at the center of the symbol period and the odd numbered samples at the symbol period boundaries as illustrated in Figure 4.5.

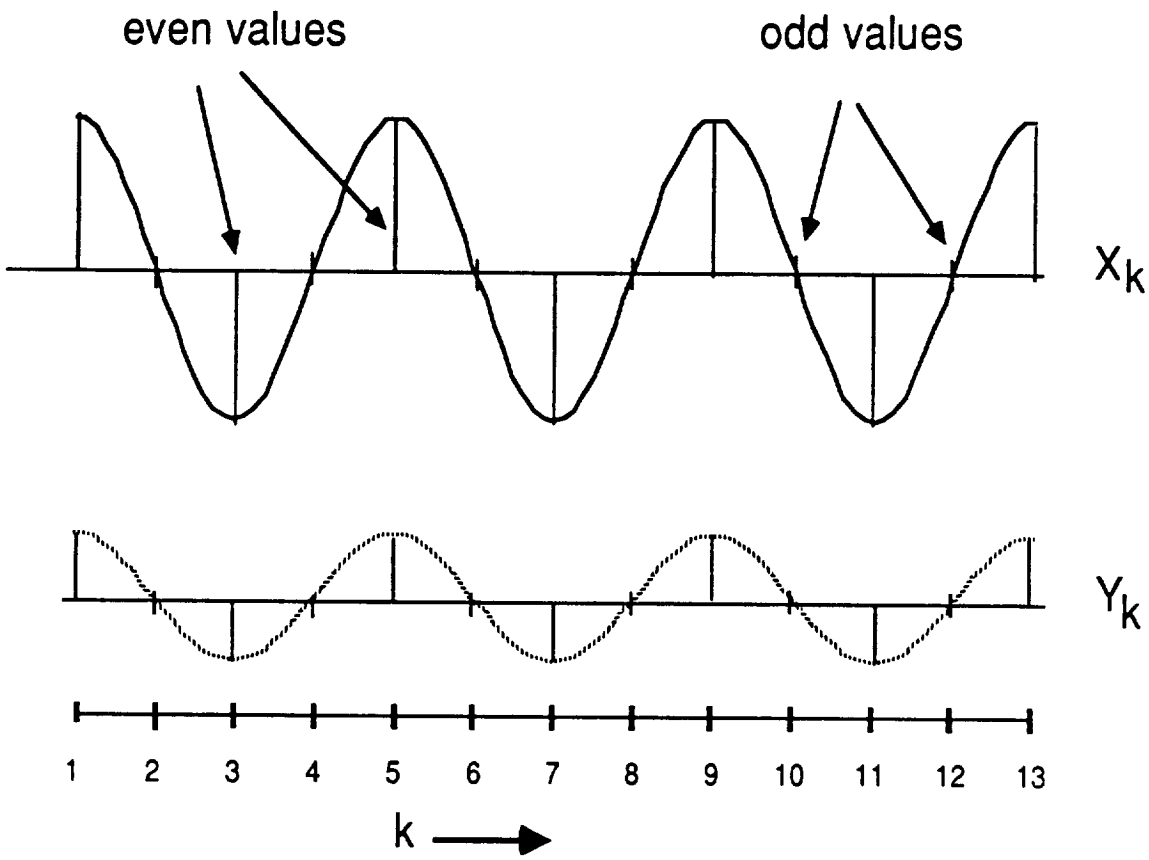


FIGURE 4.5. LOCATIONS OF ODD AND EVEN NUMBERED SAMPLES RELATIVE TO THE PREAMBLE SYMBOL SIGNAL.

4.3 SYNCHRONIZATION - TRACKING PROCESSING.

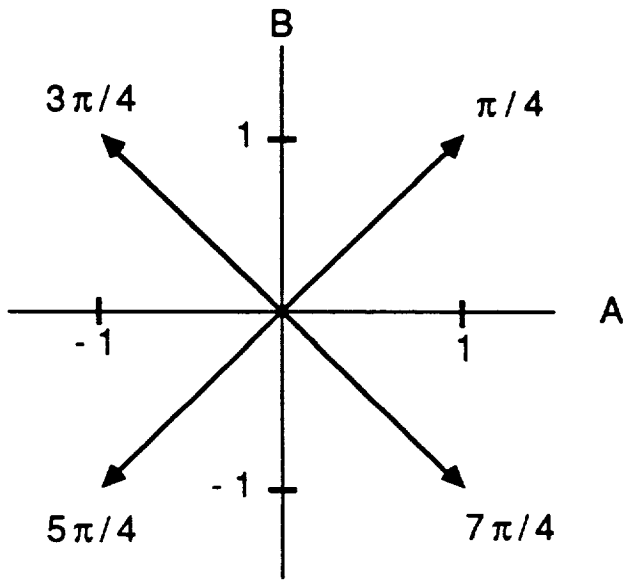
4.3.1 QPSK MODULATED SIGNAL REPRESENTATION.

The QPSK signal can be represented by the relationship

$$Q(t) = \sqrt{(2C)} \cos(\omega_c t + \theta_c - \lambda) \qquad (17)$$

where C is the carrier power, ω_c the carrier frequency, θ_c the carrier phase and λ the modulation angle. Depending on the modulating information, the angle λ can assume the values $\pi/4$, $3\pi/4$, $5\pi/4$ or $7\pi/4$.

These angles result from the assumption that the modulated signal is the sum of two quadrature signals, A cos $\omega_c t$ and B sin $\omega_c t$ where A represents the bits of the message transmitted on the X channel and B the bits transmitted on the Y channel. A and B take on the values ± 1 to represent a zero or a one. The resulting signal phases are shown in Figure 4.6.



λ	A	B
π/4	1	1
3π/4	-1	1
5π/4	-1	-1
7π/4	1	-1

FIGURE 4.6. QPSK CARRIER MODULATION PHASES

In terms of the modulation angle λ it is evident that A and B can be expressed as

$$A = \sqrt{2} \cos \lambda \tag{18a}$$

$$B = \sqrt{2} \sin \lambda \tag{18b}$$

Consequently, the relation for the modulated signal can be expressed as

$$Q(t) = \sqrt{C} \, [A \cos (\omega_c t + \theta_c) + B \sin (\omega_c t + \theta_c)] \tag{19}$$

This signal is quadrature demodulated by multiplying it by cos ($\omega_c t + \theta_r$) to recover the X channel and sin ($\omega_c t + \theta_r$) to recover the Y channel and recovering the low passed difference frequency components. The recovered low passed signal can be expressed as a vector:

$$Z = \sqrt{(2C)}\, e^{j(\theta+\lambda)} = X + jY \tag{20}$$

where

$$X = \sqrt{(2C)} \cos(\theta+\lambda) = \sqrt{C}\,[A\cos\theta - B\sin\theta] \tag{21a}$$

$$Y = \sqrt{(2C)} \sin(\theta+\lambda) = \sqrt{C}\,[B\cos\theta + A\sin\theta] \tag{21b}$$

The expressions given above are for continuous representation. For digital demodulation implementation, it is necessary that the signal be represented in discrete sampled data form. To represent the quadrature modulated information content, it is sufficient that each of the two phases be sampled twice during each symbol interval of duration $T_S = 1/R_S$ with the samples equally spaced. For optimum recovery of the information assuming Nyquist filtering, one sample should occur at mid symbol and the other the end of the symbol where the transition to the next symbol occurs. Sampling at mid symbol is optimum with Nyquist filtering because at the instant of sampling all intersymbol interference contributions are theoretically zero and in the practical case certainly nulled.

During the preamble, A = B and the modulation is a binary alternating sequence of +1 and -1 values. Hence, transitions of π radians occur at each symbol boundary. When the resulting signal is quadrature demodulated to a lowpass band of width slightly greater than $R_S/2$, the resulting quadrature signal appears to be a sinusoid of frequency $R_S/2$. This feature has already been discussed in the section devoted to acquisition processing.

If sampling takes place with a timing offset of Δt relative to the symbols, the corresponding phase offset at the frequency of the symbol rate R_S is $\varepsilon = 2\pi\Delta t/T_S = 2\pi\Delta t R_S$. Assuming that when bit transitions occur, i.e. $B_n = -B_{n-1}$, the low passed transition signal is approximated by sinusoid shaped pulses of half period T_S, the sampled values are given by:

$$X_k = \sqrt{(2C)}\ \sin(\ \pi R_s t_k + \varepsilon/2\ )\ \sin(\ \theta + \pi/4) \tag{22a}$$

$$Y_k = \sqrt{(2C)}\ \sin(\ \pi R_s t_k + \varepsilon/2\ )\ \cos(\ \theta + \pi/4) \tag{22b}$$

Samples can be classified as even or odd depending on sampling times expressed as follows for the nth symbol.

For even sample times: $t_{2n} = (n+1/2)T_s$ (23a)

For odd sample times: $t_{2n-1} = nT_s$ (23b)

4.3.2 TRACKING OF SYMBOL TIMING AND CARRIER FREQUENCY.

4.3.2.1 Symbol Timing Tracking.

As a consequence of the acquisition process, the offset between the symbols of the modulated signal and the sampling times is determined and at the end of the preamble is administered as the initial value to start the tracking process. During tracking, this offset is maintained such that the odd numbered samples are at the locations of transitions and the even numbered samples at the center of the symbol period. This is accomplished by using the output $Z = X + jY$ from the coherent demodulator. Since the timing error is very small, even numbered sample values are not greatly affected by small timing error; however the odd numbered sample values are approximately proportional to the small timing error.

From the previous discussion regarding the preamble signal, it was demonstrated that the baseband signal recovered from a BPSK carrier is a sinusoid of period $2T_s$ and in the vicinity of the axis crossings which mark the transitions from one symbol to the next, the value of the odd numbered sample is $\pm\sqrt{(2C)}\ \sin \varepsilon/2$ where the sign depends on whether the transition is positive or negative. In the traffic portion of the QPSK modulated TDMA carrier burst, the signal occurring on either quadrature channel will be the result of modulation by binary signals that reverse phase arbitrarily at symbol boundaries depending on information content. When there is no change in the modulation value, the odd samples will

yield about the same value as the even samples. A modulation value change (known as a symbol transition) causes a zero crossing in the vicinity of the odd sampling time. Modulation changes occur when $A_n = -A_{n-1}$ or $B_n = -B_{n-1}$ (a phase reversal transition occurs on either or both channels). The situation is illustrated in the Figure 4.7. It can be assumed that the signal function during the transition is a sinusoid similar to that for transitions experienced for simple BPSK. Hence the expressions for the signals on the X and Y channels for such transitions are:

$$Y(t) = [(B_n - B_{n-1})/2][\sqrt{C} \sin(\pi R_s t)]\,[u(t-nT_s) - u(t-(n+1)T_s)] \tag{24a}$$

$$X(t) = [(A_n - A_{n-1})/2][\sqrt{C} \sin(\pi R_s t)]\,[u(t-nT_s) - u(t-(n+1)T_s)] \tag{24b}$$

where $u(t-nT_s)$ and $u(t-(n+1)T_s)$ are unit step functions.

When a transition occurs for the nth symbol for the odd numbered samples on the X and Y channels which are displaced by an error ε, the resulting relationships for the odd numbered samples are:

$$Y_{2n-1} = [(B_n - B_{n-1})/2]\,\sqrt{C} \sin \varepsilon/2 \tag{25a}$$

$$X_{2n-1} = [(A_n - A_{n-1})/2]\,\sqrt{C} \sin \varepsilon/2 \tag{25b}$$

Modulation transitions are identified by the conditions $A_n = -A_{n-1}$ and/or $B_n = -B_{n-1}$ at the A and B decision outputs of the demodulator.

If a transition is detected, the output at the odd numbered sampling instant corresponding to the symbol responsible for the transition is approximated by the above relationships. Thus, decisions on A and B can be used to convert the odd numbered samples of X and Y to estimates of the symbol timing error that can be used for symbol synchronization. These principles are used below in association with a first order phase lock loop to track symbol timing during the traffic portion of a TDMA burst. The same principles can be used to acquire phase but with less rapidity than the acquisition method previously described.

SYMBOL TIMING ERROR PROCESSING

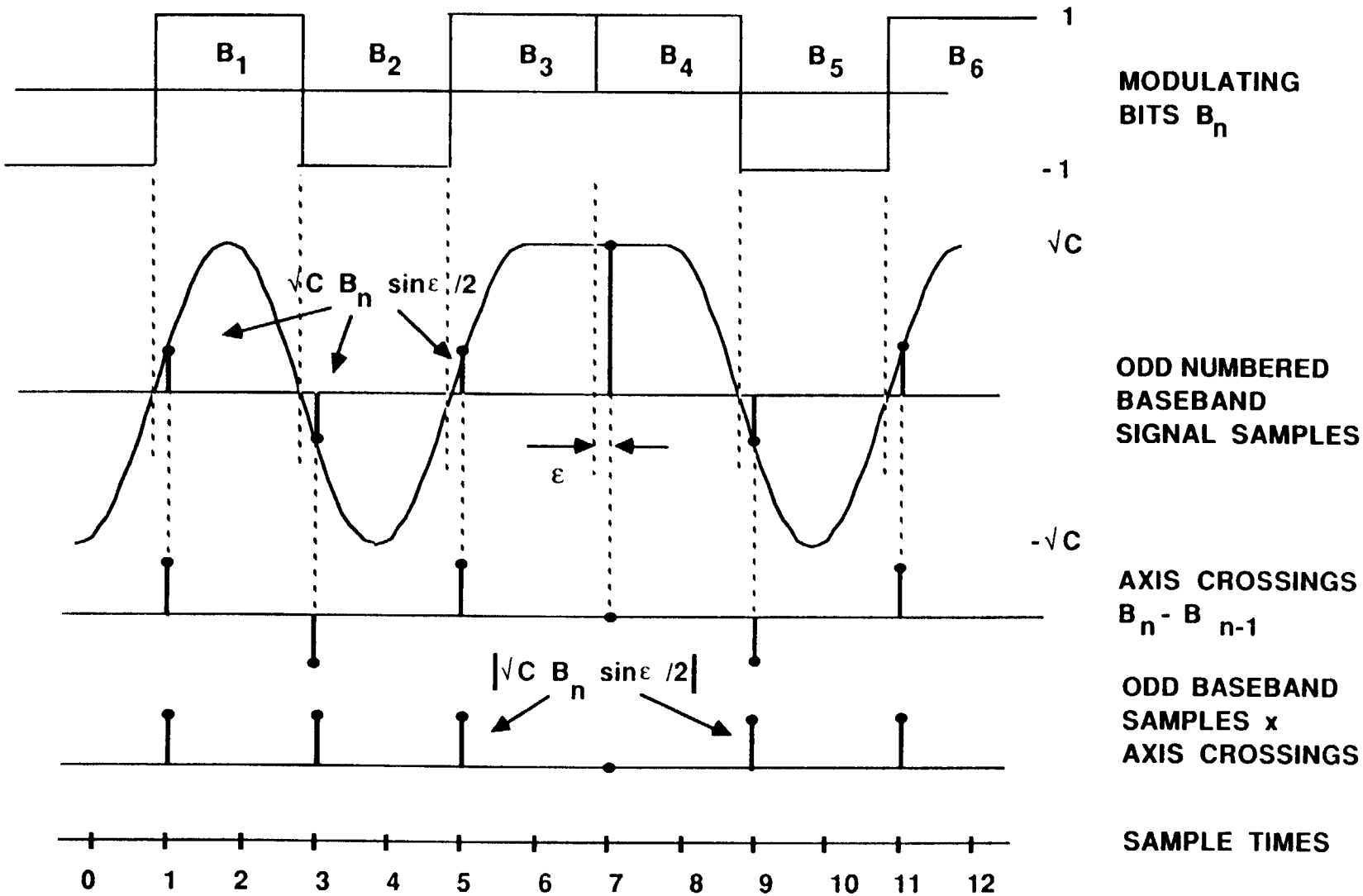


FIGURE 4.7. SYMBOL TIMING ERROR PROCESSING

As illustrated by equations (25 a&b), the samples on both quadrature components taken at odd numbered sample times have a magnitude that is proportional to $\varepsilon/2$ for small timing phase errors when transitions take place. When no transition occurs the odd sample times have large values given approximately by

$$Y_{2n-1} = B_n \text{ ,if } B_n = B_{n-1} \tag{26a}$$

$$X_{2n-1} = A_n \text{ ,if } A_n = A_{n-1} \tag{26b}$$

When the odd sample time values are multiplied by the polarity of the first difference of the detected bit decisions, the resulting values always have the same sign and the sign reverses between lagging and leading conditions. Furthermore, since transitions occur only when there is a bit reversal the method eliminates the contributions due to large sample values which would otherwise destroy the desired property. This is illustrated in Figure 4.7.

The transition detector is implemented by determining the first differences of the decisions made at the output of the decision detector. This decision process is represented by the following expressions:

$$Q\hat{}_{2n-1} = (B\hat{}_n - B\hat{}_{n-1})/2 \tag{27a}$$

$$P\hat{}_{2n-1} = (A\hat{}_n - A\hat{}_{n-1})/2 \tag{27b}$$

The values of these expressions are given by the following logic table for $Q\hat{}_{2n-1}$ as a function of $B\hat{}_n$ and $B\hat{}_{n-1}$:

		$B\hat{}_{n-1}$	
		1	-1
$B\hat{}_n$	1	0	1
	-1	-1	0

A similar table exists for $P^\wedge_{2n-1}$ as a function of $A^\wedge_n$ and $A^\wedge_{n-1}$.
Once each symbol the product of transition detector output and the odd numbered samples yields the following value for the error estimate:

$$\varepsilon^\wedge_n = [Q^\wedge_{2n-1} Y_{2n-1} + P^\wedge_{2n-1} X_{2n-1}]/\sqrt{C} \tag{28}$$

which can be expected to have a value

$$\varepsilon^\wedge_n = [Q^\wedge_{2n-1} B_n + P^\wedge_{2n-1} A_n] \varepsilon/2 \tag{29}$$

When no transition occurs on either channel the error value is zero and consequently produces no contribution to the correction process.

4.3.2.2 Symbol Synchronizer Operation

The symbol synchronizer is shown in Figure 4.8. It consists of a phase detector which obtains estimates of the phase error $\varepsilon^\wedge_n$ every symbol period followed by an amplifier of gain GT_s which in turn is followed by an accumulator which sums the amplified phase error estimates. The output
of the phase detector for the nth symbol is

$$\varepsilon^\wedge_n = (s_n - s^\wedge_n)/T_s \tag{30}$$

where s_n is the received symbol phase and $s^\wedge_n$ the currently estimated symbol phase. The output timing phase is updated by the accumulator to yield:

$$s^\wedge_n = s^\wedge_{n-1} + GT_s \varepsilon^\wedge_n \tag{31}$$

The corrected sampling signal phase estimates, $s^\wedge_n$, are supplied to the interpolator stage of the IFFT where they are used to adjust the phase of the sampling clock, hence sampling times, so that the value of $\varepsilon^\wedge_n$ is driven to values that meander about zero with small magnitude.

SYNCHRONIZER FOR SYMBOL TIMING TRACKING

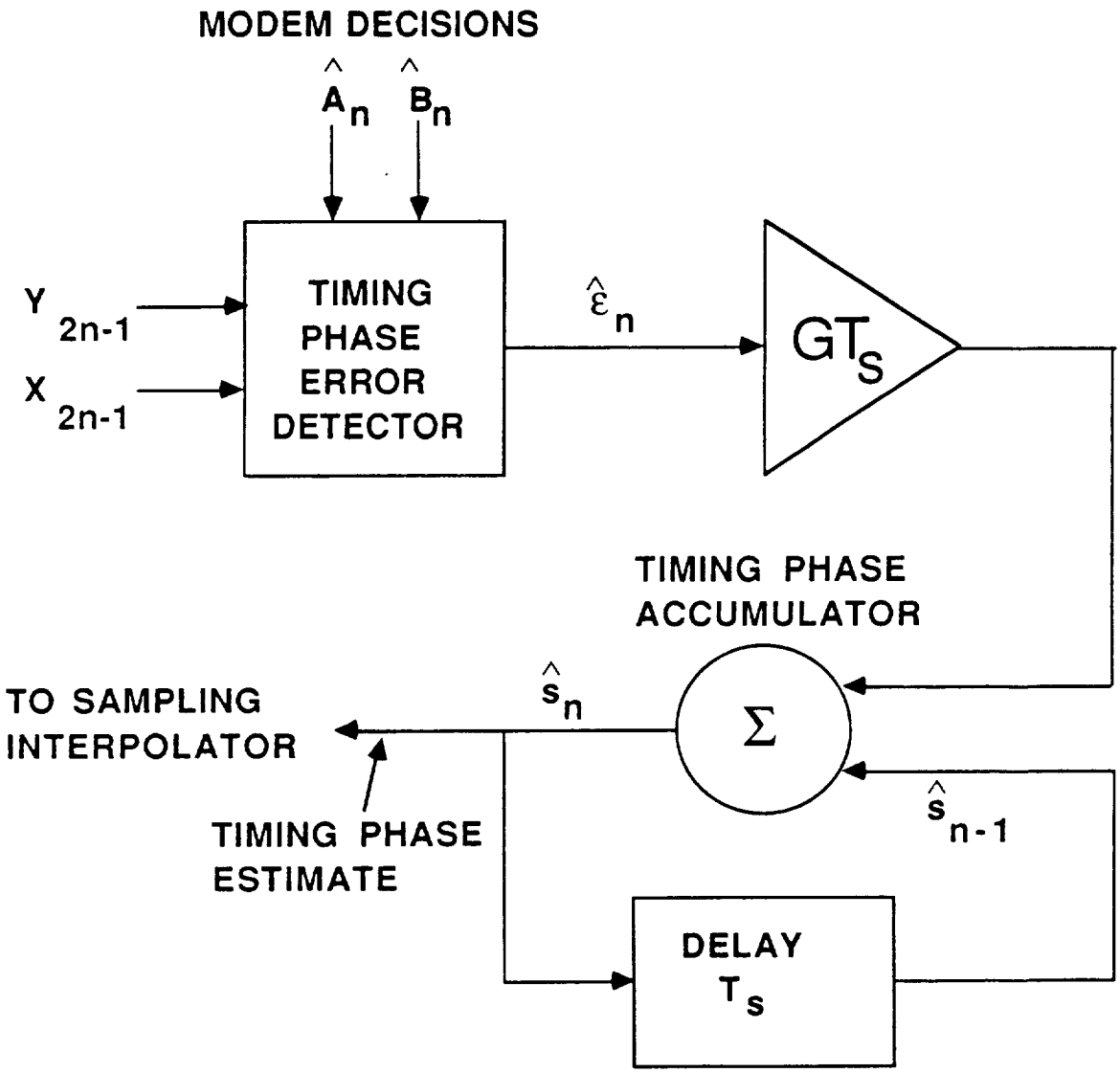


FIGURE 4.8. SYNCHRONIZER FOR SYMBOL TRACKING

The noise bandwidth B_N and loop bandwidth B_L of the discrete PLL are related to the gain G by

$$B_N = 2B_L = G/2 \tag{32}$$

Consequently, the averaging time is approximately

$$t_e = 2/G \tag{33}$$

which can be expressed in terms of symbols as

$$n_e = t_e/T_s = 2/(G\,T_s) \tag{34}$$

The variance in the estimate of ε obtained during each symbol interval averages

$$\sigma_1^2 = 2/(E_s/N_o) \tag{35}$$

Consequently, the variance over the smoothing interval t_e (with n_e symbols) averages

$$\sigma_\varepsilon^2 = \sigma_1^2/\,n_e = 2/\,[n_e\,(E_s/N_o)] \tag{36}$$

As a typical application, assume that GT_s is set to yield a value of n_e that results in a tracking error of $T_s/100$ when $E_s/N_o = 4$ corresponding to 6 dB. Then since $\varepsilon = 2\pi\,\Delta t/T_s$

$$\sigma_\varepsilon^2 = (2\pi)^2(\,0.01)^2 = 1/253 \tag{37}$$

With $E_s/N_o = 4$, the averaging time in terms of symbols is

$$n_e = 253/2 = 127 \tag{38}$$

and consequently the gain of the discrete PLL is

$$GT_s = 2/n_e = 0.0157 \tag{39}$$

4.3.2.3 Carrier Phase Tracking

During the traffic data portion of a TDMA burst, a sampled data second order phase lock loop shown in Figure 4.9 is used to maintain carrier synchronization. A second order loop contains two accumulators, one calculating estimated carrier phase $\theta^\wedge$ and the other estimated carrier phase rate $(d\theta/dt)^\wedge$, which of course is frequency. These accumulators are initialized by the estimates of phase and phase rate obtained from the preamble acquisition processing.

Because the synchronization has been acquired, the even numbered samples are located very near the mid symbol position. Under these circumstances, the actual carrier phase is θ and the estimated carrier phase is $\theta^\wedge$ and there is a small phase difference $\phi = \theta - \theta^\wedge$ between them. Under these conditions the quadrature modulation components for the nth symbol can be expressed in terms of ϕ by

$$Y_{2n} = \sqrt{C}\,(B_n \cos\phi + A_n \sin\phi) \tag{41a}$$

$$X_{2n} = \sqrt{C}\,(A_n \cos\phi - B_n \sin\phi) \tag{41b}$$

When $\phi = 0$ the cross coupling between the channels becomes zero. Consequently, the binary decisions on the samples Y_{2n} and X_{2n} should be very reliable estimates of the modulation variables A_n and B_n. Hence,

$$Y^\wedge_{2n} = B^\wedge_n \tag{42a}$$

$$X^\wedge_{2n} = A^\wedge_n \tag{42b}$$

Substituting the above relations into the following decision feedback cross product relation

$$F(n) = (X^\wedge_{2n} Y_{2n} - Y^\wedge_{2n} X_{2n})/2\sqrt{C} \tag{43}$$

yields the result:

CARRIER PHASE TRACKING LOOP

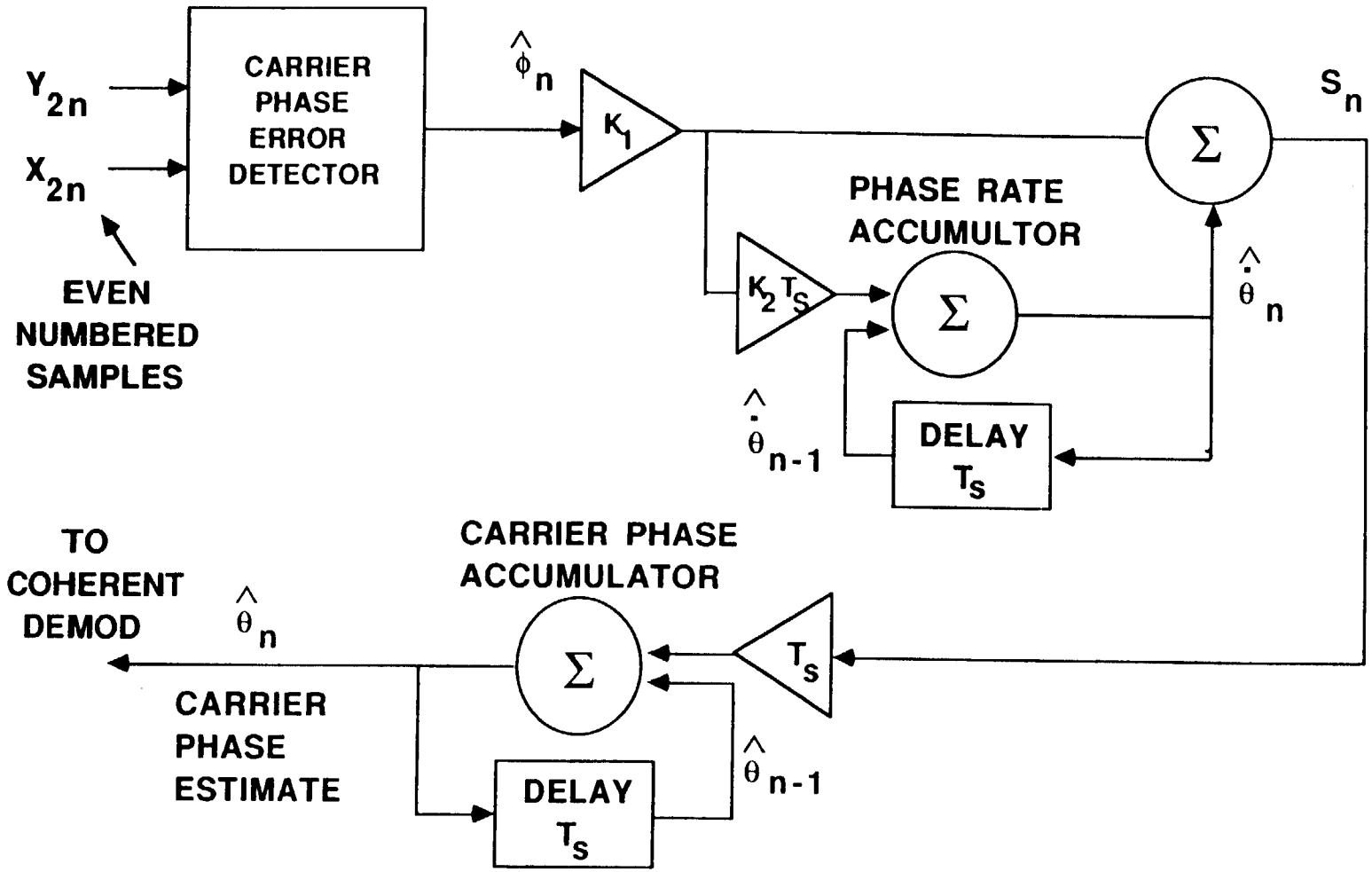


FIGURE 4.9. CARRIER PHASE TRACKING LOOP

$$F(n) = (A_n A^\wedge_n + B_n B^\wedge_n)(\sin\phi)/2 + (A^\wedge_n B_n - B^\wedge_n A_n)(\cos\phi)/2 \tag{44}$$

Consider the average of the above expression over a relatively large number of symbols. Cross product terms $A^\wedge_n B_n$ and $B^\wedge_n A_n$ average to zero since the bits comprising the information on the quadrature are randomly related. Co product terms $A_n A^\wedge_n$ and $B_n B^\wedge_n$ each average to 1 over the same averaging interval. There will be residual variance in the cross and co product terms which depends on the length of the averaging interval and contributes to error in the estimate. Hence, provided the averaging interval is sufficiently long,

$$\overline{F(n)} = \sin \phi^\wedge_n \approx \phi^\wedge_n \tag{45}$$

4.3.2.4 Carrier Synchronizer Operation.

The estimated value of the phase error, $\phi^\wedge_n$, determined by the phase error detection method described above is used to generate a new estimate, $\theta^\wedge_{n+1}$, of the carrier phase by means of the 2nd order discrete phase lock loop shown in Figure 4.9. For each new value of the phase error $\phi^\wedge_n$ the first summation loop generates an output S_n which is given by the expression

$$S_n = \theta^{\bullet\wedge}_n + K_1 \phi^\wedge_n \tag{46}$$

the differential term $\theta^{\bullet\wedge}_n = (\Delta\theta/\Delta t)^\wedge_n$ is the current estimate of the phase rate which is the frequency offset between the actual carrier and the recovered carrier. The phase rate accumulator inside the first summation loop also computes a new phase rate estimate

$$\theta^{\bullet\wedge}_{n+1} = \theta^{\bullet\wedge}_n + K_1 (K_2 T_s) \phi^\wedge_n \tag{47}$$

S_n is passed on to the second accumulator where it is summed with the old value of the phase to generate a new value according to the relation

$$\theta^\wedge_n = \theta^\wedge_{n-1} + S_n T_s \tag{48}$$

When the phase lock loop is ideally locked, the phase rate estimate $\dot{\theta}^{\wedge}_n$ is equal to the frequency difference between the actual and recovered carrier causing $\theta^{\wedge}_n$ to advance by an amount $T_s\,\dot{\theta}^{\wedge}_n$ for each symbol which is the precise amount needed to maintain the error estimate $\phi^{\wedge}_n$ equal to zero.

The value of $\theta^{\wedge}_n$ thus determined is supplied to the carrier phase corrector which is implemented as shown in Figure 4.10. This processor rotates the phase of the recovered carrier by $\theta^{\wedge}_n$ keeping it aligned with the phase of the signal carrier.

Performance of the phase lock loop can be expressed in terms of two parameters, the damping coefficient ζ and the natural undamped frequency ω_v. These are related to the loop gain parameters K_1 and K_2 by the expressions:

$$\zeta = [\sqrt{(K_1/K_2)}]/2 \tag{49}$$

$$\omega_v = \sqrt{(K_1 K_2)} \tag{50}$$

The effective noise bandwidth B_N of the phase lock loop, which is twice the low pass bandwidth B_L, is given by the expression

$$B_N = 2B_L = (\omega_v/2)(2\zeta + 1/2\zeta) \tag{51}$$

If a value $\zeta = 1/2$ is selected the following relationships result:

$$K_1 = K_2 = \omega_v = B_N \tag{52}$$

The loop carrier to noise ratio which determines the standard deviation in the recovered phase estimate is

$$C/N|_L = (E_s/N_o)(R_s/B_N) \tag{53}$$

where E_s/N_o is the symbol energy to noise spectral density ratio and R_s is

CARRIER PHASE CORRECTOR

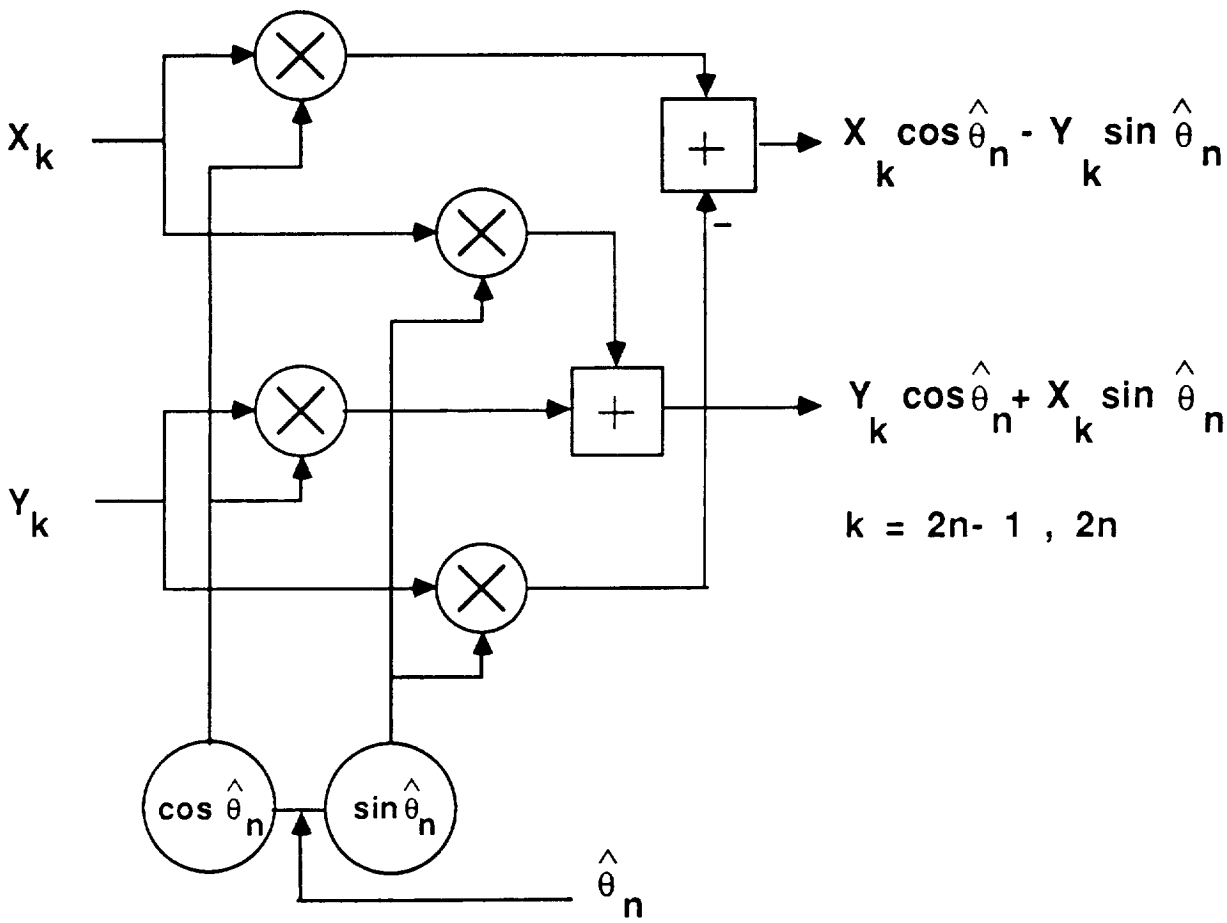


FIGURE 4.10 CARRIER PHASE CORRECTOR

the symbol rate. A convenient relation that results from the above expression is

$$B_N T_s = (E_s/N_o)/\ C/N|_L \tag{54}$$

or

$$B_N = K_1 = K_2 = \omega_v = R_s\ (E_s/N_o)/\ C/N|_L \tag{55}$$

Typically, to obtain a standard deviation of 3.2° in phase $C/N|_L$ must be 160. Furthermore, assuming $R_s = 10^6$ sym/s and $E_s/N_o = 6$ dB,

$$B_N = K_1 = K_2 = \omega_v = (4/160)\ 10^6 = 25600 \tag{56}$$

This result is for a damping coefficient $\zeta = 1/2$. Other values will result for other values of the damping coefficient.

4.4 COMPUTATIONAL REQUIREMENTS.

4.4.1 SYMBOL TIMING AND CARRIER ACQUISITION.

For acquisition of symbol timing and carrier phase and frequency, the preamble is divided into halves each containing $N_s/2$ symbols. For each half the following number of multiplications must be performed:

1) $N_s/2$ additions for each Σ_{Xo}, Σ_{Xe}, Σ_{Yo}, Σ_{Ye}, totaling $2N_s$.
2) 1 multiplication for each X_o^2, X_e^2, Y_o^2, Y_e^2, X_eY_e, X_oY_o, totaling 6.
3) 1 addition for each $X_o^2+X_e^2$, $Y_o^2+Y_e^2$, $X_o^2+Y_o^2$, $X_e^2+Y_e^2$, $X_eY_e+X_oY_o$, $X_oY_e+X_eY_o$, totaling 6.
4) 2 inverse tan operations implemented using PROMs.

Thus, the total requirement for the entire preamble of N_s symbols for each TDMA burst is $4N_s + 12$ additions, 12 multiplications and 4 inverse tan operations.

4.4.2 SYMBOL AND CARRIER TRACKING.

a) Symbol Tracking.

Symbol tracking, also referred as clock synchronization, requires the following:

1) 2 additions for every odd numbered sample to compute

$$Q\hat{}_{2n-1} = (B\hat{}_n - B\hat{}_{n-1})/2$$

$$P\hat{}_{2n-1} = (A\hat{}_n - A\hat{}_{n-1})/2$$

Since these involve values of only ±1 they can be performed by logic and don't count.

2) 2 multiplications and 1 addition every odd numbered sample to compute

$$\varepsilon\hat{}_n = [Q\hat{}_{2n-1}\, Y_{2n-1} + P\hat{}_{2n-1}\, X_{2n-1}]/\sqrt{C}$$

The multiplications involve values of ±1 and don't count.

3) 1 multiplication and 1 addition every odd numbered sample to compute

$$s\hat{}_n = s\hat{}_{n-1} + GT_s\, \varepsilon\hat{}_n$$

Thus a total of 2 additions and 1 multiplication are needed for each odd numbered sample to track the symbol timing. For a burst containing M traffic segment symbols there are M odd samples yielding a total requirement of 2M additions and M multiplications for each burst.

b) Carrier Tracking

Carrier tracking, also referred as carrier synchronization, requires the following:

1) 2 multiplications and 1 addition for each even numbered sample (hence for each symbol) to compute

$$F(n) = (X\hat{}_{2n}\, Y_{2n} - Y\hat{}_{2n}\, X_{2n})/2\sqrt{C}$$

2) 2 multiplications and 3 additions per symbol to update the carrier phase and frequency estimates as follows:

$$S_n = \theta^{\bullet}\hat{}_n + K_1\, \phi\hat{}_n \quad \Rightarrow \quad S_n T_s = \theta^{\bullet}\hat{}_n T_s + K_1 T_s\, \phi\hat{}_n$$

$$\theta^{\bullet}\hat{}_{n+1} = \theta^{\bullet}\hat{}_n + (K_1 K_2 T_s)\, \phi\hat{}_n \quad \Rightarrow \quad \theta^{\bullet}\hat{}_{n+1} T_s = \theta^{\bullet}\hat{}_n T_s + (K_1 K_2 T_s^2)\, \phi\hat{}_n$$

$$\hat{\theta}_n = \hat{\theta}_{n-1} + S_n T_s \quad \Rightarrow \quad \hat{\theta}_n = \hat{\theta}_{n-1} + S_n T_s$$

3) 4 multiplications and 2 additions per symbol for carrier phase rotation to compute

$$X_k \cos\hat{\theta}_n - Y_k \sin\hat{\theta}_n \quad \text{and} \quad Y_k \cos\hat{\theta}_n + X_k \sin\hat{\theta}_n, \quad k = 2n-1, 2n$$

Thus a total of 8 multiplications and 6 additions are needed for each symbol to perform the carrier tracking processing.

4.4.3 TOTAL DEMODULATOR REQUIREMENT.

The total requirement for processing the symbol timing and carrier acquisition and tracking is summarized in TABLE 4.1.

TABLE 4.1.
SYMBOL TIMING AND CARRIER ACQUISITION AND TRACKING COMPUTATIONS REQUIREMENTS PER SYMBOL

COMPUTATIONAL REQUIREMENT	MULTIPLIES/SEC	ADDITIONS/SEC
SYMBOL TIMING & CARRIER ACQ.	$12/N_s$	$4 + 12/N_s$
SYMBOL TIMING & CARRIER TRACK	9	8

From the above, the following relation can be derived for the computational requirement to process a shared TDMA carrier having a bit rate of R_b among a community of TDMA terminals:

$$\text{MULTIPLIES/SEC} = (R_b/2)(12 + 9M)/(M+N_s+G)$$

$$\text{ADDITIONS/SEC} = (R_b/2)(4N_s + 12 + 8M)/(M+N_s+G)$$

where G is the number of symbols allowed for guard time between bursts

and each terminal is assumed to transmit a burst having a preamble N_s symbols long and a traffic segment M symbols long.

For example consider a TDMA system having an average burst such that:

R_b = 120.832 Mbit/s
N_s = 128 symbols
M = 12288 symbols (24 64kb/s channels, 8ms frame)
G = 16 symbols

The resulting computational rates are:

MULTIPLIES/SEC = 538×10^6

ADDITIONS/SEC = 480×10^6

4.4.4 INTERPOLATION REQUIREMENT

Interpolation is performed on the output samples generated by the IFFT. It introduces the symbol timing correction $\varepsilon\hat{}_n$ and generates the samples X_k and Y_k that comprise the input to the demodulation process. The interpolation computational requirement is based on an interpolation filter with an impulse response that extends 4 symbols in each direction. This requires 16 multiplications for each sample on each quadrature channel yielding a total of 64 multiplications for each symbol. Thus the interpolation requirement is:

INTERPOLATION REQ. = $64 \times R_s$ multiplications/sec

For a composite rate of 120Mbit/s (R_s = 60Msamp/s) this is 3.866×10^9 mult/sec.

It is important to point out that the 120 Mbit/s TDMA example discussed above represents operation in a broadband channel of 80 MHz width and would not require either FFT/IFFT or interpolation processing if it were the only carrier to be processed. It is only when the signal to be processed is a composite of many different carriers that these latter processing elements are used.

4.5 DEMODULATION OF BPSK, 8-PSK AND OFFSET-QPSK

4.5.1 GENERAL

This section describes the operation of a completely digital demodulator for BPSK, 8-PSK and offset-QPSK. Because of the strong similarity with the QPSK demodulator, which was previously described in great detail, the description here is abridged. The presentation indicates the differences compared with the QPSK demodulator thereby avoiding duplicating a large body of identical material.

4.5.2 BPSK DEMODULATION

4.5.2.1 Acquisition Processing.

The acquisition process is identical to that used in QPSK both for the carrier and the clock.

4.5.2.2 Tracking Processing.

The tracking loops are identical to those used for QPSK but the estimates fed to these loops are slightly different. Referring to Equation 28, there is a similar equation here, except that $Q^\wedge = 0$ since only 1 bit is transmitted per symbol in BPSK. The error estimate at the input of the clock loop is therefore:

$$\varepsilon^\wedge_n = P^\wedge_{2n-1} X_{2n-1}/\sqrt{C} \qquad (57)$$

Similarly, in Equation 43 $Y^\wedge_{2n} = 0$ and the error estimate at the input of the carrier loop is:

$$F(n) = X^\wedge_{2n} Y_{2n} /2\sqrt{C} \qquad (58)$$

It is clear from the above that the differences between QPSK and BPSK demodulators are very minor and that a QPSK demodulator can be easily modified via microprocessor control to demodulate BPSK and vice-versa.

4.5.3 8-PSK DEMODULATION.

4.5.3.1 Acquisition Processing.

The acquisition process is identical to that used in QPSK both for the carrier and the clock.

4.5.3.2 Tracking Processing.

The tracking loops are also identical to the ones used for QPSK as is the error estimate supplied to the carrier loop. Only the clock error estimate is different because of the multiphase nature of the 8-PSK signals. This manifests itself when a transition occurs from one octal symbol to another. The transitions on the X and Y channels are no longer simple zero crossings but several transition levels are possible. Referring to Figure 4.11 and denoting the estimates for symbols n-1 and n, on the X channel as $A^\wedge_{n-1}$ and $A^\wedge_n$ respectively (and similarly $B^\wedge_{n-1}$ and $B^\wedge_n$ for the Y channel), yields

$$M^\wedge_n = (A^\wedge_n + A^\wedge_{n-1})/2 \tag{59}$$

$$N^\wedge_n = (B^\wedge_n + B^\wedge_{n-1})/2 \tag{60}$$

which are the estimated transition levels, and

$$P^\wedge_n = (A^\wedge_n - A^\wedge_{n-1})/2 \tag{61}$$

$$Q^\wedge_n = (B^\wedge_n - B^\wedge_{n-1})/2 \tag{62}$$

which are half the transition magnitudes.

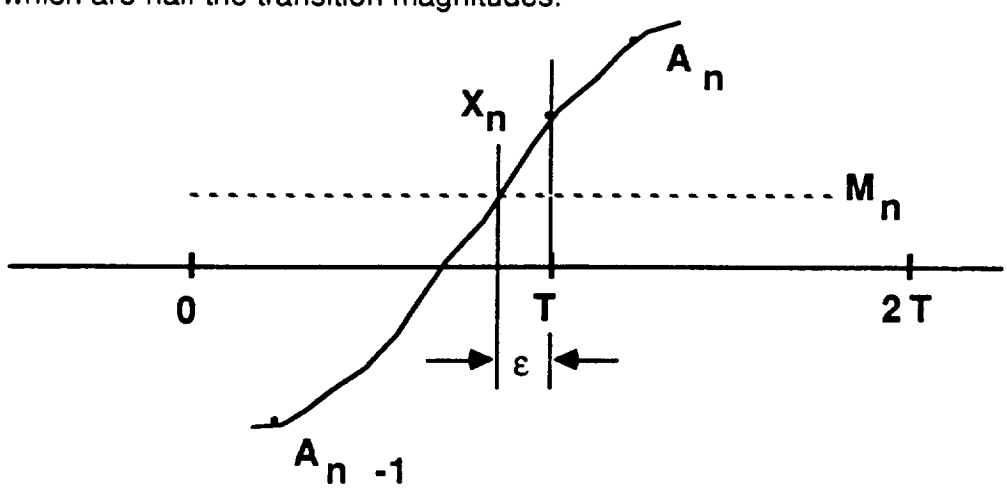


FIGURE 4.11 8-PSK TRANSITION ON THE X CHANNEL

Next, form an error estimate based on transition detections properly weighed to assign more weight to large transitions as follows:

$$\varepsilon\hat{} = (X_n - M\hat{}_n)\, P\hat{}_n + (Y_n - N\hat{}_n)\, Q\hat{}_n \tag{63}$$

This error estimate is then fed to a first order digital loop identical to that used for QPSK. Finally the decision rule is different from QPSK and is easily implemented. From the above discussion on 8-PSK it is concluded that with little effort it is possible to modify a QPSK demod via microprocessor control to demodulate 8-PSK signals.

4.5.4 OFFSET-QPSK DEMODULATION

4.5.4.1 Acquisition Processing

The preamble for offset-QPSK must be different than the QPSK alternating preamble, otherwise acquisition fails. This can be demonstrated as follows:

For the alternating preamble and due to the half symbol offset on the Y channel, the transmitted signal during the preamble has the form:

$$X(t) = \sin \pi R_s t \tag{64}$$

$$Y(t) = \cos \pi R_s t \tag{65}$$

After mixing at the receiver's oscillator, which has a phase offset θ and with a clock misalignment $\varepsilon/2$, the following results:

$$X = \sqrt{(2C)}\, [\, \sin(\pi R_s t + \varepsilon/2) \cos\theta - \cos(\pi R_s t + \varepsilon/2) \sin\theta] \tag{66}$$

$$Y = \sqrt{(2C)}\, [\, \sin(\pi R_s t + \varepsilon/2) \sin\theta + \cos(\pi R_s t + \varepsilon/2) \cos\theta\,] \tag{67}$$

Using well known trigonometric identities, the above equations may be rewritten as:

$$X = \sqrt{(2C)} \sin(\pi R_s t + \varepsilon/2 - \theta) \tag{68}$$

$$Y = \sqrt{(2C)} \cos(\pi R_s t + \varepsilon/2 - \theta) \tag{69}$$

Thus ($\varepsilon/2 - \theta$) can be determined but $\varepsilon/2$ and θ cannot be determined separately and thus the acquisition process fails. Therefore a different preamble must be used.

A suitable alternative is the alternating 45°, -45° sequence provided by:

$$A_n = 1 \quad \text{(constant)} \tag{70}$$

$$B_n = (-1)^n \quad \text{(alternating)} \tag{71}$$

For the even samples this yields:

$$X_{2n} = \sqrt{(2C)}\,[\cos\theta - (-1)^n \cos\varepsilon/2 \sin\theta] \tag{72}$$

$$Y_{2n} = \sqrt{(2C)}\,[\sin\theta + (-1)^n \cos\varepsilon/2 \cos\theta] \tag{73}$$

and for the odd samples we get

$$X_{2n-1} = \sqrt{(2C)}\,[\cos\theta - (-1)^n \sin\varepsilon/2 \sin\theta] \tag{74}$$

$$Y_{2n-1} = \sqrt{(2C)}\,[\sin\theta + (-1)^n \sin\varepsilon/2 \cos\theta] \tag{75}$$

For carrier acquisition simply add all the Y samples over the first half of the preamble and similarly for the X samples and obtain the arctan of the ratio of the Y sum over the X sum. Do the same for the second half of the preamble and then proceed as for the QPSK case.

For clock acquisition, begin by determining the mean signal value at the nth symbol interval from the 4 complex samples over this and the preceding symbol as follows:

$$m_{X,n} = (1/4)(X_{2n} + X_{2n-1} + X_{2n-2} + X_{2n-3}) = \sqrt{(2C)}\cos\theta_n \tag{76}$$

$$m_{Y,n} = (1/4)(Y_{2n} + Y_{2n-1} + Y_{2n-2} + Y_{2n-3}) = \sqrt{(2C)}\sin\theta_n \tag{77}$$

Note that the means must be calculated for each value of n because in the presence of a frequency offset ($\theta^{\bullet} \neq 0$), θ would vary with n.

Next subtract the means from the original samples and obtain new quantities as follows:

$$p_{2n} = X_{2n} - m_{X,n} = \sqrt{(2C)}\,(-1)^n \cos \varepsilon/2 \sin\theta \tag{78}$$

$$q_{2n} = Y_{2n} - m_{Y,n} = \sqrt{(2C)}\,(-1)^n \cos \varepsilon/2 \cos\theta \tag{79}$$

$$p_{2n-1} = X_{2n-1} - m_{X,n} = \sqrt{(2C)}\,(-1)^n \sin \varepsilon/2 \sin\theta \tag{80}$$

$$q_{2n-1} = Y_{2n-1} - m_{Y,n} = \sqrt{(2C)}\,(-1)^n \sin \varepsilon/2 \cos\theta \tag{81}$$

Next, proceed with the above p and q samples in the same way as with the X and Y samples for QPSK.

The preprocessing given above is needed to remove the samples means for offset-QPSK compared to QPSK due to the different nature of the preamble. Once the sample means are removed the remainder of the processing parallels that of QPSK.

4.5.4.2 Tracking Processing

After acquisition has been achieved, tracking proceeds as for QPSK after the X samples are delayed by a sample to give coincident alignment with the Y samples.

From the above discussion the tracking processing for offset-QPSK is almost identical to that of QPSK. The acquisition processing on the other hand needs some preprocessing after which it proceeds in the same way as for QPSK.

4.5.5 SUMMARY

The overall conclusion drawn from examining the various demodulators for PSK signals is that the processing involves the same types of computations and it is very possible to build one generic digital demod that can be programmed off-line via microprocessor control to demodulate BPSK, QPSK, 8-PSK or offset-QPSK signals. Digital implementation of the demodulator for MSK and SMSK has not as yet been considered in detail; however, except for differences in the computational procedures their implementation can certainly be accomplished using the same approach already used for the methods presently solved.

5.0 TECHNOLOGY SURVEY

5.1 GENERAL

Because of the high speed requirements of the on-board processor and because power is at a premium onboard the satellite, the implementation technology used must provide high speed, low power consumption and a high level of integration. A survey of commercial static RAMs and multipliers was performed by COMSAT LAB engineers by contacting high speed digital device manufacturers. The results are summarized in Tables 1 to 4. This information has been helpful in arriving at estimates of the power requirements for the various parts of the on-board processor and in carrying out trade studies between power requirements and performance.

Of paramount importance however, is the use of radiation-hardened devices. Unshielded devices in space are exposed to several hundred krads per year (one rad corresponds to the absorption of 100 ergs per gram of material). Proper shielding is essential although high launch cost per pound discourages extensive shielding of electronic devices in satellites. The use of proper grounding and coupling techniques in the design of devices is very important to reduce effects of radiation. An example of this is the insertion of resistors in the feedback paths of cross-coupled bistable circuit elements to dissipate the energy imparted by high-energy particle radiation and prevent an undesired change of state. These proper shielding, grounding and coupling techniques go hand-in-hand with the use of radiation hardened devices.

Three modes of a failure can be attributed to radiation exposure. Functional failures, parametric failures and single-event upsets. Functional failure is the failure to operate properly. Parametric failures occur when a device no longer meets its data-sheet specifications, although it may continue to function properly. A single event upset occurs when a high-energy particle imparts sufficient energy to a bistable circuit to change its state. Clearly the concern here is with the total dose of radiation as well as the dose rate. Both Si based and GaAs based technologies are promising for application requiring high speed, low power consumption and radiation hardened devices.

5.2 SILICON TECHNOLOGY

First consider Si based technologies. Several IC manufacturers are involved in producing CMOS and CMOS/SOS radiation hardened devices.

Based on examination of the available manufacturer's information, the most pertinent data is presented in Table 5. This data indicates that the power requirements and speeds of radiation-hardened components are comparable to those of their nonradiation hardened counterparts. However, the level of integration of high speed radiation-hardened components is still low. High levels of integration have been achieved at somewhat lower speeds. One example is the 80C86RH chip from Harris which is a 16-bit CMOS microprocessor that provides a total dose hardness level as great as 1 Mrad, consumes only 0.05 w/MHz and operates at clock frequencies up to 5 MHz.

5.3 VHSIC TECHNOLOGY

For use by the military, the very high speed integrated circuit (VHSIC) phase I program, sponsored by the Office of the Secretary of Defense (OSD), addressed the objective of providing radiation hard, high speed, silicon 1.25 μm technology integrated circuits for application in military systems. The VHSIC Phase II program extends the requirement of radiation hardened electronics to the 0.5 μm design rule regime.

Under the VHSIC program, several contractors have been developing radiation hardened gate arrays, memories and special purpose chips operating at frequencies above 25 MHz with modest power consumption. CMOS technologies have been developed at Westinghouse, NMOS at IBM, CMOS/SOS at Hughes and 3D bipolar as well as CMOS at TRW.

Some highly integrated chips that are of great interest to this study came out of these efforts. Multiport memories and high speed programmable matrix switches are among such chips. 64K SRAMs with access times of 35 nsec, 8 K CMOS/SOS configurable gate arrays operating at speeds above 25 MHz with less than 1/2 watt power dissipation are also among the achievements of the VHSIC program which are relevant to digital on-board processing. With the pipeline FFT as the workhorse of the demux/demod architecture special attention was paid to the recent developments in high speed FFTs in the VHSIC (as well as non VHSIC) areas. IBM and TRW are among the leaders in this area.

IBM has produced a complex multiplier accumulator (CMAC) NMOS chip that operates at 25 MHz. This chip is used for the butterfly computations of a radix 4 FFT. However, instead of computing the individual butterflies as 4 point FFTs, it computes them as 4 point DFTs. This results in 16 rather than 3 complex multiplications per butterfly. This amounts to more

than 500 percent waste in power needed for the multiplications. The maximum throughput rate of the IBM CMAC FFT processor is only 6.25 Mhz complex. Thus it takes 164 μsec to compute a 1024 point transform. Therefore it is concluded that IBM's highly integrated CMAC chip was designed as a general purpose complex multiplier accumulator and was not tailored for FFT applications.

TRW on the other hand has produced 2 CMOS chips specifically designed for butterfly computations as part of the VHSIC program. The first called the FFT arithmetic unit (FFTAU) is about 1 x 1 inch, has 105 pins and consumes 0.9 watts of power. The other called the FFT control unit (FFTCU) is also about 1 x 1 inch, has 105 pins and consumes 0.4 watts of power. These 2 chips operate in conjunction with 4 port RAMs in a radix 2 decimation-in-time, in-place FFT architecture. Because pipelining is lacking in this architecture, the maximum clock frequency is only 16.7 MHz. Nonetheless this is substantially higher than the 6.25 MHz of the IBM CMAC FFT. Also, TRW chips are more radiation hardened because of TRW's greater emphasis on space applications.

A faster FFT architecture found in the technology survey was also from TRW but not as part of the VHSIC program. By using pipeline architectures like the ones outlined in the text, FFT throughput rates higher than 20 MHz were achieved. The power consumption for a 512 point 20 MHz complex CMOS FFT was about 100 watts. This figure is high for two reasons. The first is that it uses 32 bit floating point arithmetic. Floating point arithmetic is more power consuming than fixed point arithmetic and is only needed in certain applications (our demultiplexer is not one of them) requiring very large dynamic ranges. The second is the level of integration. Before the end of the decade, much higher levels of integration are expected and it will be possible to put a 1024 or more point FFT on a single wafer resulting in a drastic decrease in weight and power consumption. Today's pipeline FFTs (such as TRW's and IBMs) are power consuming because higher levels of integration are yet to be achieved.

Comsat Labs has begun implementing fixed point pipeline FFT processor with throughputs larger than 20 M complex samp/s and a great deal of experience has been accumulated in this area. This fixed point technology will be more power efficient than the floating point implementations and hence more suitable for on-board use. High level integration of this approach should be pursued to achieve further reduction in power and size.

5.4 GaAs TECHNOLOGY

Consider now GaAs based technologies. On the positive side, GaAs digital circuits are capable of very high speed operation at low powers and possess a high tolerance to radiation. On the negative side cost has become a critical issue as a result of low yields. Also, high levels of integration are yet to be achieved. Provided R&D continues, it is only a matter of time until yields improve and integration levels increase. Facilities to produce LSI GaAs digital devices are being established with DARPA funding at Rockwell, McDonnell Douglas and Honeywell. Rockwell has developed 4 Kbit SRAMs with access times of 5 nsecs, and Honeywell is projecting 4 K, 1 ns memory with a maximum power dissipation of 1w by the end of 1987. GaAs gate arrays operating at frequencies above 1 GHz are also being produced with power dissipation less than 200 μw/gate.

The application of GaAs FET (field effect transistors) technology in radiation environments is attractive because of the high tolerance of MESFET (metal semiconductor FET) devices to total ionizing dose (10^6 to 10^8 rads). There is little information available on single event upsets in GaAs ICs, but the reports published so far are very promising.

NASA has also entered the GaAs digital arena with a program for an adaptable, programmable processor targeted for high speed processing of on-board space sensor data.

The conclusion from our technology survey is that for the near future high speed, low power digital signal processing will be mainly based on Si technologies (CMOS, CMOS/SOS) with GaAs being used mostly for high speed memories and at the analog to digital interface. In the farther future, as a result of continuing R&D in GaAs, a new generation of high speed, digital signal processing devices with enhanced radiation resistance will emerge. This can easily happen by the 1995 to 2005 time frame in which an operational satellite incorporating flight worth hardware that uses the concepts put forth in this study is likely to appear. In the immediate future, proof-of-concept laboratory units can be constructed from existing commercially available Si components and experimental components being developed as a result of the VHSIC program.

Table 5.1 8 x 8 Multipliers

Technology Family	Manufacturer	Multiply Time (ns)	Power (MW)	Power For One Multiplication/ns (W)
CMOS	Analog Devices	85	75	64
CMOS	TRW	45	31	1.4
GaAs	Gigabit Logic	10	500 (50)	5 (0.5)
GaAs	Rockwell	5.25	2,200	12
GaAs	Toshiba	12	160	1.9

Table 5.2 16 x 16 Multipliers

Technology Family	Manufacturer	Multiply Time (ns)	Power (MW)
NMOS	Bell	20	1,000
CMOS	TRW	165	500
CMOS	Analog Devices	75	175
CMOS	NEC	45	100
CMOS/SOS	Toshiba	27	150
GaAs	Fujitsu	10.5	950

Table 5.3 1 kbit of RAM

Technology Family	Manufacturer	Access Time (ns)	Power (MW)
ECL	Fairchild	10	940
ECL	NTT	0.85	950
CMOS	Cypress	15	450
GaAs	Fujitsu	1.3	300
GaAs	NEC	6	38
GaAs	Gigabit Logic	2	1,500 (150)
HEMT	Fujitsu	3.4	290

Table 5.4 4 kbits of RAM

Technology Family	Manufacturer	Access Time (ns)	Power (MW) 1K
ECL	Fujitsu	3.2	750
ECL	NEC	2.3	400
ECL	NTT	1.1	980
ECL	Hitachi	2.5	250
NMOS	Bell	5.0	100
GaAs	NTT	2.8	300
GaAs	Fujitsu	3.0	175
HEMT	Fujitsu	4.4	215

Table 5.5 Radiation Hardened CMOS and CMOS/SOS

SRAMS

Size (kbits)	Technology	Manufacturer	Access Time (ns)	Power (MW)
16	CMOS	Honeywell	110	1,000
16	CMOS	Harris	100	800
64	CMOS	Harris	220	
4	CMOS/SOS	CTI	70	125
16	CMOS/SOS	CTI	100	400

GATE ARRAYS

Number	Technology	Manufacturer	Time Delay (ns)	Power (MW)
3,500	CMOS	Honeywell	2	500
4,000	CMOS	Harris	2	
3,000	CMOS/SOS	CTI	2	480

6.0 RECOMMENDATIONS

6.1 GENERAL

This report describes an architecture for a flexible, modular, digital demultiplexer/demodulator for space applications. The building blocks of the architecture are pipeline processors for forward and inverse FFTs, a digital adaptive interpolating filter and a generic digital demodulator that can be programmed via microprocessor control to demodulate carriers of different modulation types and bit rates. In order to make the transition from the concept presented in this report to a space qualified processor, development efforts will be needed in two main areas.

6.2 PROOF OF CONCEPT MODEL

The first area of development is to build a proof of concept model of a Flexible Demultiplexer/Demodulator for bulk demodulation of a wideband channel such as 40 MHz with current state of the art components. This will provide a valuable opportunity to work out complex structural details details and control of the Down Converter/Sampler Pipeline FFT, Carrier Channel Filter, Pipeline IFFT, Interpolator and Demodulator needed to bulk process multiple carriers of different bit rates. Lessons learned from such a model will reveal opportunities to improve the current architecture and significantly reduce the difficulties and uncertainties that can be encountered in the later evolution to a VLSI intensive implementation. Computer simulations to support development of such an exploratory hardware model are already underway at COMSAT LABs.

6.2.1 FLEXIBLE BULK DEMUX/DEMOD POC BREADBOARD

The FLEXIBLE BULK DEMUX/DEMOD POC BREADBOARD would consist of the cardinal functional components shown in Figure 6.1 which are described briefly below.

6.2.1.1 DOWN CONVERSION AND SAMPLING

The channel to be processed will be 40 MHz in bandwidth and centered at an onboard IF frequency of approximately 3 GHz. The wideband channel will be down converted such that its center is at zero Hz. Complex sampling which uses two 40 Msamp/s A/D converters operating synchronously but independently on each quadrature phase will be

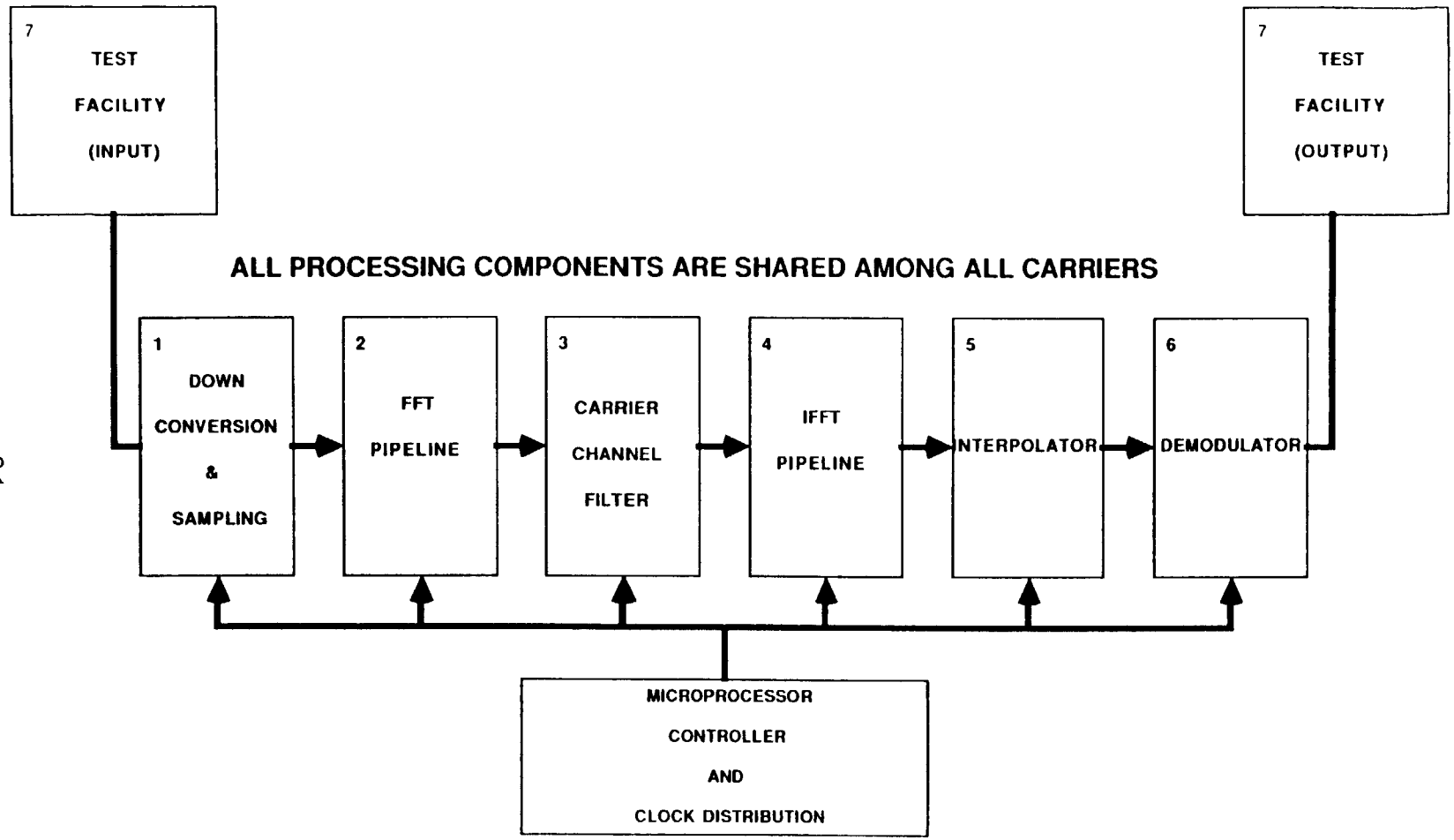


FIGURE 6.1 FLEXIBLE BULK DEMUX/DEMOD POC BREADBOARD /TEST FACILITY

incorporated. This is entirely possible in the current state of the art. To test the processor, an arrangement will be provided for representing multiple carriers ranging over carrier bit rates from 64 kbit/s to 6.144 or 6.3 Mbit/s using QPSK modulation. Both continuous duty FDMA and TDMA/FDMA carriers will be represented.

6.2.1.2 FFT PROCESSOR

To accommodate the lowest bit rate carrier, it is necessary that the spectrum be divided into frequency coefficients such that a minimum of 16 occur per carrier. To accomplish this, an FFT capable of resolving 16384 complex frequency coefficients over the 40 MHz wideband will be provided. The FFT processor will be based on a pipeline architecture using 25ns, 16x16 bit complex multipliers. Technology at this speed is currently emerging. The same FFT processor can accommodate any carrier bit rate up to a maximum of approximately 60 Mbit/s for QPSK modulation.

6.2.1.3 CARRIER CHANNEL FILTER

This filter processes sets of FFT frequency domain coefficients to select the desired channel using a matched filter approach. It can be programmed from the ground via the microprocessor controller and clock distribution unit to accommodate any arrangement of carrier frequencies and bit rates in the wideband channel. Its output is a set of filtered frequency domain FFT coefficients representing the information content of individual carriers.

6.2.1.4 INVERSE FFT (IFFT) PROCESSOR

The IFFT processor converts the sets of frequency domain coefficients for each carrier back to the time domain. Its implementation is such that a single pipeline FFT processor can be shared to perform the processing for all of the carriers. To do this, its internal operation and timing is properly controlled by the microprocessor controller and clock distribution unit according to the distribution of the carriers in the wideband spectrum. This can be adjusted to accommodate different arrangements of the carrier center frequencies and bit rates.

6.2.1.5 INTERPOLATING FILTER

The time domain samples delivered at the output of the IFFT processor are timed relative to the clock that controls the demultiplexer and this clock is established by the wideband signal sampler located at the input to the forward FFT. The time domain samples that are used in the demodulator are established by the need to sample the carrier signal appearing at the input to the demodulator at twice the symbol rate. Furthermore, the phase of the samples must be adjusted according to a phase control signal from the demodulator to align the samples at the proper positions in each symbol. To accomplish this, a sample interpolator will be provided between the IFFT output and the demodulator. The interpolation processor uses an impulse response that represents additional filtering of the channel and must be carefully chosen.

6.2.1.6 DEMODULATOR

A digitally implemented demodulator architecture for extracting the baseband digital information from the filtered carriers will be provided. For the POC unit, the demodulator will be implemented for QPSK since this is considered to be sufficient for demonstrating the important principles involved in the demultiplexing/demodulation processing. This requires processing to recover the carrier frequency and phase, the clock frequency and phase and the data. The signals are presented to the demodulator in the form of discrete time domain samples at a rate of two samples on each of two quadrature channels for each symbol interval. These samples are processed to recover the modulated data bits. To accomplish this, it is necessary to acquire and maintain both symbol timing and carrier frequency synchronization. The demodulation processor is shared to demodulate all of the carriers. It must be controlled by the microprocessor controller and clock distribution unit to accommodate the arrangement of carriers and bit rates assigned in the wideband channel.

6.2.1.7 MICROPROCESSOR AND CLOCK DISTRIBUTION UNIT

Operation of the flexible demultiplexer/ demodulator POC unit must be tightly synchronized to provide the timing discipline needed to control the flow of information within and between its constituent processing elements. This unit provides the clocks needed to accomplish this smooth

flow. It also provides program control of the clocks, relative timing of clocks and memory contents needed to adjust the system to accommodate different arrangements of carrier center frequencies and bit rates.

6.2.1.8 TEST FACILITY

The bulk Demux/Demod POC breadboard will include a Test Facility that appears at both its input and output. At the input it will provide a means for generating an environment of multiple carriers and bit rates in both FDMA and TDMA formats. It will include a source for generating a typical bit stream at the bit rates of interest (probably using a pseudo random bit stream generator) and a means for measuring the BER encountered when the stream is processed and appears at the output of the bulk demux/demod. Provision will also be made to introduce carrier frequency uncertainty typical of the satellite uplink FDMA and FDMA/TDMA transmissions. It will also contain a means for injecting thermal noise and cochannel and adjacent channel interference to allow for testing under practical application scenarios.

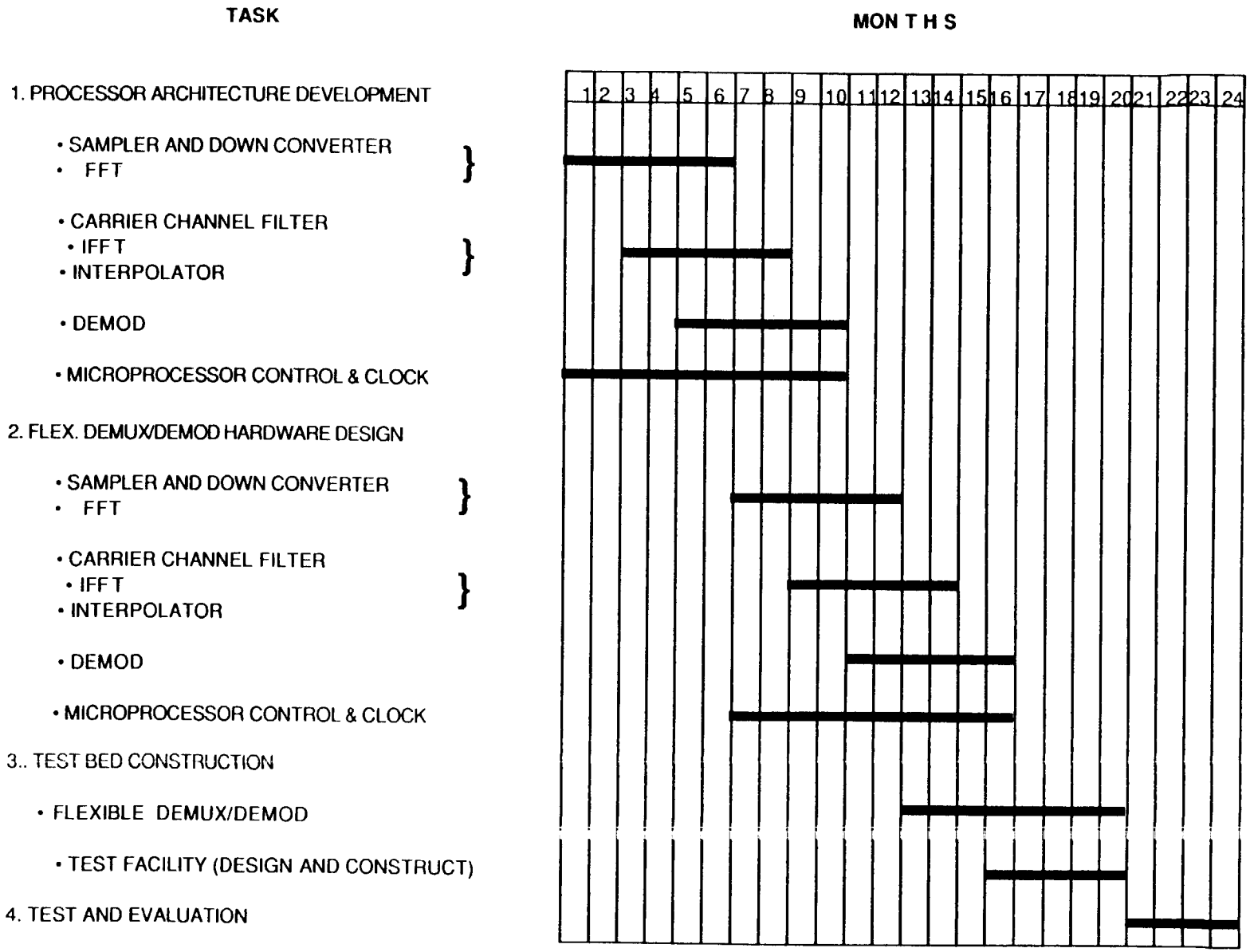
6.2.2 PROGRAM SCHEDULE

A schedule for performance of the Bulk Demux/Demod POC Breadboard development, spanning 24 months, is shown in Figure 6.2. The work program is divided into four principle elements:

1. PROCESSOR ARCHITECTURE DEVELOPMENT
2. FLEXIBLE BULK DEMUX/DEMOD HARDWARE DESIGN
3. TEST BED CONSTRUCTION AND
4. TEST AND EVALUATION

Under the Architecture Development element, processing details will be examined to arrive at structures of the seven major processor functions, illustrated in Figure 6.1 which when integrated will meet the needs for bulk demultiplexing/demodulation of multiple carriers with multiple bit rates in a 40 MHz wideband channel. Careful attention must be paid to minimizing the total computation load by using efficient procedures and algorithms and distributing the load across multiple calculating elements so as to arrive at a practical implementation. Computer modeling of key processor functions will be used to study

FIGURE 6.2 FLEXIBLE BULK DEMULTIPLEXER/ DEMODULATOR PROCESSOR DEVELOPMENT PROGRAM



implementations and to assure the proper working and interworking of the components. The tasks under this element are divided into five groupings comprising combinations of the processors illustrated in Figure 6.1 that are logically associated. The work is time phased to allow for distribution of the talent of the experts needed to do the job over all of the processing components. The architecture development is fully completed by the 10th month.

Under the Hardware Design element, the resulting architectures are to be committed to a hardware design. Because of the high speed of the processor components, care must be exerted to maintain a chip layout that minimizes transport delays and fully considers race conditions. It is expected that compact multilayer printed circuits designed to minimize reflections will be extensively incorporated. Both VHSIC and other commercially available VLSI components will be used extensively to achieve hardware that exhibits a practical balance between speed, compactness and power. Use of high speed gate array chips will be considered where costs permit. The effort on the processors will be grouped and time staggered in the same manner as done for the architectual development to distribute the work load. Hardware design begins in the 7th month and is completed by the 17th month.

Flexible Bulk Demux/Demod POC Test Bed construction begins in the 13th month with release of the Down Converter and Sampler design. Construction will continue on all processing components as design releases occur and be completed by the 20th month. The Test Facility design and construction will be initiated in the 16th month and its completion will coincide with the completion of the Flexible Bulk Demux/Demod. The test bed will be of a quality of construction suitable for use as a laboratory test and evaluation tool.

Following completion of construction, a rigorous Test and Evaluation Program will be performed from the 20th to the 24th month. Tests will be designed to evaluate the performance of the Flexible Bulk Demux/Demod under the frequency uncertainty, interference, noise and signal fade environment characteristic of satellite onboard signal reception expected at Ka band.

A final report will be prepared and delivered 2 months following completion of the work. It will contain full documentation of the architecture, design and construction and the results of the test and evaluation. It will provide sufficient information to create a design plan

for a space flight model. It will also contain a technology update of the components becoming available that may promise further improvements in the design and its radiation robustness of the processor.

The POC model is intended to provide an opportunity to develop the details of the flexible demultiplexer/demodulator processor architecture using available components and to provide a vehicle to experience its operating principles. Hence the accent should be on precise inspection of the fine details of the processor and its control. Power and weight are also important considerations but in this effort they should be subservient to the need to define the most efficient processing system that can later be implemented with advanced VLSI components to minimize power and weight.

6.3 SEMICONDUCTOR TECHNOLOGY

The second area where NASA should direct its R&D resources is in advancing the state of the art of semiconductor technology as it relates to the on-board processor. Needless to say that advances in technology stimulated by such a program will have spillovers in both civilian and military areas as well, as has happened many times in the past in NASA sponsored programs. Specific recommendations or areas of technology where efforts need to be directed are given below.

6.3.1 HIGHLY INTEGRATED CHIPS

Development of chips with high levels of integration that are designed to perform specific tasks is essential. The butterfly operation of the FFT is a good example. Rather than using individual multiplier and adder chips combined with memory and control chips as the building blocks to construct a butterfly element, large savings in power, weight and size can be realized by a single butterfly chip that embodies all of these functions in a single processor. This level of integration is certainly within the realm of today's technology, an example being IBM's complex multiply accumulator chip CMAC. What is needed is a proven formula for the implementation such as that that can be realized by pursuing the directions outlined in this report to the next phase, namely the construction of a discrete component proof of concept model.

6.3.2 FFT ELEMENTS

The FFT plays such a fundamental role in digital signal processing

that the design and fabrication of a special purpose chip to perform its fundamental operation, i.e. the butterfly, is a very sensible objective. For high speed real-time applications, a pipeline FFT similar to the one discussed in this report is often needed. Such a pipeline processor requires in addition to the butterfly elements, a commutator element involving delays of various sizes as well as switches to control it. Again, a special purpose chip (as opposed to combining several chips with low levels of integration) to perform the commutator action would significantly reduce the power and weight of the processor. Indeed, provided that adequate funding is available, it should be possible to build an entire pipeline processor on a single wafer before the end of this decade.

6.3.3 MICROPROCESSORS

Another area where technological advances are needed is radiation hardened microprocessors operating at high speeds with low power consumption. Harris has produced a 32 bit radiation hardened microprocessor operating on a 5 Mhz clock with as little as 0.05 w/Mhz. A great variety of timing and control functions will need to be performed at great speeds on-board the satellite and microprocessors working on faster clocks will be needed.

6.3.4 GaAs TECHNOLOGY

An area that is showing great promise is GaAs technology with its high speed, low power and high radiation resistance. Efforts to increase the packing density of GaAs chips are needed before the benefits of this technology can be fully reaped. In the area of memory storage (particularly PROMs) the high immunity of GaAs to single event upsets (that could reverse a stored 1 into a 0 or vice versa) makes it particularly attractive. Work is needed to produce large GaAs memories on smaller chips.

6.3.5 A/D CONVERTERS

At the analog to digital interface, today's A/D converters operating at speeds above 50 Mhz can provide 8 bits of quantization with good linearity. In order to process wideband transponders of 80 Mhz or more and resolve them into hundreds of narrowband carriers, a large dynamic range calling for A/Ds of 10-12 bits will be needed. This is particularly true when operating at K_u band where severe fades occur. Work has been

going on in this area for many years using bipolar technology and more recently CMOS and GaAs and should continue.

6.4 SUMMARY

Some areas of development have been outlined above that are important to achieving technological advances to establish a position to build a space-qualified advanced digital processor. Clearly progress in these technological areas will have benefits reaching far beyond any one particular program. More project oriented efforts should focus on building an exploratory model of the digital processor as a stepping stone before embarking on a more elaborate and costly VLSI implementation. Such an effort should go hand in hand with efforts on the technology development side so that in a few years both areas will have matured enough to realize a very sophisticated on-board processor. Parallel development efforts aimed at improving the implementation algorithms at the same time as the technology needed to realize the implementation is advanced is the real secret to successful realization of advanced onboard processing machines of the future. It is important that these developments be pursued vigorously with the goal of a practical implementation by 1995 if the satellite communications industry is to make use of the technology in the next generation of commercial satellites.

7.0 REFERENCES

[1] A. Antoniov, Digital Filters: Analysis and Design, McGraw-Hill Book Company, 1979.

[2] M. A. Bellanger and J. L. Daquet, "TDM-FDM Transmultiplexer: Digital Polyphase and FFT," IEEE Transactions on Communications, Vol. COM-22, September 1974, pp. 1199-1205.

[3] R. E. Chochiere and L. R. Rabiner, Multirate Digital Signal Processing, Englewood Cliffs, New Jersey: Prentice-Hall, Inc. 1983.

[4] F. M. Gardner, Phaselock Techniques, John Wiley & Sons, Second Edition, 1979.

[5] S. Kato, T. Arita, and K. Morita, "Onboard Digital Signal Processing for Present and Future TDMA and SCPC Systems," IEEE Journal on Selected Areas in Communications, Vol. SAC-5, May 1987, pp. 685-700.

[6] N. J. Nussbaumer, Fast Fourier Transform and Convolution Algorithms, New York: Springer-Verlag, 1982.

[7] A. Oppenheim, Editor, Applications of Digital Signal Processing, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1978.

[8] A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1975.

[9] L. R. Rabiner and B. Gold, Theory and Applications of Digital Signal Processing, Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1975.

[10] J. J. Stiffler, Theory of Synchronous Communications, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1971.

[11] E. E. Swartzlander, Jr., VLSI Signal Processing Systems, Kluwer Academic Publishers, 1986.

[12] E. E. Swartzlander, Jr., TRW Systems Defense Group, private communication.

[13] A. J. Viterbi, Principles of Coherent Communication, McGraw-Hill Book Company, 1966.

[14] B. Widrov and S. D. Stearns, Adaptive Signal Processing, Englewood Cliffs, New Jersey: Prentice-Hall, Inc. 1983.

[15] E. Yam and M. Redman, "Development of a 60-Channel FDM-TDM Transmultiplexer," COMSAT Technical Review, Vol. 13, No. 1, Spring 1983.

[16] "Advanced On-Board Digital Processing: Study Phase," COMSAT Laboratories Final Report CTD-86/228, submitted to INTELSAT Satellite Services.