

# Chip Design for Monobit Receiver

David S. K. Pok, Chien-In Henry Chen, *Member, IEEE*, John J. Schamus, Christine T. Montgomery, and James B. Y. Tsui, *Fellow, IEEE*

**Abstract**—A design for the monobit-receiver application-specific integrated circuit (ASIC) will be described. The monobit receiver is a wide-band (1-GHz) digital receiver designed for electronic-warfare applications. The receiver can process two simultaneous signals and has the potential for fabrication on a single multichip module (MCM). The receiver consists of three major elements: a nonlinear RF front end, a signal sampler and formatting system (analog-to-digital converter (ADC) and demultiplexers), and a patented “monobit” algorithm implemented as an ASIC for signal detection and frequency measurement. The receiver’s front end, ADC, and algorithm experimental performance results were previously presented [1]. The receiver uses a 2-b ADC operating at 2.5 GHz whose outputs are collected and formatted by demultiplexers for presentation to the ASIC. The ASIC has two basic functions: to perform a fast Fourier transform (FFT) and to determine the number of signals and report their frequencies. The ASIC design contains five stages: the input, the FFT, the initial sort, the squaring and addition, and the final sort. The chip will process the ADC outputs in real time, reporting detected signal frequencies every 102.4 ns.

**Index Terms**—Analog-to-digital converter (ADC), fast Fourier transform (FFT), instantaneous frequency measurement (IFM) receivers.

## I. INTRODUCTION

THE characteristics of instantaneous frequency measurement (IFM) receivers make them suitable for electronic-warfare applications. An IFM receiver has a wide instantaneous RF bandwidth—sometimes as much as several octaves. The receiver can measure short pulses with high-frequency accuracy (i.e., 1-MHz resolution on 100-ns pulse). A conventional IFM receiver is limited to processing only one signal. If two signals arrive at the receiver simultaneously, the receiver may generate erroneous information without the operator knowing. Various techniques have been used to detect the existence of simultaneous signals or detect the existence of erroneous frequency, but only limited success has been achieved.

Manuscript received March 31, 1997; revised August 15, 1997. This work was supported in part by the U.S. Air Force Office of Scientific Research under Contract F49620-96-1-0423. The work of C.-I. H. Chen was supported by the U.S. Air Force, the U.S. Air Force Office of Scientific Research, the state of Ohio, LSI Logic Corporation, and Texas Instruments Incorporated.

D. S. K. Pok and C.-I. H. Chen are with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435 USA.

J. J. Schamus is with the System Research Laboratory, Wright-Patterson AFB, OH 45433 USA.

C. T. Montgomery and J. B. Y. Tsui are with Wright Laboratory, WL/AAMP, Wright-Patterson AFB, OH 45433 USA.

Publisher Item Identifier S 0018-9480(97)08354-3.

This paper presents a very simple digital-receiver design as an replacement for the analog IFM receivers. The digital design can cover 1-GHz bandwidth and can correctly process two simultaneous signals. This design uses a fast Fourier transform (FFT) to obtain frequencies on up to two simultaneous signals. It has a better sensitivity than IFM receivers because the FFT channelizes the input into narrower bandwidth. It has fine frequency resolution (able to separate two close frequencies) and good frequency accuracy. The single- and two-signal spur-free dynamic ranges are very high. The only deficiency in this design is that the instantaneous dynamic range (receiving a strong and a weak signal simultaneously) is low. This paper presents the: 1) technical approach to design the receiver monobit-receiver application-specific integrated circuit (ASIC); 2) experimental results; and 3) performance comparison with a typical IFM receiver.

## II. MONOBIT RECEIVER

The design of this receiver can be divided into three areas, as shown in Fig. 1. They are the RF front end, the analog-to-digital converter (ADC), and data-formatting circuitry, and an ASIC to perform the FFT operation and the frequency selection.

### A. RF Front End

The RF front end will be similar to a conventional IFM receiver. The input signal will pass a bandpass filter followed by a limiting amplifier with a 60-dB gain to amplify the input and limit the output at a constant level. After the limiting amplifier, another bandpass filter limits the out-of-band noise [1]. In an IFM receiver this second filter is not needed. In this design, the filters have a passband from 1.375 to 2.375 GHz. This design will provide high single-signal dynamic range. The two-tone spur-free dynamic range is also high because the receiver processes only two signals and the spurs will be neglected. The nonlinear characteristic of the limiting amplifier will cause capture effect, which limits the instantaneous dynamic range.

### B. Signal Sampler and Formatting System

Because the signal from the limiting amplifier has a constant amplitude, a 2-b ADC will be satisfactory. Experimental results showed that a 2-b ADC is better than 1 b, but three or more bits show very little improvement because of the limiting amplifier and the unique FFT design discussed in the following section. In order to cover 1-GHz bandwidth,

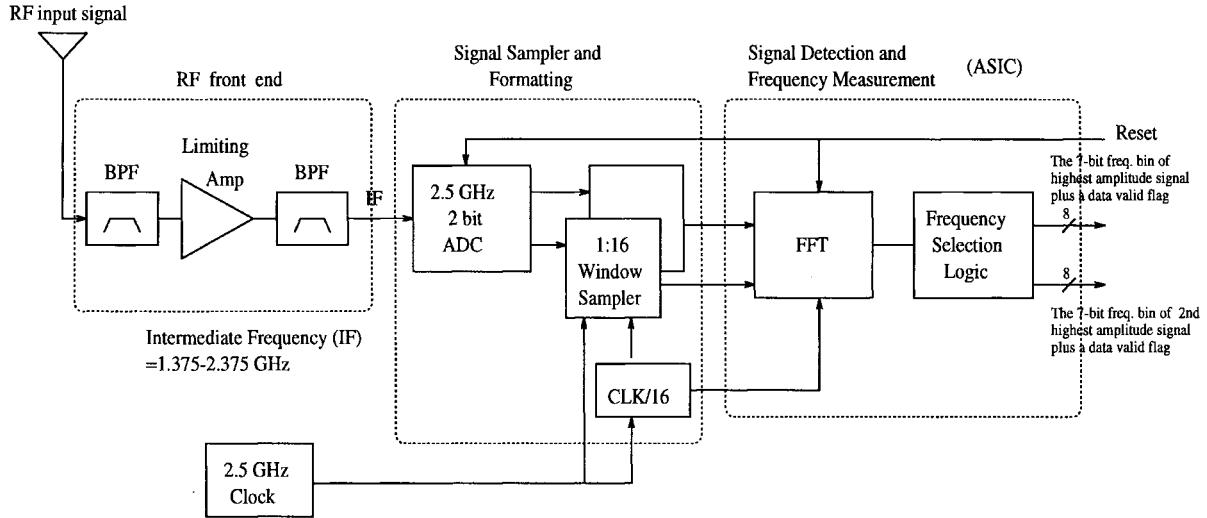


Fig. 1. Three areas of monobit receiver.

the ADC should operate at approximately 2.5 GHz. The two lowest unambiguous ranges are from 0 to 1.25 and 1.25 to 2.5 GHz. A 1-GHz portion (1.375–2.375 GHz) from the second unambiguous frequency range is selected as the input bandwidth. The second bandpass filter in the RF chain removes the noise in the 0–1.25-GHz range. The input frequency response of the ADC must be high enough to accommodate the input bandwidth of the receiver.

The input signal is first passed to the ADC, which samples the signal every 0.4 ns to produce 2-b amplitude measurements. Each bit is then passed to an associated windowing circuitry, which collects a 16 sample serial window of data and outputs the data in parallel to the detection algorithm chip. Thus, the windowing circuitry has two key functions: converts the serial data stream to parallel and slows down the data rate by a factor of 16, i.e.,  $(2.5\text{-GHz sampling rate})/(16\text{-sample window}) = 156.25\text{-MHz data rate}$ . The slowing of the data rate is necessary to accommodate the speed at which the detection chip can receive data. Note a reset flag between the ADC and ASIC coordinates the beginning of a data collection. The reset signal can be provided by test equipment. In system integration development, the reset signal would be provided by a post-processor who would also be collecting the outputs.

### C. Signal Detection and Frequency Measurement

1) *FFT Design:* This is the key component of the monobit design. The purpose is to eliminate multiplications and keep only adders in the discrete Fourier transform (DFT) chip design. The DFT can be written as [2], [3]

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad (1)$$

where  $j = \sqrt{-1}$  and  $N$  is the total number of sampled input points. In this equation, the result is obtained from the product of two functions: the input  $x(n)$  and the kernel function  $e^{-j2\pi kn/N}$ . If either one of these two functions is 1 b (monobit), i.e., + or -1, the operation requires only

additions. With limited investigation, it appears that it is easier to implement the monobit kernel function in hardware than the monobit input. The kernel function is rounded to + or -1 and + or - $j$  and this is mapped to a time-decimated, radix-2 FFT algorithm. The FFT contains 256 points. Sampling at 2.5 GHz the total time is approximately 100 ns, which can be considered as the minimum pulselength. There is no certainty that the signal will completely fill a 100-ns observation period. For this case, the signal will be detected in a 200-ns period. The frequency cell is  $1250/128 = 9.77$  MHz. The sensitivity of the monobit receiver is determined by this bandwidth, but the sensitivity of the IFM receiver is determined by 1250 MHz and the video bandwidth. In order to further simplify the design, the adders are limited to a maximum of 7 b (6-b amplitude and 1-b sign). If the outputs from the adders are beyond 7 b, they will be truncated to 7 b.

The FFT ASIC inputs are two 16-b data windows at a rate of 156.25 MHz. The input stage receives and stores each 16-b data window until 16 windows have been collected, i.e., total data is  $(16 \text{ windows}) \times (16 \text{ data samples/window}) = 256 \text{ data samples}$ . Thus, a complete data set is ready every  $(16 \text{ windows}) \times (6.4 \text{ ns per window}) = 102.4 \text{ ns}$  where the window sampling circuitry is feeding the FFT chip every  $(0.4\text{-ns ADC sampling rate}) \times (16 \text{ samples}) = 6.4 \text{ ns}$ . Therefore, each stage of the pipeline is being designed to a maximum of 102.4-ns worst-case processing.

2) *Frequency Selection Logic:* This is one of the most difficult designs in electronic-warfare receivers with multiple-signal capability. The goal is to select the correct input frequencies and avoid picking up spurious responses. Since the number of input signals is unknown, it is difficult to obtain the correct answer, especially if high instantaneous dynamic range is desired. In the monobit receiver design, the maximum number of signals to be processed is limited to two. Thus, the receiver is only required to determine between zero and two signals. In addition, the instantaneous dynamic range of this receiver is low, because of the RF front-end design and the 2-b ADC. These two requirements simplify the logic frequency

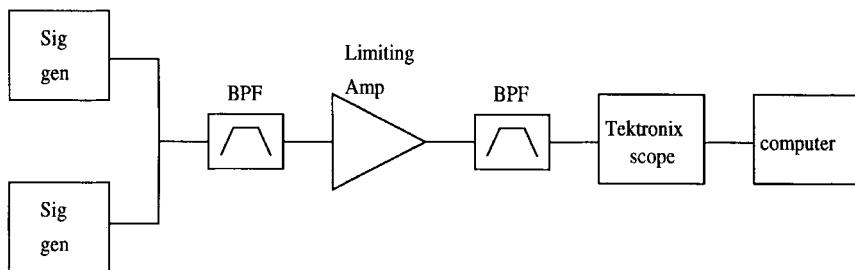


Fig. 2. Experimental setup.

TABLE I  
RESULTS FROM ONE SIGNAL

	Found Actual Signal (%)		Found False Signal (%)	
	FFT		FFT	
	7-bit	8-bit	7-bit	8-bit
Single Signal input	100.0	100.0	0.90	0.44

7-bit FFT: The adders in FFT design are limited to a maximum 7 bits.  
If the outputs are beyond 7 bits, they are truncated to 7 bits.

8-bit FFT: The adders in FFT design are limited to a maximum 8 bits.  
If the outputs are beyond 8 bits, they are truncated to 8 bits.

design significantly. One only needs to check the two highest peaks in the frequency domain to see whether they cross certain thresholds.

In the FFT chip, the frequency-selection logic mainly provides two outputs to a post-processor: the 7-b frequency bin of the highest amplitude signal plus a data valid flag, and the 7-b frequency bin of the second highest amplitude signal plus a data valid flag.

### III. EXPERIMENTAL RESULTS

Since the RF limiting amplifier and 2-b ADC are highly nonlinear, it is difficult to accurately simulate the results. An experimental setup was used to evaluate the performance of the receiver. The experimental setup is shown in Fig. 2, where the limiting amplifier has a gain of approximately 60 dB. The input bandwidth of this setup was 1 GHz (from 1.375 to 2.375 GHz).

A Tektronix TDS 684A oscilloscope was used as the ADC to collect the digitized data. The scope operated at 2.5 GHz and had 8-b output. The 8-b outputs were converted to 2-b through a software program. These 2-b data were processed through a 1-b kernel function simulated in a computer program. The maximum number of output bits of the adders was limited to seven to reduce hardware when it is fabricated on a chip. The highest two frequencies to cross certain thresholds will be declared as the desired signals. These threshold values are eight and four for 7-b FFT (18 and 10 for 8-b FFT). 8-b outputs were also used in the simulation to check the difference with 7 b.

First, no signal was applied to the input, and the program was run to detect false alarms. For 350 000 runs there is no false alarm, but this only represents 35 ms ( $350\ 000 \times 100 \times 10^{-9}$ ) in real testing time. Second, one signal with random frequency was applied to the input of the setup with

amplitude ranges from -70 to 10 dBm in 10-dB steps. At each power level, 100 runs were performed. If the output frequency is within 6 MHz of the input signal, it is considered as the correct answer. The results are shown in Table I. The frequency reading was always correct. However, some spurs were recorded as a second signal. Third, when the input signal amplitude was at -75 dBm, the receiver detected the input signal 88% of the time and generated one false alarm. Finally, two simultaneous signals were applied to the input. The two signals were random in frequency, but their amplitude must be very close, otherwise the receiver will miss the weaker signal. The minimum frequency separation was 20 MHz (slightly wider than two-channel width) and the maximum amplitude separation was set to 5 dB. If the two signals are separated by more than 5 dB, the receiver will read the strong signal only. One signal amplitude changed from 10 to -70 dBm in 10-dB steps. At each of these power levels the second signal changed from 0 to -5 dB with respect to the first signal in 1-dB steps. At each combination of power levels, 100 runs were taken. The results are shown in Table II. The receiver usually reads both frequencies correctly when the two signals are close in amplitude.

Sometimes the receiver misses both signals, because neither signal crosses the threshold. Sometimes, the receiver reads a spurious signal rather than the true signal. In this table, each value was obtained from 900 runs. The overall performance of the receiver can be considered as follows: 99.89% (100% - 0.11%) probability of detection and 1.37% of false data for 7-b FFT and 99.87% (100% - 0.13%) probability of detection and 0.76% of false data for 8-b FFT. Thus, the 8-b output is slightly better than the 7-b output.

The performance of the monobit receiver can only be measured because the front end of the receiver is nonlinear

TABLE II  
RESULTS FROM TWO SIGNALS

Magnitude of 2nd Singal vs. 1st Signal (dB)	Found 1st Signal (%)		Found 2nd Signal (%)		Found Both Signals (%)		Found Neither Signal (%)		Found False Signal (%)	
	FFT		FFT		FFT		FFT		FFT	
	7-bit	8-bit	7-bit	8-bit	7-bit	8-bit	7-bit	8-bit	7-bit	8-bit
0	69.1	68.1	73.1	65.0	42.3	33.3	0.11	0.22	1.0	0.44
-1	82.6	82.7	58.1	47.2	40.9	30.1	0.22	0.22	1.3	0.56
-2	92.3	89.0	38.6	35.8	30.4	24.9	0.11	0.11	2.0	0.78
-3	94.9	96.1	26.7	19.2	21.2	15.3	0.0	0.00	1.7	1.44
-4	97.8	98.6	17.9	11.5	15.9	10.1	0.22	0.11	1.3	0.78
-5	99.2	99.5	11.7	5.11	15.9	4.7	0.0	0.11	0.89	0.56
average	89.3	89.0	37.7	30.60	27.8	19.7	0.11	0.13	1.37	0.76

TABLE III  
COMPARISON OF TYPICAL IFM AND MONOBIT RECEIVERS

PARAMETER	IFM receiver	Monobit receiver
Evaluation Method	Measured	Simulated
Sampling Rate (GHz)	NA	2.5
Points of FFT	NA	256
Bandwidth (GHz)	2 - 16	1
Sensitivity	medium	high
Number of Signals	1	2
Single Signal Dynamic Range (dB)	70	75
2 Signal Spurfree DR (dB)	NA	75
2 Signal Instantaneous DR (dB)	NA	4
Channel Bandwidth (MHz)	2000 - 16000	10
Frequency Accuracy (MHz)	1	6
Time Resolution (ns)	NA	100
Minimum Pulse Width (ns)	100	200

and the 2-b ADC is also highly nonlinear. The performance of a typical IFM receiver is known from existing operational hardware. Table III compares the performance of these two receivers.

#### IV. IMPLEMENTATION

In this monobit receiver, the analog signal is first sampled at 2.5 GHz and then converted to 2-b digital data. The bit stream is then demultiplexed by two 1-to-16 demultiplexers to

produce 32-b parallel data. These 32-b parallel data are then fed into the designed ASIC where the signals will be detected. Because the ASIC is doing a 256-point FFT, 256-point inputs will be required. Each point contains 2 b and, thus, a total of 512 b of input data. As the demultiplexer can only do 32 b of multiplexing at a time, demultiplexing needs to be done 16 times before all 512 b of input data can be obtained. Thus, a complete set of inputs will be available every 102.4 ns. Consequently, the ASIC processes the input data at such a rate too.

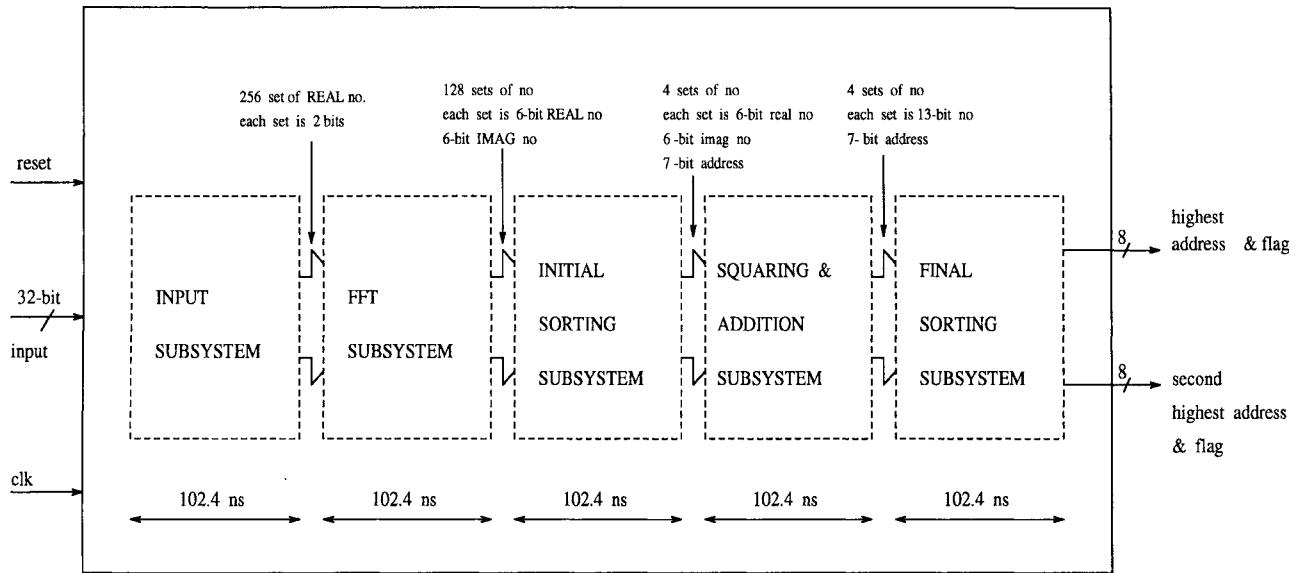


Fig. 3. Overall block diagram for the FFT chip.

The overall block diagram for the FFT chip is shown in Fig. 3. The inputs to the chip are 32-b data, a reset signal and an input clock. The outputs of the FFT chip consist of two sets of data (highest address and flag) shows the address of the signal with the highest peak. The flag indicates the validity of the address. The address is valid when the flag shows one. The second set of data (second highest address and flag) shows the address of the signal with the second highest peak. Similarly, its corresponding flag is used to indicate the validity of the address. This flag will be zero when there is no second signal that has amplitude close to the first one.

#### A. Overall Description of the FFT Chip

This section gives an overall description of the ASIC with explanation on the function of each subsystem. The detailed description of each subsystem will be covered in the subsequent sections. As mentioned in the earlier section, the 256 sets of inputs will be loaded about every 102.4 ns. Thus, in order to attain this speed, the whole chip is broken down into five different subsystems pipelined together. The processing in each subsystem will be completed within 102.4 ns with the results conveyed over to the following subsystem. The ASIC design consists of 5 pipelined subsystems shown in Fig. 3:

- 1) input subsystem;
- 2) FFT subsystem;
- 3) initial sorting subsystem;
- 4) squaring and addition subsystem;
- 5) final sorting subsystem.

The ASIC inputs (reset, clock (clk), and 32-b data) are directed into the input subsystem. The function of the input subsystem is to receive 32 b of parallel input data that flow in consecutively from the demultiplexer, store them, and finally produce 256 sets of real numbers for the FFT subsystem. Each

set of the number is a 2-b binary number. The other function of the subsystem is to produce a system clock to drive all the pipelined flip-flops in each stage. The subsystem also produces another three clocks (clk\_out1, clk\_out2, and clk\_out3) to be used in the initial sorting subsystem.

The main function of the FFT subsystem is to perform FFT on the 256 sets of input data. The results of the transform are 128 sets of output data. Each set of this output data is limited to a 7-b real number and a 7-b imaginary number (6-b magnitude and 1-b sign). Thus, after performing the absolute operation on these two 7-b numbers, the FFT generates two 6-b numbers. Actually, there should be 256 sets of output data; however, since the other 128 sets of the results are conjugate to these 128 sets of data, the conjugate sets are not used in order to save space. The outputs from this subsystem are fed into the initial sorting subsystem.

The main function of the initial sorting subsystem is to locate a maximum of four signals from the 128 sets of output data of the FFT subsystem that have the highest amplitudes. A physical circuit to sort all 128 signals would be very large and, therefore, not practical. Having found the highest signals, the addresses, real and imaginary numbers, and flag bits of these signals will be stored in registers.

With the data obtained from initial sorting subsystem, the squaring and addition subsystem will square the real and imaginary numbers of each set of data and these two results are added together within its own set. The maximum outputs of this subsystem are four sets of data available to be sent to the final sorting subsystem. Each set of these data consists of a 7-b address, flag bit, and 13-b computed result.

The function of the final sorting subsystem is to determine from its four sets of input data, the addresses of the two signals with the highest and second highest amplitudes. The outputs from this subsystems are a 7-b address and a flag bit for each of the highest and the second highest signals. If there is not any signal present, the two flag bits will be zero. Likewise,

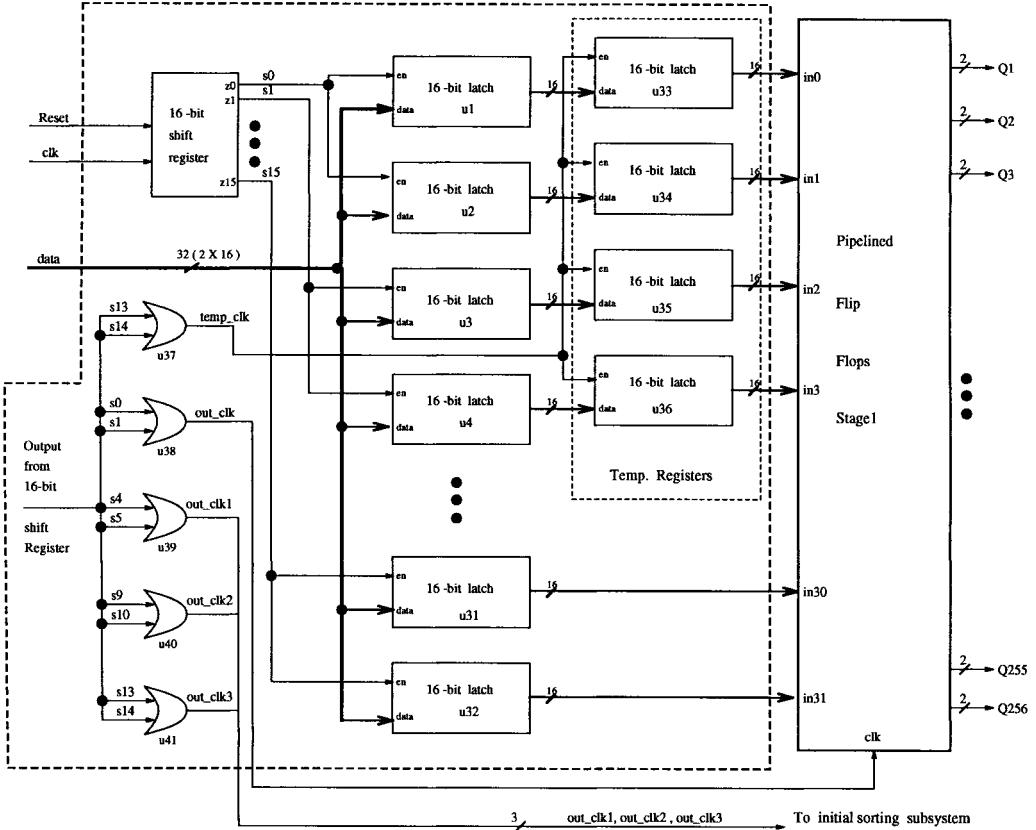


Fig. 4. Block diagram of the input subsystem.

if there is only one signal present, the second flag bit will be zero, indicating that there is only one signal.

### B. Input Subsystem

This section gives a description of the input subsystem, shown in Fig. 4. The inputs (reset, clock (clk), and 32-b data) to the chip are directed to the input subsystem. The main function of the input subsystem is to receive 32 b of parallel input data that consecutively flow in, store them, and finally produce 256 sets of real numbers for the FFT subsystem. Each set of the number is a 2-b binary number. The other function of this subsystem is to produce a system clock to drive all the pipelined flip-flops in each stage. The subsystem also produces three clocks (clk\_out1, clk\_out2, and clk\_out3) to be used in the initial sorting subsystem.

At the front end of the subsystem is a 16-b shift register. An one at its reset pin will reset all its outputs to zero except  $s_0$  which will be one. With the reset signal at zero and clock pulses going into this register, the one at  $s_0$  will be shifted to  $s_1$ , then  $s_2$  and so on until  $s_{15}$  and then back to  $s_0$  again. The 32-b input data from the external pins are connected simultaneously to 32 16-b latches ( $u_1$  to  $u_{32}$ ). However, only two of these latches will be enabled at a time by the enable signal from the 16-b shift register. At the end of 16 clock pulses, all the latches would have been loaded with input data consisting of a total of 512 b. These 512 b of data will have to be written into the flip-flops in the pipelined flip-flop

stage one before they are overwritten by the subsequent inflow data. The clock that drives this component is out\_clk produced when  $s_0$  or  $s_1$  is high. This out\_clk will also be used to synchronize all other pipelined flip-flops in other stages. As the out\_clk stretches for a duration from  $s_0$  to  $s_1$ , thus to prevent informations in latches  $u_1$  to  $u_4$  from being overwritten by the incoming data, the earlier informations will first be transferred over to temporary registers. This is done by a clock named temp\_clk at the time duration when  $s_{13}$  or  $s_{14}$  is high. The purpose of ORing  $s_0$  and  $s_1$  is to lengthen the pulsewidth of the out\_clk as it is the system clock to be used to drive all other pipelined flip-flops in the rest of the subsystems. Similarly, three other clocks, out\_clk1, out\_clk2, and out\_clk3, are generated and to be used in the initial sorting subsystem.

### C. FFT Subsystem

This section gives a description of the FFT Subsystem, shown in Fig. 5. The main function of the FFT subsystem is to perform the FFT on the 256 sets of input data. The result of the transform is a 128 sets of output data. Each set of this output data comprises of a 6-b real number and a 6-b imaginary number. The outputs from this subsystem are fed into the initial sorting subsystem.

There are nine levels of transformation to be done in this subsystem. Each level of transformation comprises of approximately 256 operations. As shown in Fig. 5, the operations are identified as  $A$  or  $C$ . All the  $C$  operations are either

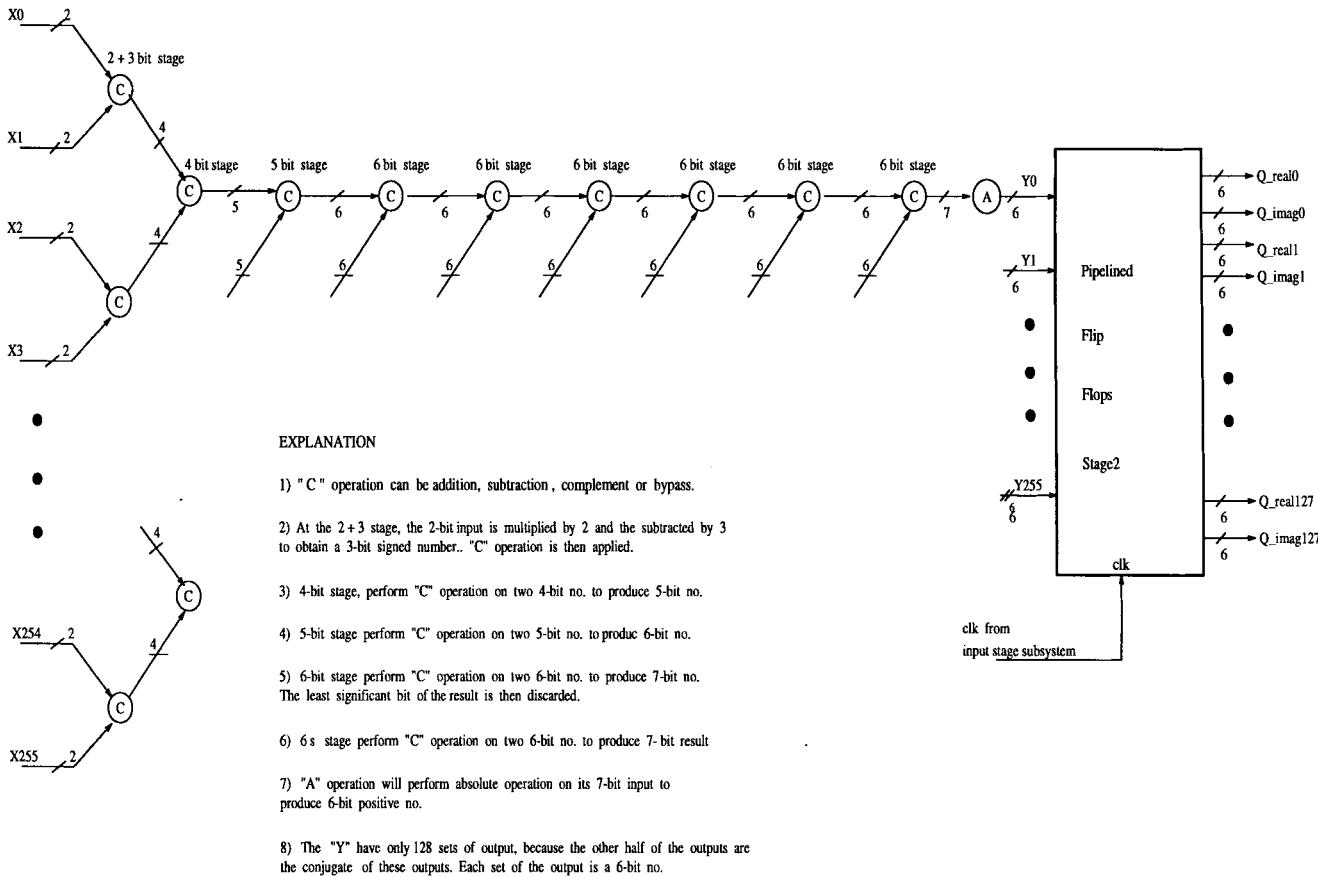


Fig. 5. Block diagram for the FFT subsystem.

TABLE IV

2-b Input	Coding Information
00	-3
01	-1
10	+1
11	+3

an addition or subtraction of two numbers. Beside this, it can also be a bypass, complement, or no operation. The operations are determined by the 256-point FFT architecture (no multiplication because the kernel function is 1 b).

The inputs to this subsystem are 256 set of data. Each set of data is 2 b. The codes of this 2-b data are as shown in Table IV.

The transformation of the 2-b input into the coded information is done at the first level namely the "2 + 3 b stage." Each of the 2-b inputs is first multiplied by two and then subtracted by three. The first level starts with a 2-b operation and produces 4-b results (see explanation in Fig. 5). It is then followed by the 4-, 5-, and 6-b operation stages at level two, three, and four, respectively. From level five until level eight, all operations are 7 b. In these levels, the inputs are 7 b. The results obtained after the operation are 8 b, which are then truncated to 7 b by discarding the least significant bit.

The last level (level nine) is slightly different in the sense that the operations produce 7-b results. These results are in two's complement form. In order to obtain an absolute number

at the output of the subsystem, an "absolute operations" stage denoted by *A* has been added after level nine. This stage converts all the 7-b results obtained from level nine into 6-b positive numbers.

The outputs from the FFT subsystem are 128 sets of data. Each set of this output data consists of a 6-b real number and a 6-b imaginary number. They are stored into flip-flops in the pipelined flip-flop stage two. Here the clock that does the latching is out\_clk from the input subsystem. These outputs are then fed to the initial sorting subsystem.

#### D. Initial Sorting Subsystem

This section gives a description of the initial sorting subsystem, as shown in Fig. 6. The main function of the initial sorting subsystem is to locate the addresses of a maximum of four signals from the 128 sets of output data of FFT subsystem that have the highest amplitudes. Only these signals will be squared and summed in the following operation instead of performing squaring and summation on all 128 outputs. The inputs to this subsystem are clk1, clk2, and clk3 from the input subsystem. Besides these signals, there are also 128 sets of 6-b real and imaginary data from the FFT subsystem. These 128 sets of 6-b real and imaginary data are first fed to 128 *S*-comparators (see Fig. 7). Here, the real and imaginary data are compared with two different threshold values, which are set at eight and four for 7-b FFT (set at 18 and 10 for 8-b FFT).

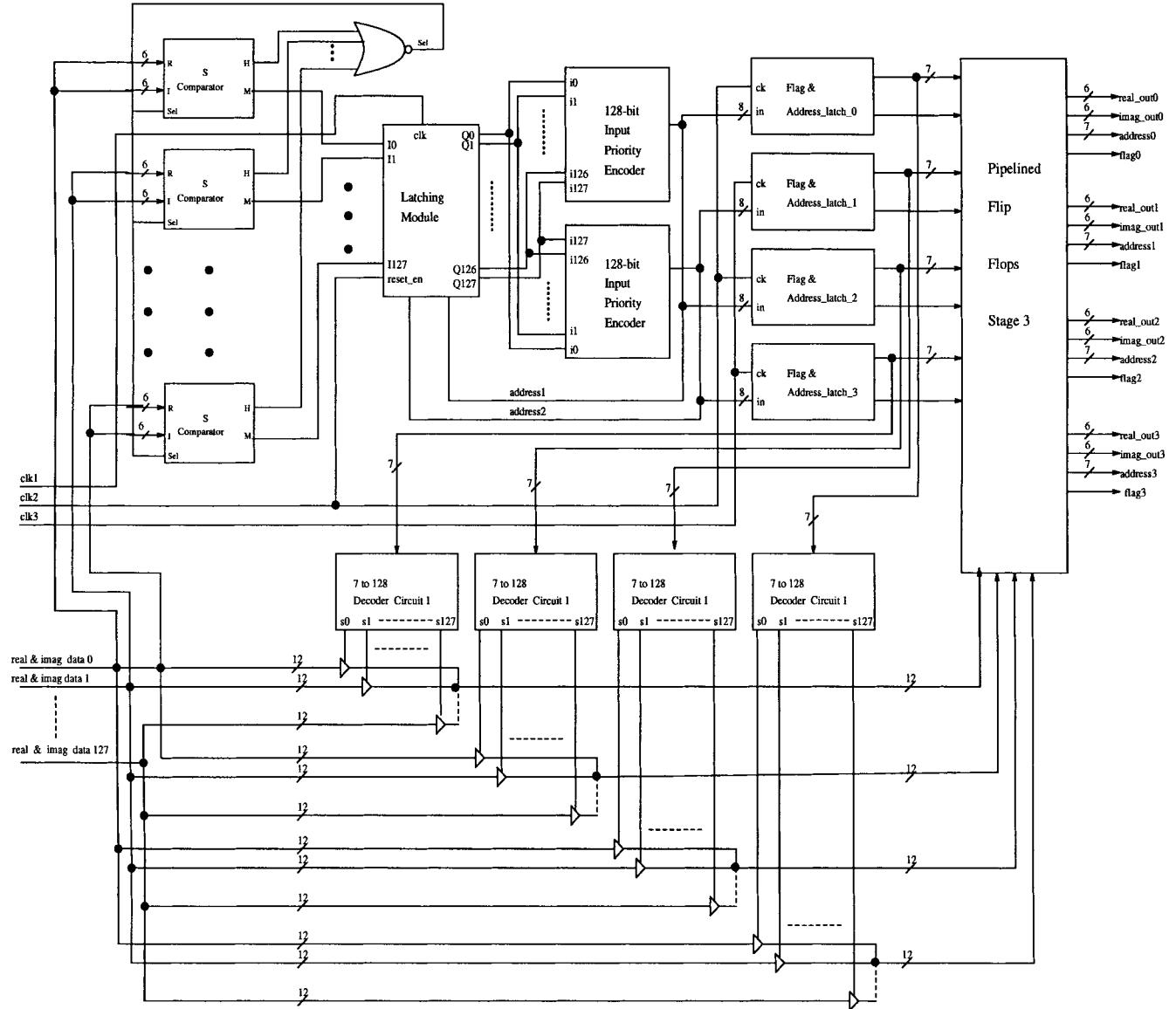


Fig. 6. Block diagram for the initial sorting subsystem.

In the following discussion, we use 7-b FFT as an example. The results of the comparisons are fed to inputs  $A$  and  $B$  of a multiplexer controlled by the  $Sel$  line. If any of the 128 sets of inputs, whether real or imaginary, exceeds the threshold value eight, the  $Sel$  line will be set to zero, outputting the high level comparison result through the multiplexer. However if none of the 128 sets of inputs exceeds the threshold value 8,  $Sel$  line will be set to one, outputting the low-level comparison result through the multiplexer. Therefore, the high threshold indication line of all the 128  $S$ -comparators are connected to an OR gate to produce  $Sel$  signal (see Fig. 6).

The reason of using two threshold levels is due to the non-linear effect of the RF front end. Two thresholds can increase probability of detection and also reduce false detection. The rule of using two thresholds is that if the high threshold is crossed, neglect the low one. If the high threshold is not crossed, use the low one. Fig. 8(a) shows one strong signal

cross the high threshold and a spur cross the low threshold. Under this condition, the high threshold is used for detection. The signal is detected and the spur is neglected. In Fig. 8(b), neither signal crosses the high threshold, but both signals cross the low one. Under this condition, the lower threshold is used for detection. If only the high threshold is used, the receiver will miss signals as shown in Fig. 8(b). If only the low threshold is used, the receiver will generate a false detection, as shown in Fig. 8(a).

The search for four highest signal is completed within two cycles with search for two per cycle. The outputs from the multiplexer of the 128  $S$ -comparators are latched into a latching module by  $clk1$ . The outputs from the latching module are fed into two 128-b input priority encoders. One encoder searches its inputs in ascending order from  $i0$  to  $i127$  and produces the address of the first active line it encounters. The other priority encoder searches its inputs in descending

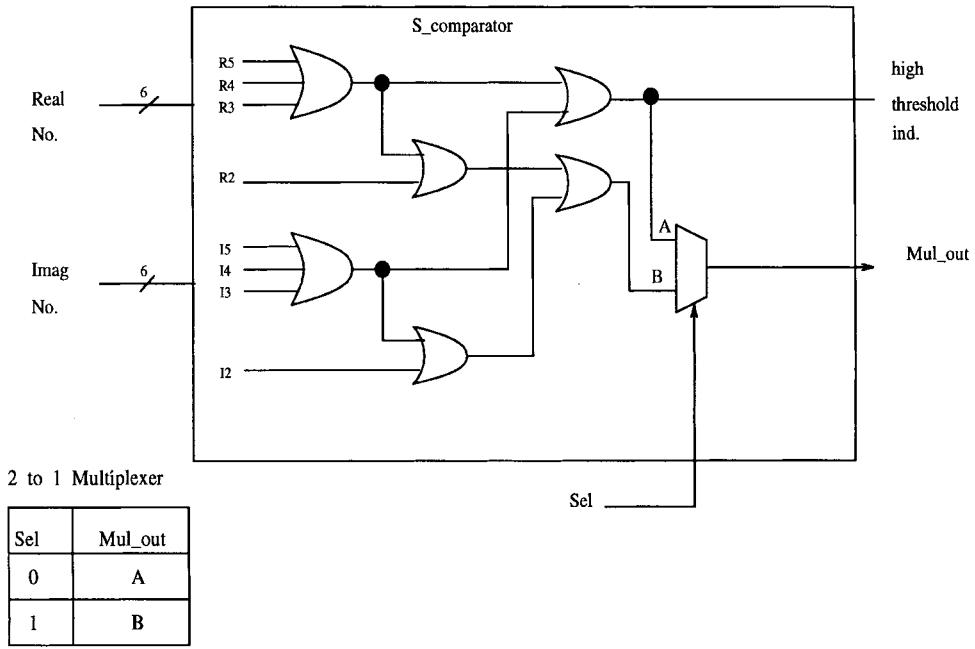
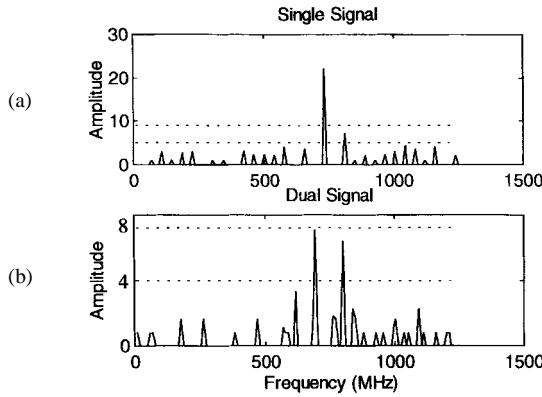
Fig. 7. Circuit diagram for the *S*-comparator.

Fig. 8. Single signal and dual signal detection. (a) Single signal. (b) Dual signal.

order from  $i127$  to  $i0$  and similarly produces the address of the first active line it encounters. The two addresses found and their flag signals are latched into the flag and address latch zero and one by  $clk2$ . During the same instances, these two addresses are also fed back to the latching module to clear the corresponding active lines that have already been encoded. That starts the second cycle of search for the next two highest signals. The next active lines in the two priority encoders will then be encoded into the next two addresses. This time they are, together with their flag bits, latched into the flag and address latch two and three by  $clk3$ . The addresses from the flag and address latches are then decoded by four 7-128 decoders, which consequently enable the selected tri-state buffers and allow the real and imaginary data, addresses, and flag bits of the four highest signals to be loaded into the flip-flops in the pipelined flip-flop stage three. The outputs from these flip-flops are fed to the squaring and addition subsystem.

#### E. Squaring and Addition Subsystem

This section gives a description of the squaring and addition subsystem, shown in Fig. 9. With the data obtained from the initial sorting subsystem, the squaring and addition subsystem will square the real and imaginary numbers of each set of data and these two results are added together within its own set. The output of this subsystem are four sets of data which are inputs to the final sorting subsystem. Each set of the data consists of a 7-b address, flag bit, and 13-b computed result.

From the block diagram in Fig. 9, it can be seen that the subsystem consists of four blocks of identical circuits. A detailed look at the block of circuit reveals that it consists of two squaring circuits to square the real and imaginary data. The obtained results are then added together in a 12-b adder to produce a 13-b result. No operation has been done on the address and flag lines coming into the subsystem. Eventually these addresses, flag bits, and computed results are latched into the flip-flops in the pipelined flip-flop stage four by  $out\_clk$  generated from the input subsystem. The outputs from these flip-flops are then fed to the final sorting subsystem.

#### F. Final Sorting Subsystem

This section gives a description of the final sorting subsystem, shown in Fig. 10. The function of the final sorting subsystem is to determine from its four-set input data the addresses of the two signals with the two highest amplitudes. The outputs from this subsystem are 7-b addresses and flag bits of the highest and the second highest signals. If there is not any signal present, the two flag bits will be zeros. Likewise, if there is only one signal present, the second flag bit will be zero, indicating that there is not any second highest signal.

From the block diagram, it can be seen that there exist four 13-b comparators. The four sets of input data from the squaring and addition subsystem are connected to two comparators  $U1$

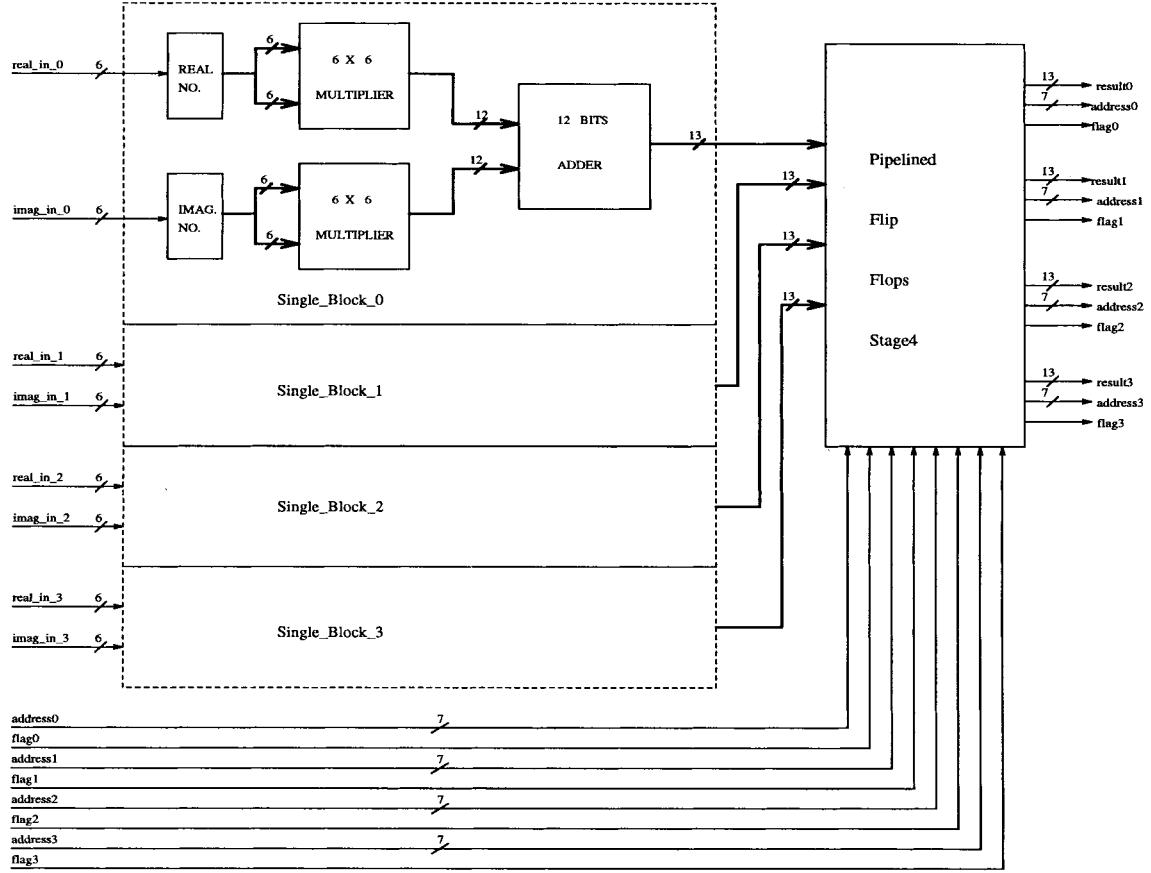


Fig. 9. Block diagram for squaring and addition subsystem.

and  $U2$ . First  $Z0$  and  $Z1$  are signals used to indicate the greater of the two input data in the comparators  $U1$  and  $U2$ .  $Y0$  and  $Y1$  are 2-1 multiplexers that allow only the greater input data from  $U1$  and the greater input data from  $U2$  to go into comparator  $U3$ . Comparator  $U3$  is used to find the highest signal of the four.  $Z2$  is the result of the comparator  $U3$ . Similarly,  $Y2$  and  $Y3$  are also 2-1 multiplexers. This time they are controlled by signals  $Z0$ ,  $Z1$ , and  $Z2$ . Selected input data will flow into comparator  $U4$ , which is used for detecting the second highest signal.  $Z3$  is the result of this comparator.

The value of  $Z0$ ,  $Z1$ ,  $Z2$ , and  $Z3$  are fed into a location encoder circuit (see Fig. 11). This circuit will produce five signals.  $W0$  and  $W1$  are signals used to enable the tri-state buffers for loading the highest signal address and flag bit to the flip-flops.  $S0$  and  $S1$  are signals used to enable the tri-state buffers for loading the second highest signal address and flag bit into the flip-flops. The selected signals output from the tri-state buffers are latched into the flip-flops in the pipelined flip-flop stage five and the outputs of these flip-flops are then connected to the output pins of the chip.

#### G. Layout and Simulation Results

The ASIC is designed using double-metal 0.5- $\mu$ m scalable CMOS technology and packaged in an 84-pin CPGA. The number of primary inputs and outputs of ASIC are 34 and 16, respectively. The ASIC is broken down into five different subsystems pipelined together and is estimated to perform at

TABLE V  
TRANSISTOR COUNT AND SILICON AREA OF EACH SUBSYSTEM

Subsystem	Transistor Count	Area (sq. um) after cell routing
Input stage	11,826	1,315,600
Flip-flop stage 1	10,266	995,315
FFT block	652,120	92,141,309
Flip-flop stage 2	35,966	3,630,434
Initial sorting	66,104	15,422,689
Flip-flop stage 3	2,890	441,668
Squaring and Addition	26,384	2,457,656
Flip-flop stage 4	1,928	193,404
Final sorting	5,138	481,500
Flip-flop stage 5	340	35,566
Total	812,962	117,115,050

a speed of 156.25 MHz. The chip contains about 812,931 transistors and has a die size of approximately 15 mm  $\times$  15 mm. The transistor count and silicon area after cell routing and optimization of each pipelined system are calculated and shown in Table V. The process in each subsystem is completed within 102.4 ns with the timing conveyed over to each subsystem as shown in Table VI. Two different simulators, Hspice and Compass Qsim, were used to perform the post-

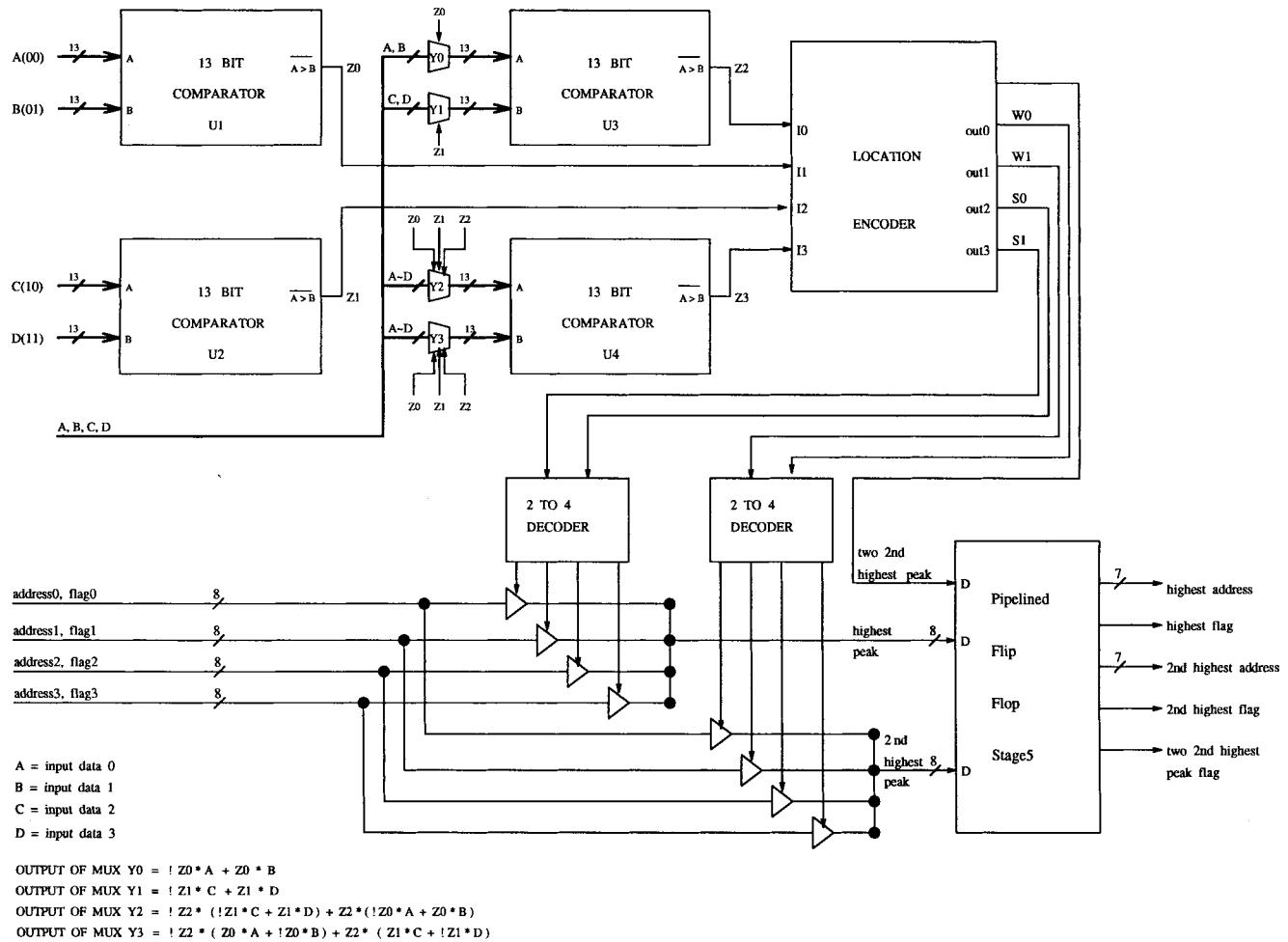


Fig. 10. Block diagram for the final sorting subsystem.

layout timing verification. The Hspice simulator, which is a transistor-level simulator, provides a more accurate timing analysis. However, its effective application is limited to small circuits. Thus, Hspice simulations were performed only on circuits whereby the timing is critical. As for the rest of the chip, Qsim simulations were performed. For this second-round simulation, the extracted parasitic capacitances and resistances of the routed netlists will be back-annotated to the Qsim simulator. Although timing analysis with Qsim is not as accurate as Hspice, it should be sufficient for timing verification after critical circuits have already been verified to meet the timing requirements. The design and performance statistics are summarized in Table VII.

## V. CONCLUSIONS

From the limited data collected, it appears that the monobit receiver can process two simultaneous signals. The performance of this monobit receiver compared with a typical IFM receiver is also presented in this paper. This receiver is designed to replace the existing IFM receivers which can process only one signal.

The simulation results of this monobit receiver should be improved through some logic circuit-design changes. A chip is being designed to take digitized data as input and

TABLE VI  
TIMING ANALYSIS OF EACH SUBSYSTEM. NOTE: THE TIMING ANALYSIS INCLUDES THE DELAY OF EACH PIPELINED FLIP-FLOPS

Subsystem	Critical Path (ns)
Input stage	99.50
FFT block	48.02
Initial sorting	90.11
Squaring and Addition	28.95
Final sorting	34.42

perform the monobit FFT. The chip also includes the frequency selection logic to select the correct input frequencies and avoid picking up spurious responses. The monobit receiver hardware including the ADC, demultiplexers and ASIC will be initially implemented as a proof-of-concept printed circuit board. A future iteration envisions implementation as a single multichip module. The overall performance can only be obtained when the receiver is built in hardware. The results are expected to have major practical impact in receiver systems as well as in other applications.

## LOCATION ENCODER ALGORITHM

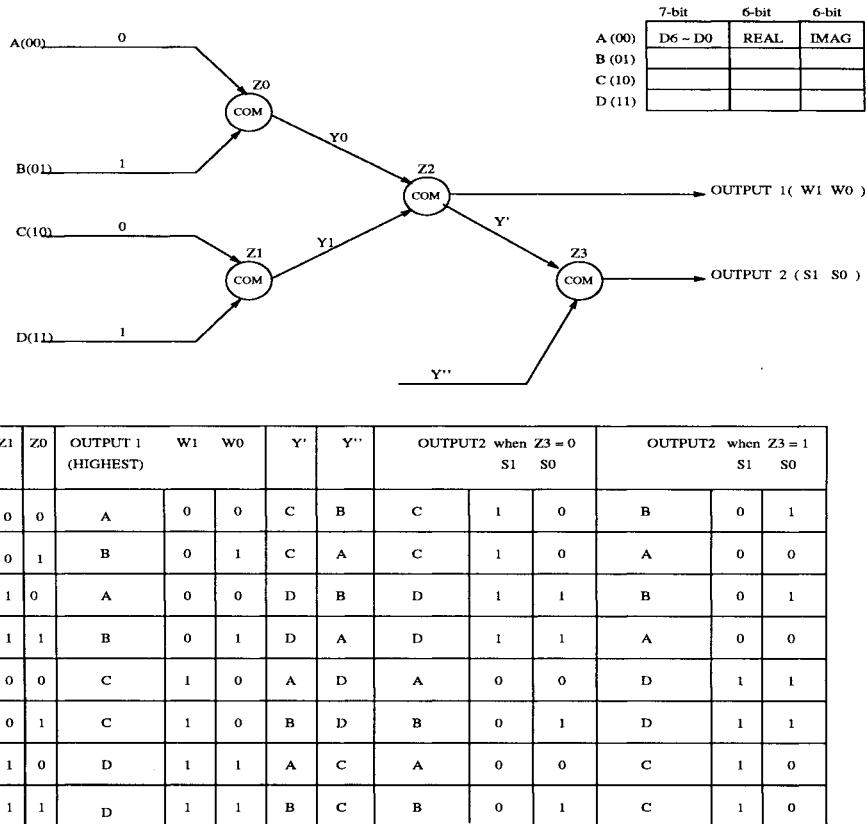


Fig. 11. Logic derivation for the location encoder.

TABLE VII  
DESIGN AND PERFORMANCE STATISTICS

Technology	0.5 $\mu$ m CMOS
Transistors	812,931
Die size	15 mm x 15 mm
Total I/O pins	84 CPGA
Power supply	5 V
Clock rate	156.25 MHz
Input data rate	5 Gb/s
Output data rate	156.25 Mb/s
Power dissipation	4.2 W

Several technical issues are currently under investigation to improve this monobit receiver. For example, the detection threshold settings need additional study, since the receiver will miss both signals if neither crosses threshold. The current overall performance of the receiver, as shown by simulation experiments, is 99.89% probability of detection and 1.37% of

false data for 7-b FFT, and 99.87% probability of detection and 0.76% of false data for 8-b FFT.

## REFERENCES

- [1] J. B. Y. Tsui and D. M. Akos, "Comparison of direct and downconverted digitization in GPS receiver front end designs," in *IEEE Int. Microwave Symp. Dig.*, vol. 3. San Francisco, CA, June 17-21, 1996, pp. 1343-1346.
- [2] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, and P. D. Welsh, "What is the fast Fourier transform?," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, pp. 45-55, June, 1967.
- [3] J. B. Y. Tsui, *Digital Techniques for Wideband Receivers*. Norwood, MA: Artech House, 1995.



**David S. K. Pok** was born in Singapore. He received the B.S.E.E. and M.S.E.E. from Wright State University, Dayton, OH, in 1995 and 1996, respectively.

Since 1996, he has worked as a Research Associate in the Department of Electrical Engineering, Wright State University, where he is involved with VLSI design.



**Chien-In Henry Chen** (S'89–M'89) received the B.S. degree from National Taiwan University, Taiwan, R.O.C., in 1981, the M.S. degree from the University of Iowa, Iowa City, in 1986, and the Ph.D. degree from the University of Minnesota at Minneapolis St. Paul, in 1989, all in electrical engineering.

He is currently an Associate Professor in the Department of Electrical Engineering, Wright State University, Dayton, OH. He has been actively involved in CAD, simulation, and testing of VLSI

systems, specifically in design for testability, built-in self-test, high-level design synthesis of digital systems, signal processing and microwave circuits, and fault-tolerant multicomputer systems. He is the principal investigator of several contracts and grants in these areas. He has authored over 45 publications in professional journal and conference proceedings.

Dr. Chen has served as a technical committee member of 1995 IEEE International ASIC Conference and Exhibit and was an invited speaker at the 6th VLSI Design/CAD Symposium in 1995.

**John J. Schamus** was born in Jersey City, NJ. He received the B.S.E.E. degree from Wright State University, Dayton, OH, in 1994, and is currently working toward the M.S.E.E. degree.

Since 1988, he has worked for Wright Laboratory, Wright Patterson Air Force Base, OH, as a Contractor with SAIC and SRL, where he has worked in the area of electronic-warfare analysis, infrared missile countermeasures simulation, and digital receivers.



**Christine T. Montgomery** received the B.S. degree in computer engineering and the M.E. degree in electrical engineering from Iowa State University, Ames, in 1992 and 1994, respectively.

Since 1994, she has served as an Officer in the U.S. Air Force, assigned to Wright Laboratory's Radio-Frequency Division, OH. Her research interests include passive radar and missile warning, signal and image processing, and hardware implementations of neural networks and parallel processors.

**James B. Y. Tsui** (S'61–M'75–SM'88–F'91) was born in Shantung, China. He received the B.S.E.E. degree from National Taiwan University, Taiwan, R.O.C., in 1957, the M.S.E.E. degree from Marquette University, Milwaukee, WI in 1961, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana, in 1965.

From 1965 to 1973, he was Assistant Professor and then Associate Professor in the Electrical Engineering Department, University of Dayton, Dayton, OH. Since 1973, he has been an Electronic Engineer at Wright Laboratory, Wright Patterson Air Force Base, OH. His work is mainly involved with electronic-warfare receivers, and his recent research is on digital microwave receivers. He has authored four books on microwave and digital receivers, and has published over 60 technical papers. He holds approximately 30 patents.