

REPORT DOCUMENTATION PAGE

J
188

Public reporting burden for this collection of information is estimated to average 1 hour per response, and maintaining the data needed, and completing and reviewing the collection of information. Send information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

OSIO

sources, gathering
/ this collection of
vis Highway, Suite

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 29 May 1998	3. REPORT TYPE AND DATES COVERED Final (01 Aug 96 - 15 Mar 98)	
4. TITLE AND SUBTITLE FFT Based VLSI Digital One Bit Electronic Warfare Receiver		5. FUNDING NUMBERS F49620-96-1-0423	
6. AUTHORS Professor Chien-In Henry Chen			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Wright State University, Electrical Engineering Dept. 3640 Colonel Glenn Hwy. Dayton, OH 45435-0001		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM 110 Duncan Avenue, Room B-115 Bolling Air Force Base, DC 20332-8080		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) 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 multi-chip module (MCM). The receiver consists of three major elements: 1) a nonlinear radio frequency (RF) front-end, 2) a signal sampler and formatting system (analog-to-digital converter (ADC) and demultiplexers), and 3) 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. The receiver uses a two-bit ADC operating 2.5 GHz whose outputs are collected and formatted by demultiplexers for presentation to the ASIC. The ASIC has two basic functions: 1) perform a fast Fourier transform (FFT), and then 2) determine the number of signals and report their frequencies. The ASIC design contains five stages: 1) the input, 2) the FFT, 3) the initial sort, 4) the squaring and addition, and 5) the final sort. The chip will process the ADC outputs in real time, reporting detected signal frequencies every 100 ns.			
<div data-bbox="699 785 1149 915" style="border: 1px solid black; padding: 5px; display: inline-block;"> DISTRIBUTION STATEMENT A <small>Approved for public release; Distribution Unlimited</small> </div>			
14. SUBJECT TERMS monobit, ASIC, ADC, FFT		15. NUMBER OF PAGES	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

19980630 046

FINAL REPORT
SUBMITTED TO
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
BOLLING AFB, DC 20332-6448

Date Submitted: May 29, 1998

Title: FFT Based VLSI Digital One Bit Electronic Warfare Receiver

Contract No.: F49620-96-1-0423

Institution Name: Wright State University

Street Address: Colonel Glenn Highway

City,State,Zip Code: Dayton, Ohio 45435

Principle Investigator: Chien-In Henry Chen

Department: Electrical Engineering

Telephone: (937) 775-5056

Fax: (937) 775-5009

E-mail: cihchen@cs.wright.edu

Contents

1 INTRODUCTION	2
2 MONOBIT RECEIVER	2
2.1 RF Front End	2
2.2 Signal Sampler and Formatting System	3
2.3 Signal Detection and Frequency Measurement	4
2.3.1 FFT Design	4
2.3.2 Frequency Selection Logic	4
3 EXPERIMENTAL RESULTS	5
4 IMPLEMENTATION	7
4.1 Overall Description of the FFT Chip	8
4.2 Input Subsystem	9
4.3 FFT Subsystem	11
4.4 Initial Sorting Subsystem	12
4.5 Square and Addition Subsystem	13
4.6 Final Sorting Subsystem	15
5 DESIGN AND SIMULATION RESULTS	16
6 CONCLUSIONS	20
7 Acknowledgements	21

FINAL REPORT

FFT BASED VLSI DIGITAL ONE BIT ELECTRONIC WARFARE RECEIVER

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 multi-chip module (MCM). The receiver consists of three major elements: 1) a nonlinear radio frequency (RF) front-end, 2) a signal sampler and formatting system (analog-to-digital converter (ADC) and demultiplexers), and 3) 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. The receiver uses a two-bit 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: 1) perform a fast Fourier transform (FFT), and then 2) determine the number of signals and report their frequencies. The ASIC design contains five stages: 1) the input, 2) the FFT, 3) the initial sort, 4) the squaring and addition, and 5) the final sort. The chip will process the ADC outputs in real time, reporting detected signal frequencies every 100 ns.

1 INTRODUCTION

The characteristics of instantaneous frequency measurement (IFM) receivers make them potentially suitable for electronic warfare (EW) applications. This kind of receiver has a wide instantaneous radio-frequency (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 nsec 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 accomplished.

This report presents a very simple digital receiver design which can cover 1 GHz bandwidth and process two simultaneous signals. This design uses a fast Fourier transform (FFT) to obtain frequencies on only 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 signal 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 report presents: 1) technical approach to design the receiver ASIC, 2) experimental results, and 3) performance comparison with a conventional digital receiver. This receiver is designed to replace existing IFM receivers which can process only one signal.

2 MONOBIT RECEIVER

The design of this receiver can be divided into three areas shown in Figure 1. They are the radio frequency (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.

2.1 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 1.375-2.375 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.

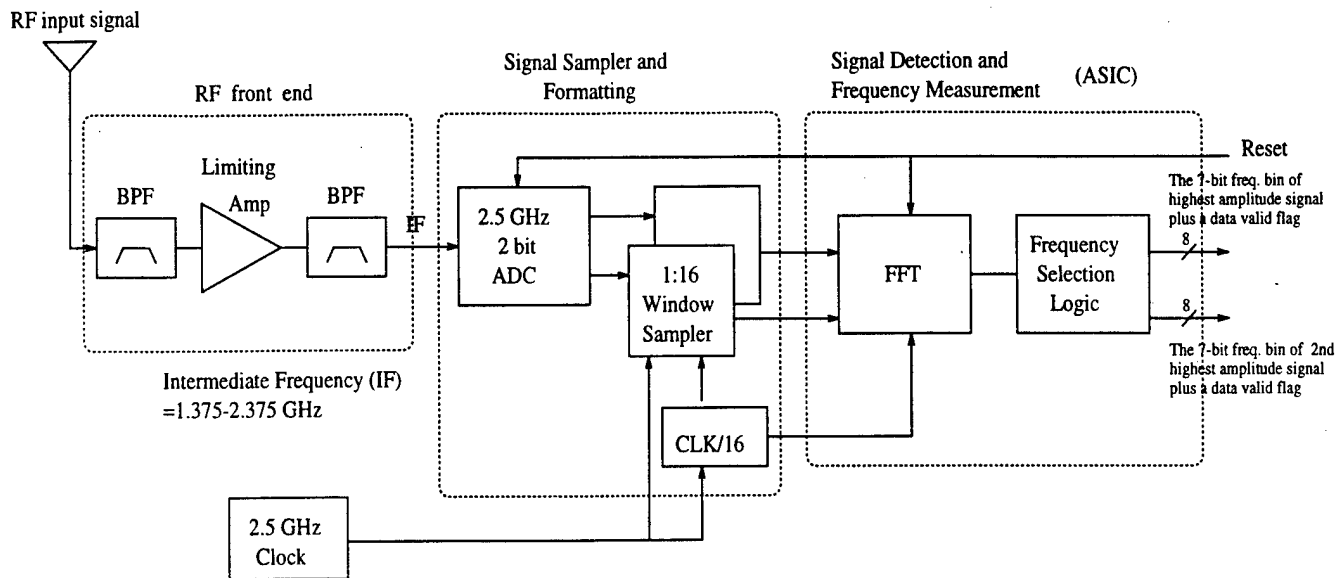


Figure 1: Three areas of monobit receiver

2.2 Signal Sampler and Formatting System

Because the signal from the limiting amplifier has a constant amplitude, a two bit ADC will be satisfactory. Experimental results showed that a two bit ADC is better than one bit, but three or more bits show very little improvement because of the limiting amplifier and the unique FFT design discussed in the next section. In order to cover 1 GHz bandwidth, the ADC should operate at about 2.5 GHz. The two lowest unambiguous ranges are from 0-1.25 and 1.25-2.5 GHz. A 1 GHz portion (1.375-2.375 GHz) from the second unambiguous frequency range is selected as the input bandwidth. The second band pass 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 nsec to produce 2-bit 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. The windowing circuitry thus has two key functions: 1) converts the serial data stream to parallel and 2) 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 algorithm chip 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.

2.3 Signal Detection and Frequency Measurement

2.3.1 FFT Design

This is the key component to 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 one bit (monobit), i.e. +1 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 +1 or -1 and +j 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 about 100 nsec which can be considered as the minimum pulse width. 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 1.250 MHz and the video bandwidth. In order to further simplify the design, the adders are limited to a maximum of 7 bits (6 bit amplitude and 1 bit sign). If the outputs from the adders are beyond 7 bits, they will be truncated to 7 bits.

The FFT ASIC inputs are two 16-bit data windows at a rate of 156.25 MHz. The input stage receives and stores each 16-bit data window until 16 windows have been collected, i.e., total data is (16 windows) x (16 data samples/window) = 256 data samples. Thus a complete data set is ready every (16 windows) x (6.4 nsec per window) = 102.4 nsec where the window sampling circuitry is feeding the FFT chip every (0.4 nsec ADC sampling rate) x (16 samples) = 6.4 ns. Therefore each stage of the pipeline is being designed to a maximum of 100 nsec worst-case processing.

2.3.2 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 two bit ADC. These two requirements simplify the logic frequency 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: 1) the 7-bit frequency bin of the highest amplitude signal plus a data valid flag, and 2) the 7-bit frequency bin of the second highest amplitude signal plus a data valid flag.

3 EXPERIMENTAL RESULTS

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

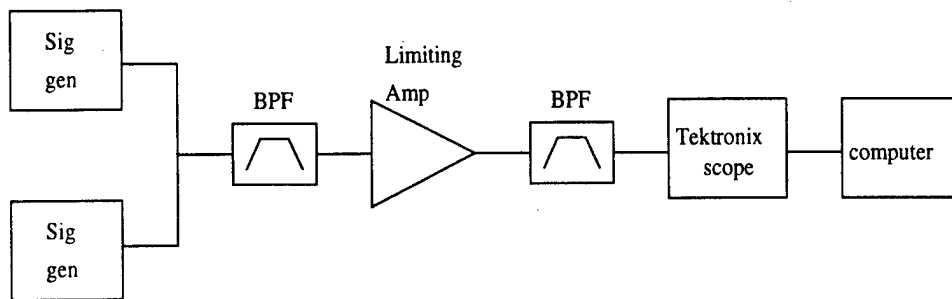


Figure 2: Experimental set-up

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 bit output. The 8 bit outputs were converted to 2 bits through a software program. These 2 bit data were processed through a one bit kernel function simulated in a computer program. The maximum number of output bits of the adders was limited to 7 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 8 and 4 for 7-bit FFT (18 and 10 for 8-bit FFT). Eight bit outputs were also used in the simulation to check the difference with 7 bits.

First, no signal was applied to the input, and the program was run to detect false alarm. For 350,000 runs there is no false alarm, but this only represents 35 ms ($350000 \times 100 \times 10^{-9}$) in real testing time. Second, one signal with random frequency was applied to the input of the set-up 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 1. 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.

Table 1: 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.

The minimum frequency separation was 20 MHz (slightly wider than 2 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. At each of these power levels the second signal changed from 0 to -5dB with respect to the first one. At each combination of power levels 100 runs were taken. The results are shown in Table 2. The receiver usually read both frequencies correctly when the two signals are close in amplitude.

Table 2: 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

Sometimes the receiver misses both signals, because neither signal crosses the threshold. Sometimes, the receiver read 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-bit FFT and 99.87% (100%-0.13%) probability of detection and 0.93% of false data for 8-bit FFT. Thus, the 8-bit output is slightly better than the 7-bit output.

The performance of the monobit receiver can only be measured, because the front end of the receiver is nonlinear and the 2 bit ADC is also highly nonlinear. The performance of a conventional digital receiver can be predicted from the ADC performance and the FFT capability. Assume that the ADC has 8 bits or more and operates at 3 GHz. The performance of the two receivers is listed in Table 3.

Table 3: Comparison of conventional digital 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	100

4 IMPLEMENTATION

In this monobit receiver, the analog signal is first sampled at 2.5 GHz and then converted to 2-bit digital data. The bit stream is then demultiplexed by two 1-to-16 demultiplexers to produce 32-bit parallel data. These 32-bit parallel data are then fed into the designed FFT chip where the signals mentioned in the preceding paragraph will be detected. Because the FFT chip is doing a 256-point fast Fourier transform, 256-point inputs will be required. Each point contains two bits and thus a total of 512 bits of input data. As the demultiplexer can only do 32 bits of multiplexing at a time, demultiplexing needs to be done 16 times before all 512 bits of input data can be obtained. Thus a complete set of input will be available about every 100 ns. Consequently, the FFT chip would have to process the input data at such a rate too.

The overall block diagram for the FFT chip is shown in Figure 3. The inputs to the chip are 32-bit data, a reset signal and an input clock. The outputs of the FFT chip consist of two sets of data. The first set of data (highest address & 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 '1'. The second set of data (second highest address & 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 '0' when there is no second signal that has

amplitude close to the first one.

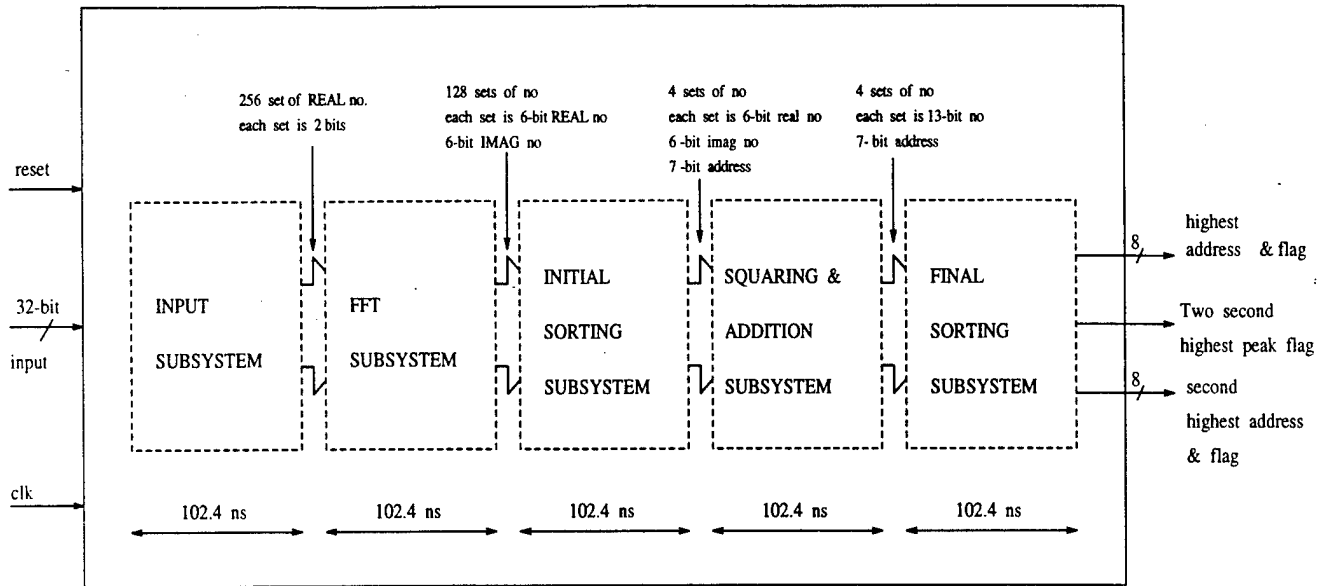


Figure 3: Overall block diagram for the FFT chip

4.1 Overall Description of the FFT Chip

This section gives an overall description of the FFT chip 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 100 nsec. 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 100 nsec with the results conveyed over to the following subsystem. As shown in Figure 3, the whole chip is made of 5 pipelined subsystems.

- input subsystem
- FFT subsystem
- initial sorting subsystem
- Squaring & Addition subsystem
- Final Sorting subsystem

The inputs (reset, clk and 32-bit data) to the chip are directed into the input subsystem. The main function of the input subsystem is to receive 32 bits 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 bit binary number. The other function of the subsystem is to produce a system clock (clk) to drive all the pipelined flip-flops in each stage. The subsystem also produce 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 the fast Fourier transform on the 256 sets of input data. The results of the transform are 128 sets of output data. Each set of this output data consists of a 7-bit real number and a 7-bit imaginary number (6 bit magnitude and 1 bit sign). So after performing the absolute operation on these two 7-bit numbers the FFT generates two 6-bit numbers. Actually there should be 256 sets of output data, however because the other 128 sets of the results are imaginary conjugate to these 128 sets of data and are not used, thus to save chip area they are not included. 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, the real and imaginary numbers, and the flag bits of these signals will be stored in registers.

With the data obtained from initial sorting subsystem, the squaring & 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-bit address, a flag bit and a 13-bit 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-bit address and a flag bit for each of the highest and the second highest signals. If there isn't any signal present, the two flag bits will be zero. Likewise, if there is only one signal present, the second flag bit will be zero indicating that there is only one signal.

4.2 Input Subsystem

This section gives a description of the input subsystem, shown in Figure 4. The inputs (reset, clk and 32-bit data) to the chip are directed to the input subsystem. The main function of the input subsystem is to receive 32 bits of parallel input data that flow in consecutively, store them and finally produce 256 sets of real numbers for the FFT subsystem. Each set of the number is a 2 bit binary number. The other function of this subsystem is to produce a system clock (clk) 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-bit shift register. An '1' at its reset pin will reset all its outputs to '0' except s0 which will be '1'. With the reset signal at '0' and clock pulses going into this register, the

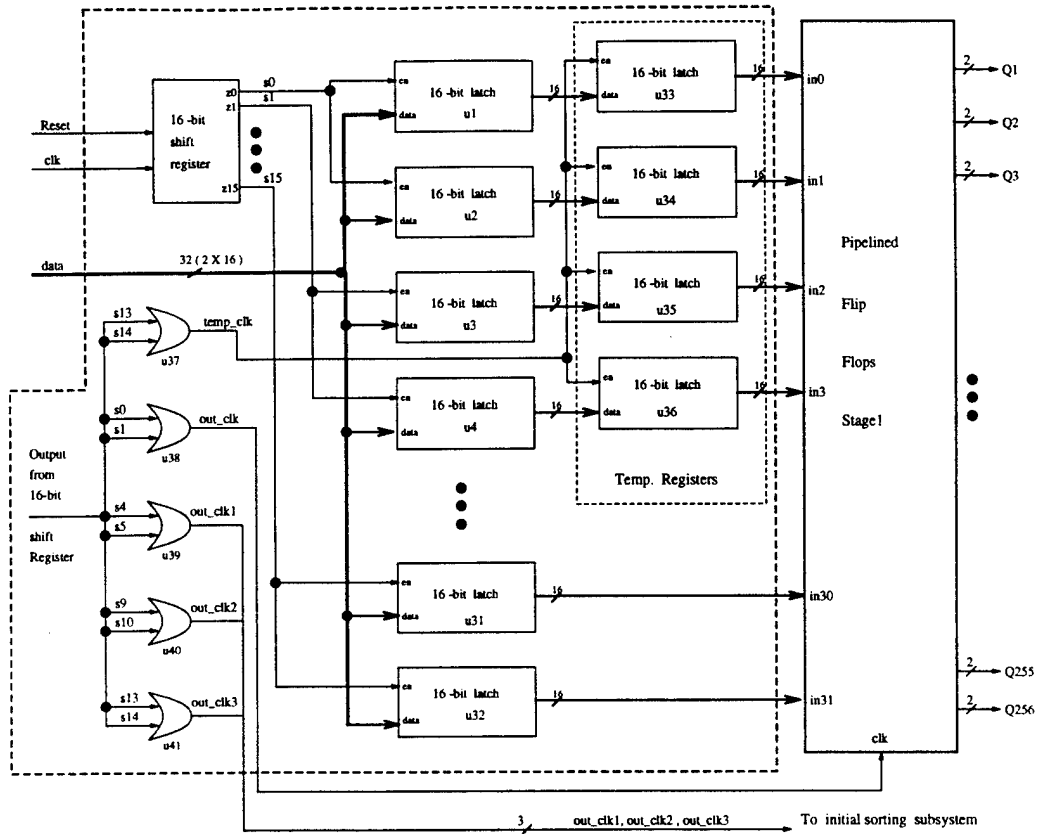


Figure 4: Block diagram of the input subsystem

'1' at s0 will be shifted to s1, then s2 and so on till s15 and then back to s0 again. The 32-bit input data from the external pins are connected simultaneously to 32 16-bit latches (u1 to u32). However, only two of these latches will be enabled at a time by the enable signal from the 16-bit 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 bits. These 512 bits of data will have to be written into the flip-flops in the pipelined flip flop stage 1 before they are overwritten by the subsequent inflow data. The clock that drives this component is out_clk produced when s0 or s1 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 s0 to s1, thus to prevent informations in latches u1 to u4 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 s13 or s14 is high. The purpose of ORing s0 and s1 is to lengthen the pulse width 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.

4.3 FFT Subsystem

This section gives a description of the FFT Subsystem, shown in Figure 5. The main function of the FFT subsystem is to perform the fast Fourier transform 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-bit real number and a 6-bit 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 about 256 operations. As shown in Figure 5, the operations are identified as "A" or "C". All the "C" operations are either 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 one bit).

The inputs to this subsystem are 256 set of data. Each set of data is 2 bits. The codes of this 2-bit data are as follows.

2-bit input	Coding Information
00	-3
01	-1
10	+1
11	+3

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

The last level (level 9) is slightly different in the sense that the operations produce 7-bit results. These results are in 2'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 9. This stage converts all the 7-bit results obtained from level 9 into 6-bit positive numbers.

The outputs from the FFT subsystem are 128 sets of data. Each set of this output data consists of a 6-bit real number and a 6-bit imaginary number. They are stored into flip-flops in the pipelined flip flop stage 2. Here the clock that does the latching is out_clk from the input subsystem. These outputs are then fed to the initial sorting subsystem.

indication line of all the 128 S-comparators are connected to an OR gate to produce *Sel* signal (see Figure 6).

The reason of using two threshold levels is due to the nonlinear 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. Figure 8(a) shows one strong signal crosses the high threshold and a spur crosses the low threshold. Under this condition, the high threshold is used for detection. The signal is detected and the spur is neglected. In Figure 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 Figure 8(b). If only the low threshold is used, the receiver will generate a false detection as shown in Figure 8(a).

The search for four highest signal is completed within 2 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-bit 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 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 & address latch 0 and 1 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 together with their flag bits are latched into the flag & address latch 2 and 3 by *clk3*. The addresses from the flag & address latches are then decoded by four 7 to 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 3. The outputs from these flip-flops are fed to the squaring & addition subsystem.

4.5 Square and Addition Subsystem

This section gives a description of the Square and Addition Subsystem, shown in Figure 9. With the data obtained from the initial sorting subsystem, the squaring & 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-bit address, a flag bit and a 13-bit computed result.

From the block digram in Figure 9, it can be seen that the subsystem consists of four blocks of identical

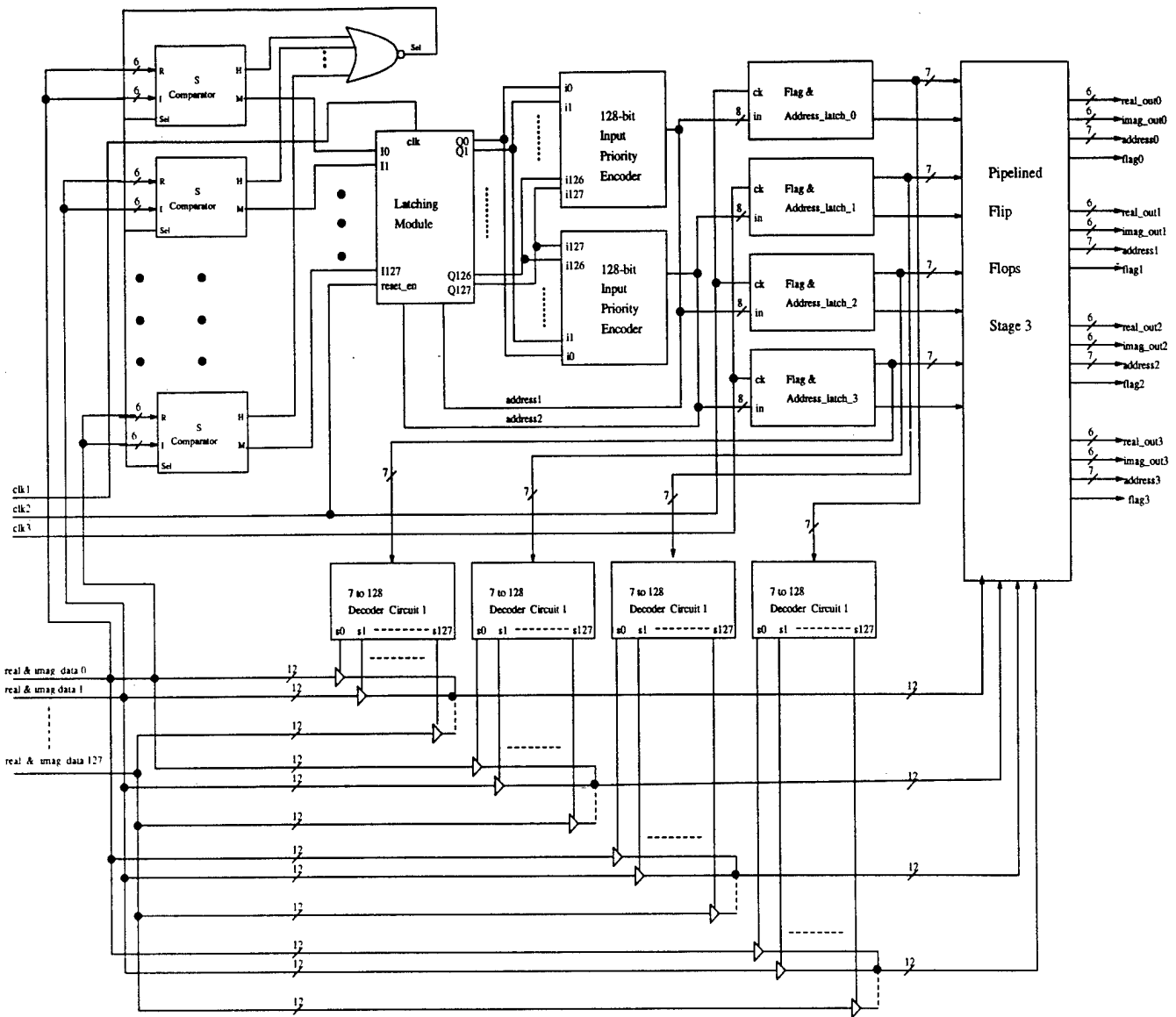


Figure 6: Block diagram for the initial sorting subsystem

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-bit adder to produce a 13-bit 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 4 by out_clk generated from the input subsystem. The outputs from these flip-flops are then fed to the final sorting subsystem.

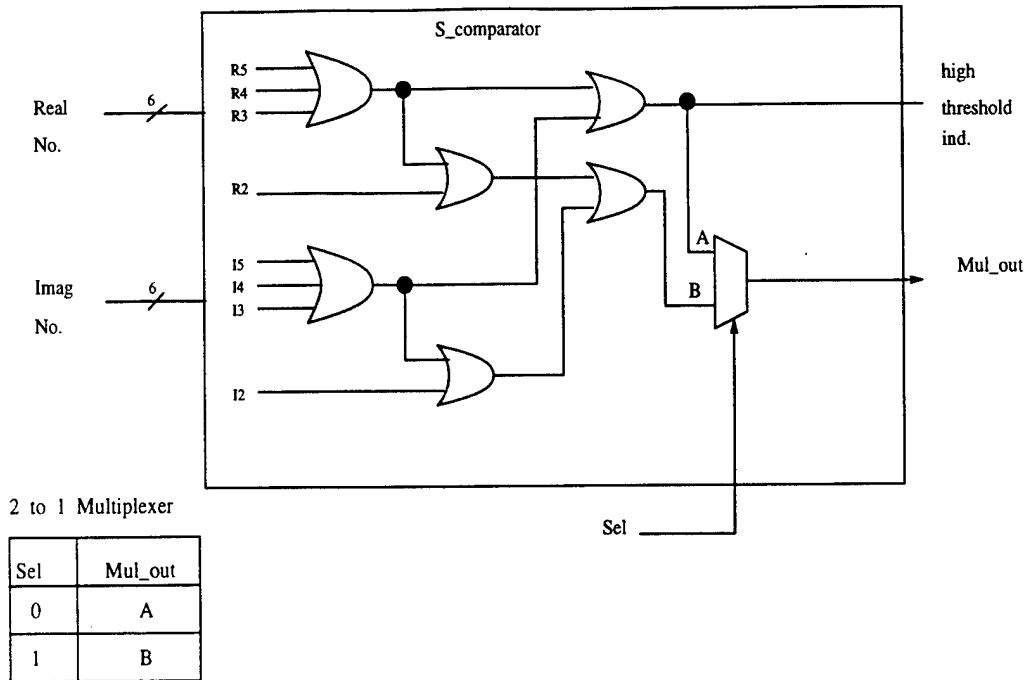


Figure 7: Circuit diagram for the S comparator

4.6 Final Sorting Subsystem

This section gives a description of the final sorting subsystem, shown in Figure 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 highest two amplitudes. The outputs from this subsystem are 7-bit addresses and flag bits of the highest and the second highest signals. If there isn't 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 isn't any second highest signal.

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

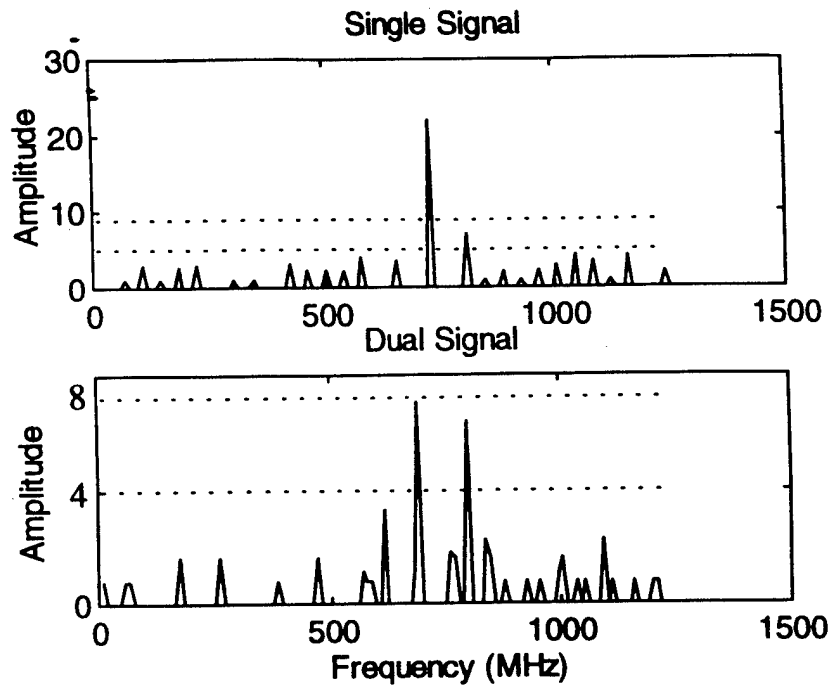


Figure 8: Single signal and dual signal detection:(a)single signal (b)dual signal

buffers are latched into the flip-flops in the pipelined flip flop stage 5 and the outputs of these flip-flops are then connected to the output pins of the chip.

5 DESIGN AND SIMULATION RESULTS

The design flow for the ASIC is shown in Figure 12. The design started at concept development stage. The developed concepts were then transformed into an abstract model in Matlab for verification. Following this, numerous Matlab simulations were done to prove and verify the theoretical concepts. When the abstract model was verified correct, the Matlab programs were then modified to a more physical-related model that could be followed and implemented in the physical chip design. For example, the physical model tracked the number of bits carried through the computation process. The physical model was then simulated and compared with the first abstract model. The actual physical chip design was based on the later model.

Designing of a chip can be done using traditional customized layout or automatic layout approaches. Due to initial approximation of the size and the complexity of the chip, automatic layout approach was considered more appropriate and chosen. Automatic layout method not only reduces tedious manual layout of components in the chip, it also enables easy modification to the design which is likely to occur in the first phase of the

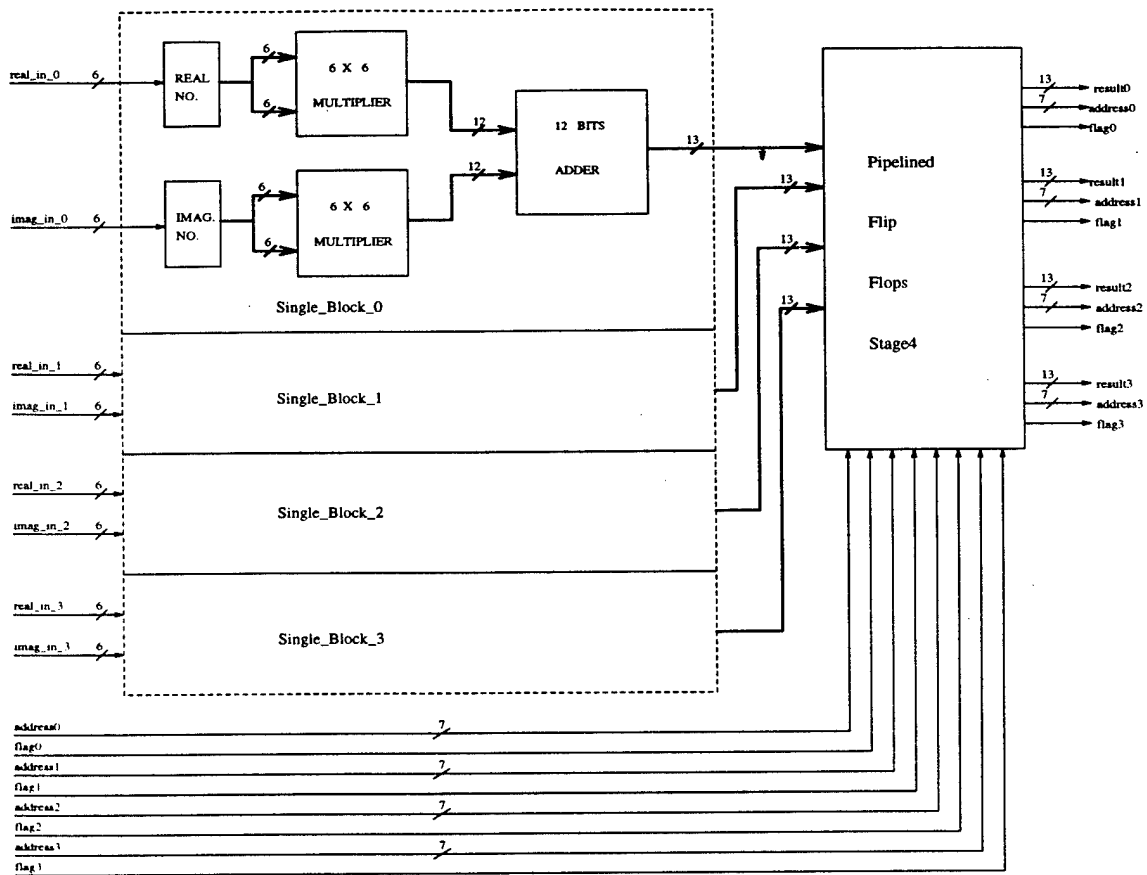


Figure 9: Block diagram for squaring and addition subsystem

design. It also facilitates design for future improved versions of the chip.

VHDL was used to design the chip. The behavior of each module in the chip was described in VHDL and then simulated with VHDL simulation tools. Modules were first combined to subsystems and then to the whole chip. At chip level, the design was again simulated to verify with the original Matlab simulations. After this step is done, the circuits are synthesized.

Synthesis is an automatic method of converting register transfer level (RTL) descriptions to gate-level netlists. These gate-level netlists consist of interconnected gate-level macro cells. Models for the gate-level cells are described in the technology libraries. The synthesis tools optimize the gate-level netlists for area, speed, and testability, etc. The synthesis process is shown in Figure 13. The inputs to the synthesis process are RTL descriptions, circuit constraints and attributes for the design, and a technology library. The synthesis process produces an optimized gate-level netlist from all these inputs. The synthesis tools that are used in this chip design are Synopsys VHDL Compiler and Design Compiler. Synthesis on the chip was done using the bottom-up approach. The generated outputs are in EDIF forms.

Next, these synthesized designs were verified again. The verification was done using Compass Qsim simulator, which is a gate-level, event-driven logic and timing simulator for MOS design. Qsim is intended

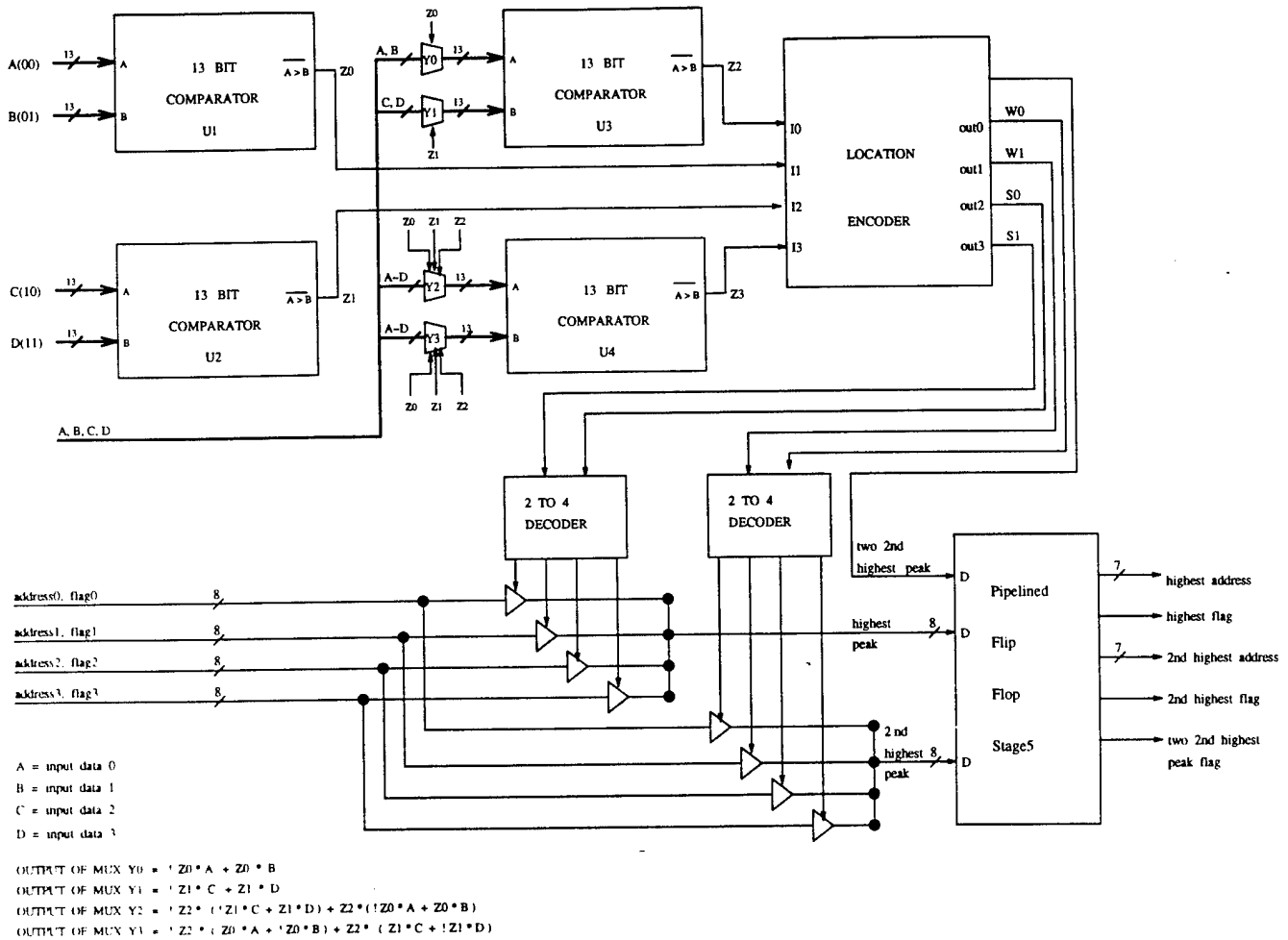
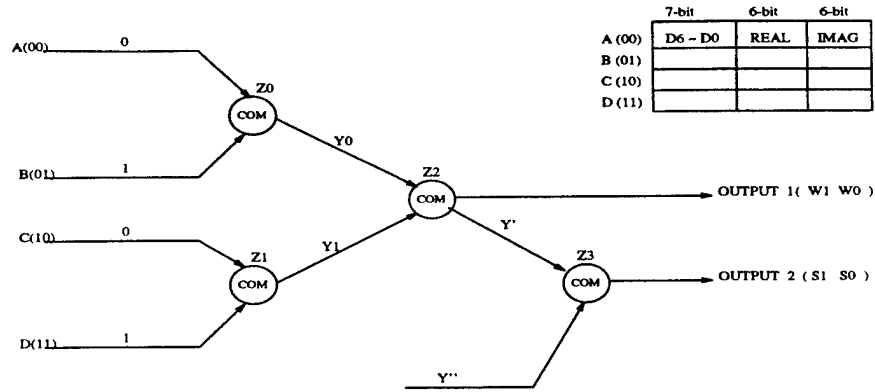


Figure 10: Block diagram for the final sorting subsystem

for use in the tool environment and was created primarily for logic and timing verification of designs using technology's portable library cells. As Qsim doesn't accept EDIF, the synthesis outputs in EDIF format were first converted to NLS format using Compass netlist utility, which is a gate-level timing description including a pin-to-pin propagation delay and a capacitance-dependent delay of each output. Qsim was also used to test various timing errors such as setup and hold times violation. Qsim simulations on the chip and its subsystems were performed on two rounds. As place and route of the cells have not been done at this stage, parasitic capacitances and resistances of the netlists are not available for inclusion in Qsim simulation. The first round simulation was done to verify the function of the chip and its subsystems after synthesis.

The subsequent stage is the automatic layout stage. The tool used is Compass Chip Compiler. The Chip Compiler is an integrated arbitrary block/standard cell placement and routing system with a floorplanning stage and an automatic floorplan evaluator. The place and route for certain portions of the design was done in bottom-up approach, while others were done in top-down approach. In the process of doing so, parasitic capacitances and resistances of the routed netlists were extracted and used to perform the post routing

LOCATION ENCODER ALGORITHM



	7-bit	6-bit	6-bit
A (00)	D6 - D0	REAL	IMAG
B (01)			
C (10)			
D (11)			

Z2	Z1	Z0	OUTPUT 1 (HIGHEST)			Y'	Y''	OUTPUT2 when Z3 = 0			OUTPUT2 when Z3 = 1		
			W1	W0				S1	S0		S1	S0	
0	0	0	A	0	0	C	B	C	1	0	B	0	1
0	0	1	B	0	1	C	A	C	1	0	A	0	0
0	1	0	A	0	0	D	B	D	1	1	B	0	1
0	1	1	B	0	1	D	A	D	1	1	A	0	0
1	0	0	C	1	0	A	D	A	0	0	D	1	1
1	0	1	C	1	0	B	D	B	0	1	D	1	1
1	1	0	D	1	1	A	C	A	0	0	C	1	0
1	1	1	D	1	1	B	C	B	0	1	C	1	0

$$W1 = Z2$$

$$S1 = Z2 \text{ xor } Z3$$

$$W0 = !Z2 * Z0 + Z2 * Z1$$

$$S0 = Z3 * !Z1 * !Z0 + Z2 * !Z1 * Z0 + !Z3 * Z1 * Z0 + !Z2 * Z1 * !Z0$$

Figure 11: Logic derivation for the location encoder

timing verification. Two different simulators, Qsim and Hspice, were used to perform this post routing 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 ASIC is designed using double-metal 0.5-micron scalable CMOS technology and packaged in a 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 a speed of 156.25 MHz.. The chip contains about 812,931 transistors and has an die size of approximately 15 mm x 15 mm. The transistor count and silicon area after cell routing and optimization of each pipelined system are calculated

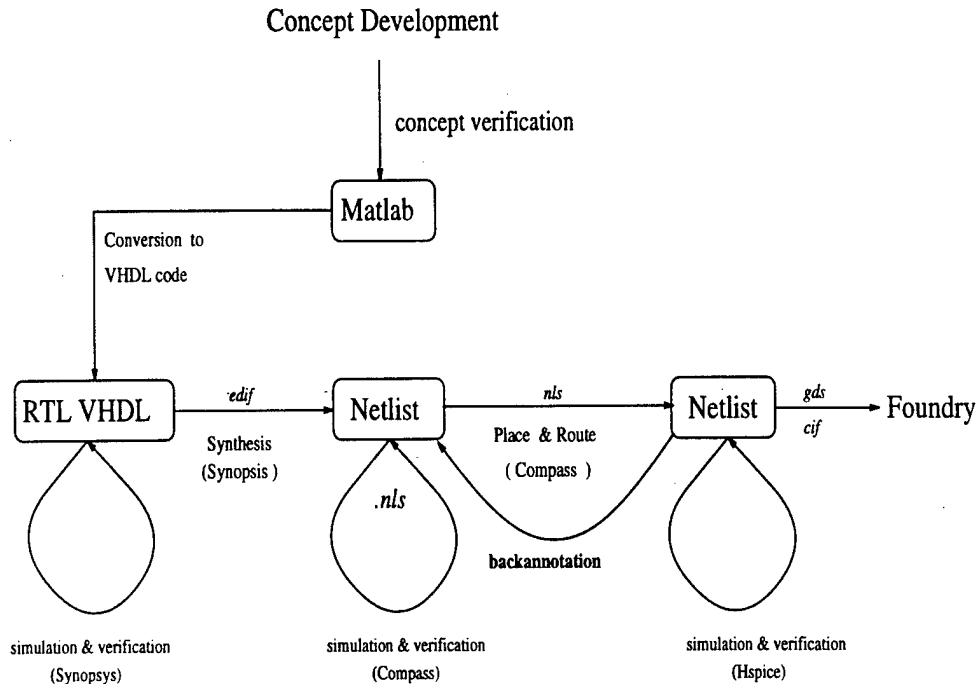


Figure 12: Design flow

and shown in Table 4. The process in each subsystem is completed within 102.4 nsec with the timing conveyed over to each subsystem as shown in Table 5. Two different simulators, Hspice and Compass Qsim, were used to perform the post 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 6.

6 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 perform the monobit FFT. The chip also includes the frequency selection logic to select the correct input frequencies and avoid picking up spurious

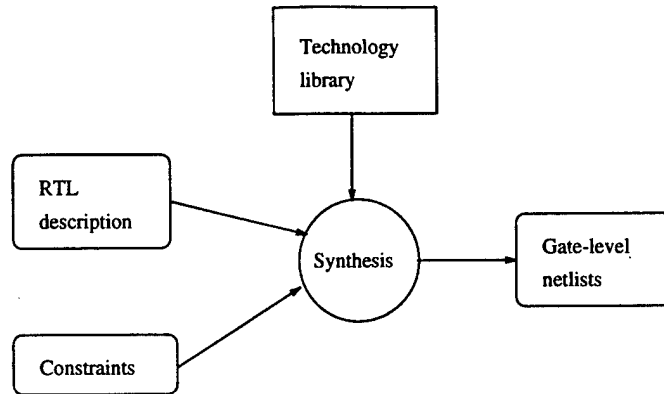


Figure 13: Synthesis process

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.

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-bit FFT and 99.87% probability of detection and 0.93% of false data for 8-bit FFT.

7 Acknowledgements

This work was supported in part by the the Air Force Office of Scientific Research under contract no. F49620-96-1-0423. Patent is in progress by Department of Defense Air Force Research Laboratories.

References

- [1] J. B. Y. Tsui and D. M. Akos, "Comparison of Direct and Downconverted Digitization in GPS Receiver Front end Designs," IEEE International Microwave Symposium Digest, vol. 3, pp. 1343-1346 June 17-21, 1996
- [2] W. T. Cochran, J. W. Cooley, D. L. Favon, 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 & Electroacoustics, vol. AU-15, pp. 45-55, June, 1967.
- [3] J. B. Y. Tsui, Digital Techniques for Wideband Receivers, Artech House, 1995.

Table 4: Transistor count and 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

Table 5: Timing analysis of each subsystem

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

Note: The timing analysis includes the delay of each pipelined flip-flops.

Table 6: Design and Performance Statistics

Technology	0.5 um 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
