Al-khwarizmi Engineering Journal

Al-Khwarizmi Engineering Journal, Vol.2,No.1,pp 35-51, (2006)

Microcontroller – Based Spectrum Analyzer

Ameera A. Salman
Control & Computers Department
University of Technology

Dr. Ismail A. Mohammed
Electrical Engineering Department
College of Engineering
University of Baghdad

Omar H. Hamad
Electrical Engineering Department
College of Engineering
University of Baghdad

(Received 14 November 2005; accepted 4 April 2006)

Abstract:-

This work includes design, implementation and testing of a microcontroller – based spectrum analyzer system. Both hardware and software structures are built to verify the main functions that are required by such system. Their design utilizes the permissible and available tools to achieve the main functions of the system in such a way to be modularly permitting any adaptation for a specific changing in the application environment. The analysis technique, mainly, depends on the Fourier analysis based methods of spectral analysis with the necessary required preconditioning processes. The software required for waveform analysis has been prepared. The spectrum of the waveform has been displayed, and the instrument accuracy has been checked. The basic hardware parts of the analyzer are the processor and the associated logic, storage media, communication to a central master computer and the data acquisition parts. However, the input / output structure is modular and may change according to the application requirements. The basic operating software modules, which are independent of an application, may be used as a part of a software package that can carryout other functions. A complete software development tools package to develop application programs, using IBM-PC and the analyzer instrument, are installed and realized at levels, the PC level and the analyzer instrument level.

Keywords: FFT, Spectral Analysis, Microcontroller - Based System

1. Introduction

The digital spectrum analyzers are represented by the Fourier analyzers. Fourier analyzers use digital signal processing techniques to provide measurements that go beyond the capabilities of spectrum analyzers. Some of their capabilities are measurement of very low-frequency or very closely spaced signals, measurement of random signals obscured by noise, and measurement of shared properties or relationships of two or more signals [1].

Fourier analyzer calculations are based on the discrete Fourier transform using an algorithm called the fast Fourier transform (FFT). This algorithm calculates the amplitude and phase of each signal component from a set of time domain samples of the input signal. Since Fourier analyzers, primarily, are digital instruments so interfacing to a computer or other digital system is relatively straightforward. Remote programming and transferring data via an interfacing system can considerably expand the range of possible applications for instrument; a Fourier analyzer will provide a high degree of accuracy, stability, and repeatability in spectrum analysis [1, 2].

2. Work Objective and Contribution

The objective of this work is to design and implement a microcontroller-based spectrum analyzer system that can accept a time – domain signal, performs processing tasks according to the designed algorithm requirements and delivers the spectrum to be graphically presented. The desired analyzer is to be an economical laboratory instrument. According to this and by utilizing the digitalism nature of the analyzer, it may be built around a PC to combine the good graphical capabilities of the PC and the availability of cost effective processors, such as the microcontroller, that have numerical processing power that can accomplish the required work tasks.

The reasons behind choosing the system to be a microcontroller – based system can be put as follows:

(a) The relatively simple processing required by DSP tasks can often be done on modest microprocessors. Microcontrollers combine microprocessor units with internal program memory and other peripherals, such as:

- data memory
- digital I/O ports
- timers and counters
- high speed serial I/O

(b) In many cases, all the hardware which is needed for DSP is on a single chip! Obviously, if there is already a microcontroller in a product, the additional cost to add DSP could be minimal [3].

The catch is that most microcontrollers are foolishly designed for control applications, not Signal Processing. Some microcontrollers, such as M68HC16, do have special enhancements for DSP, but most microcontrollers have been optimized for digital control operations such as I/O and bit manipulation, not for number crunching. This makes DSP on microcontrollers harder; for the same performance, there will be a need to work a bit more than with DSP –specific chips. When we can use them, however, microcontroller solutions may be a fraction of the cost of a fancier DSP – specific solution. In future, the expectation to see both enhanced general – purpose mathematical operation and DSP – specific functionality in microcontrollers is possible [3].

3. System Design and Implementation

3.1. Hardware Structures

In the device (instrument) design and implementation, a top – down approach is adopted. In this approach, the main task is divided into a number of smaller sub - tasks, and every one is then managed separately. The main task in essence is how to design and implement a microcontroller – based system that accepts a continuous time – domain signal and performs frequency – domain representation and analysis for this signal. All functions of the device (analyzer) may be performed and governed by certain software routines.

After the major task is thus divided, each sub – task is converted into a number of blocks or modules. When the design and implementation of every single module is accomplished, all modules are assembled and integrated together to form the target system. This system consists of the following modules:

- Data Acquisition Module,
- CPU Module, and
- Data Output and Display Module.

The main function of the data acquisition module is to acquire a time record from the underlying time - domain signal. The overall function of this module is supervised and controlled by the microcontroller in the CPU module through a set of status and control lines. However, when an order has been released from the microcontroller to the data acquisition module to acquire a time record, the time – domain signal at the module input terminal will be sampled and digitized and a record with a specific number of digital samples will be collected and sent to the CPU module. The data acquisition module circuit can be separated into the following units:

- Signal conditioning circuit,
- Anti – aliasing filtering circuit,
- Signal sampling circuit, and Conversion into digital format circuit.

The function of the signal conditioning unit is to make the analogue signal at the input port of the system conditioned in such a way to be effectively processed by the following system units and modules. Therefore this unit is important; it is usually inserted at the beginning of the system front end. Anti – aliasing filtering unit will attenuate frequency components above half the sampling frequency in the input signal. This ensures that any frequency components which could cause aliasing are suppressed. The main function of the sampling unit is to convert the continuous – time – domain signal into a sequence of continuous samples at specific time intervals (depending on the operating sampling rate). It is necessary to add another important function to the sampling circuit unit. This function is the sample holding or capturing function. .

When a microprocessor or computer is used as part of the measurement, the analogue signal has to be converted into a digital form before it can be applied to the microprocessor or computer. Analogue – to – digital converters accomplish this process. The input to an analogue – to – digital converter is an analogue signal and the output is a binary word that represents the level of the input signal.

Any interface with the microcontroller is through the buses for commands and to move the data between the Data Acquisition module or the Data output module and the CPU module. For this work the main tasks carried out by the CPU module are: micro-processing and storage media, both program and data memory and communication with the PC, by executing and responding to PC commands. The microcontroller-based system was designed and implemented using the general purpose 89C51 Intel microcontroller.

All communication with the microcontroller or CPU module, in general, is carried out by three sets of parallel wires or lines called buses. Any interface must use these buses for commands and to move the data between the Data Acquisition module or the Data output module and the CPU module. The storage media includes EEPROM for program or data storing and SRAM for data storing in addition to the flash EPROM of the microcontroller that's used for programming and reprogramming in-system issues.

One of the main tasks of the data output and displaying module is to establish an interaction between the CPU and the PC by executing and responding to commands through the serial communication circuitry of the microcontroller and the necessary external ICs of the communication circuitry. In fact, in this work the required interaction between the CPU module and the digital output and display system module (PC) has established. So, the designed system can be regarded as a micro-computing environment built around the PC which has excellent capabilities for displaying and showing the processed output data from the analyzer hardware, in addition to the interacting with external devices and systems.

The hardware structure and its modules can be described by the following block diagram:

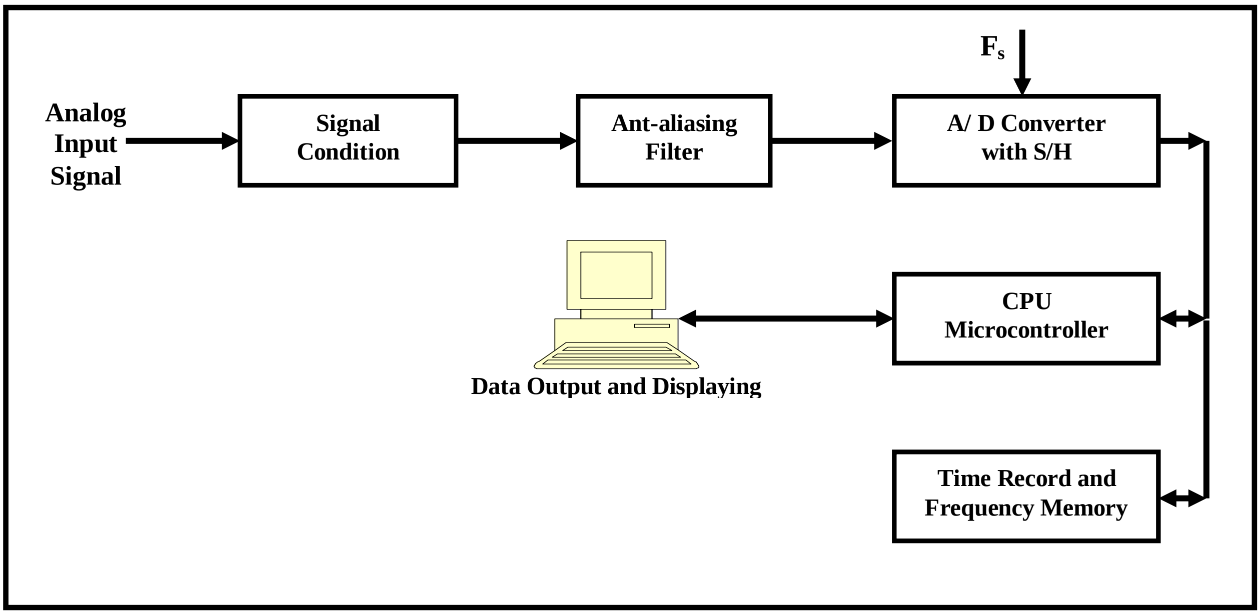


Fig.1 the functional block diagram of the system hardware structure

The hardware design approach in this work is to implement these modules at a chip level rather than using ready made modules. This approach provides board size control, specific selection of function and lower cost.

The schematic diagram of the system hardware structure is shown in **fig.2.** We refer the reader to reference [4] for detailed information about the hardware design and operation approaches.

3.2. Software Structure

In order to achieve the required functions of the analyzer, the software is assembled with strictly defined functional modules. Each functional module can be regarded as a capsule consisting of one or more programs and the associated data structure. The designed software almost incorporates computing the spectrum values and the communication with a PC to establish a user interface environment and to display the spectrum values. However, the designed and implemented software consists of a set of microcontroller (the processor) assembly language programs for computing the amplitude spectrum values and transferring these values and the associated time – domain values to the PC, and a visual basic program resident on the PC for displaying both the time – domain signal (time record) and its amplitude spectrum.

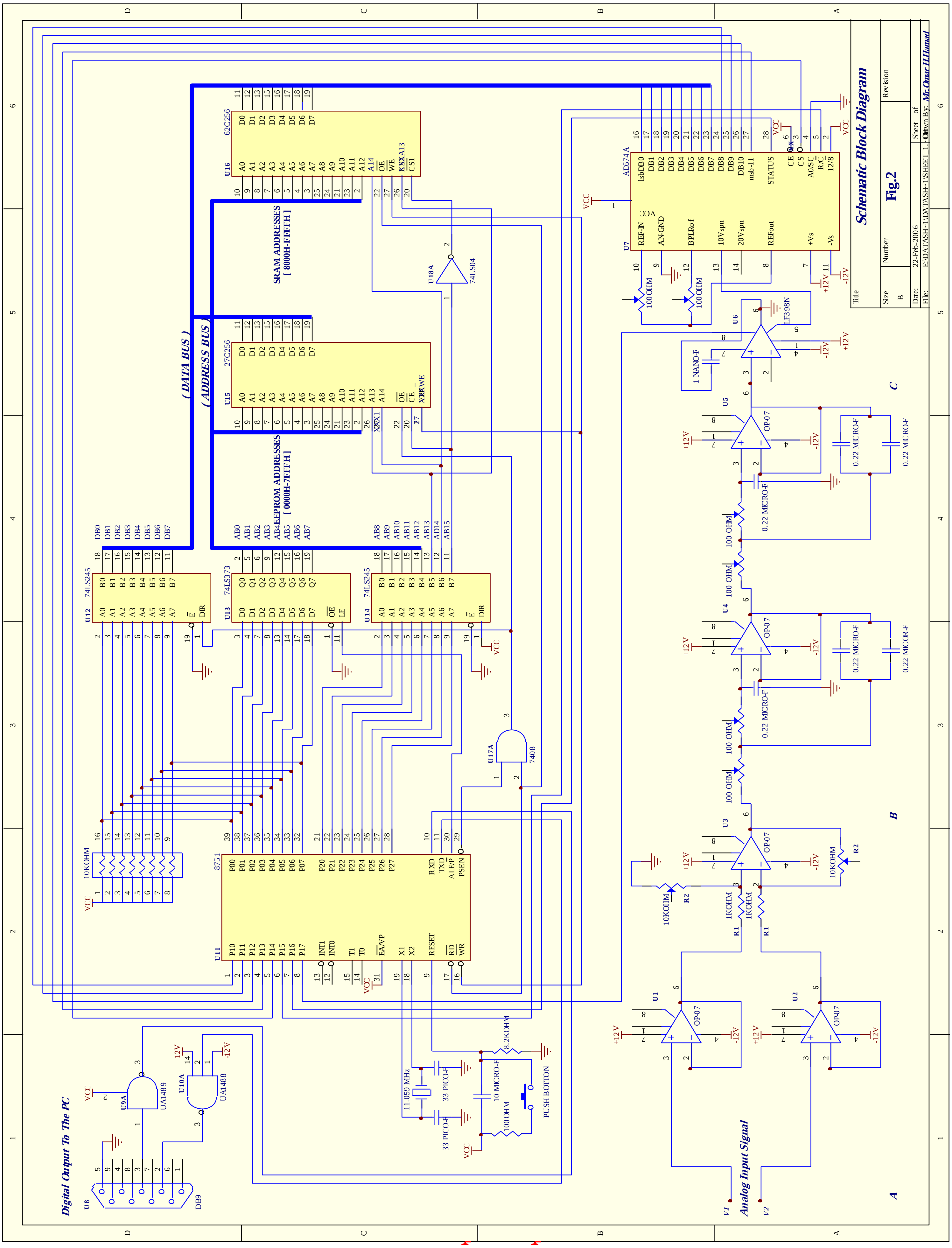
3.2.1. Microcontroller Programs

The microcontroller code, written for the SXA51 assembler, almost consists of a main program, three subroutines, and the communication service routine, these programs (routines) are:

- main routine,
- data collection routine,
- communication and data transfer routine,
- memory management and arrays initialization routine, and
- data processing (spectrum estimation) routine.

The main routine represents the heart of the analyzer software. It is used to initialize the system and direct the operations of the overall system functions. A polled structure is adopted in this work. It includes only the basic structure of a loop and subroutines which carry out the following operations:

- System initializations.
- Acquiring a time record which represents a full set of samples (data points from the underlying time – domain analogue input signal) from the data acquisition mod
- ule circuit.


Schematic Block Diagram
Fig.2
Digital Output To The PC
Analog Input Signal
SRAM ADDRESSES [8000H-FFFFH]
EEPROM ADDRESSES [0000H-7FFFH]
(DATA BUS)
(ADDRESS BUS)
62C256
27C256
74LS245
74LS373
8751
AD574A
LF398N
OP-07
UA1489
UA1488
DB9
7408
74LS04
11.059 MHz
PUSH BOTTON

- Communicating with the PC and sending the acquired time record (the PC will also process this record according to the same processing algorithm of the microcontroller). It is to be sure that the sending process is valid.
- Managing and initializing the external data RAM to be ready to hold real, imaginary and magnitude of output data from the processing algorithm operations and to transfer the collected data samples from the data acquisition collection area in the external data RAM into the real data array area with the necessary word length adjustment.
- Now, the microcontroller will process the data in the real data array according to the processing algorithm and gives the output data as real, imaginary and magnitude in the real, imaginary and magnitude data arrays, in the external data RAM, respectively.
- Communicating with the PC and sending the data of the magnitude array (the PC will save the data and it can display the time record data samples and the associated magnitude spectrums, one computed by the microcontroller (which is still sent) and the other by computer). Also be sure the sending operation is valid.
- Return to the main program and repeat.

The following flowchart shows the mechanism of the main routine.

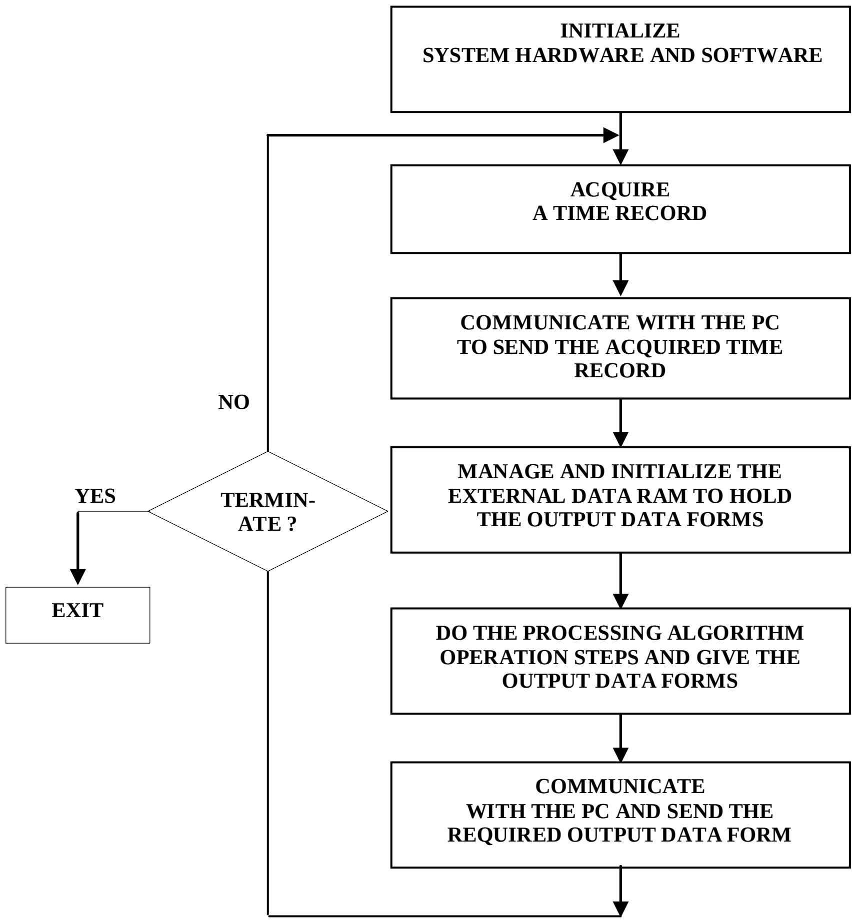


Fig.3 A polled structure of the main program.

Some of these modules or subroutines of the main program can be described as follows:

3.2.1.1. Initialization

The initialization action in the main program can be explained as follows: usually, the initialization supplements the system initialization. Firstly, when the CPU is first energized (also known as resetting or booting), it will automatically generate a reset code memory address that's designed in the CPU hardware and fetch the code bits at that address. From here, as a programmer, the responsibility is to concern on getting code and data properly organized in the memory [5]. Well, the major role of the initialization portion of the main program is to prepare the system hardware to be serially communicated with the PC.

3.2.1.2. Data Collection

After the initialization action of the main program has finished, the processor branches to the data collection subroutine action to acquire a time record from the data acquisition circuit. During the action of this subroutine the processor will manage the acquisition process by well using the control and status lines of both the analogue – to – digital conversion circuit (ADC) and the sampling and holding circuit (S/H) in the data acquisition module.

The acquired samples will be collected in the time record buffer place in the static RAM. The acquired samples from the ADC are of 12-bit resolution, but they are stored as 16-bit (2-bytes) format by augmenting zeros. The implemented software of the data acquisition subroutine has been designed to utilize the permissible sampling frequency from the designed data acquisition module circuit which is around *23 Ksample/sec* [4].

3.2.1.3. Communication and Data transfer

After, the time record has been collected, the processor will branch to the communication and data transferring routine according to the actions sequence of the adopted main program algorithm. Through the initialization action the communication circuit of the microcontroller (UART, Universal Asynchronous Receive / Transmit) has been prepared and still being ready to receive and transmit data.

Firstly, the data to be transmitted must be determined by its boundaries (boundaries means the address of the head and tail locations in this area) then these boundaries will be loaded into general-purpose registers used as pointers. The next step is to calculate the checksum for the data in this area. After that the transmission operation will begin. A crystal of 11.0592 MHz was used to give a machine cycle time about 1 µsec and the UART circuit is programmed to give standard baudrate of 1200 bit/sec [4, 5].

With each data transmitting process a checksum is computed and sent. After the data and its checksum received by the machine (the PC or the microcontroller) a new checksum is computed for the received data by the same computation manner followed for the sent one. If the two checksums are the same then the ok command code is sent (which means that the data received without error), else the retransmit command code is sent. The mechanism of data byte transmitting process from or to the microcontroller or PC sides and the details of this operation is detailed in literature [4].

3.2.1.4. Memory management and Arrays initialization

The major role of this subroutine is to initialize and manage the data memory (SRAM) addresses that will hold the temporary stored data bytes during the data processing (spectrum estimation) subroutine and the output data from that routine. However, the data memory (SRAM) which is of 32kbytes (begins from 8000h – to – FFFFh as it was designed and shown in the previous section), accordingly, it can be divided into four sequential familiar arrays. The data collection array, the real data array, the imaginary data array and the magnitude array. Normally, the boundaries of each array are determined by related labels that have been declared in the beginning of the main program (see the main routine subsection). However, through this subroutine the processor, as a first step, will erase the last three arrays (fill with zeroes). The data collection array holds the collected data samples after the processor executes the data collection subroutine and each sample is of two bytes format (word). The processor, as a second step, will transfer the 2-bytes (word) samples in the data collection array to the real data array to be stored as 4-bytes samples (zero extended) to satisfy the resolution and to avoid the overflow situations during data processing subroutine. The processor will deal with data in the last three arrays during the execution of the next subroutine (spectrum estimation routine).

Finally, the reason behind such data memory organization is supplemented by the adopted software algorithm especially by the data processing (spectrum estimation) routine in the main program routine as will be explained later.

3.2.1.5. Data processing (Spectrum estimation

The FFT (Fast Fourier Transform) is the heart of the spectrum estimation. It is *a radix-2, decimation in time (DIT) algorithm* based on the *Cooley-Tukey algorithm*. This *algorithm* is the simplest and the most commonly used. It is well explained in literatures [7, 8]. The concern will be on the processor actions during this subroutine. Firstly, the processor will branch to condition the collected time record samples (which are now resident in the real array place) by multiplying each data sample of the time record in the real array by an appropriate window coefficient (Hamming window coefficient). In fact, the processor will take the samples from the real array sequentially and for each data sample it will compute the related window coefficient (which has the same index of sample in the real array), multiply them to get a weighted data sample which will be restored in the same location of the real data array (in place computation). The second step is to scramble the sequence of the data samples in the real data array according to the demands of the *Cooley- Tukey DIT-FFT algorithm*. Now, after the data samples in the real array have been windowed and scrambled, it is ready to be FFTed. The processor will branch to do the computations supplemented by the adopted FFT algorithm and the computation will also be in placed. When the processor finishes these operations the output data samples (spectrum samples) will be in real – imaginary form, the real part of each output data sample is in the real array and the corresponding imaginary part of the same sample is in the imaginary array, it will begin to compute the magnitude spectrum samples as follows: the processor will take two output data samples, the first one is from the real array and the other is the corresponding one in the imaginary array (of the same index in the array) and then it will calculate the magnitude of a spectrum sample (square the real, square the imaginary, sum the two and find the square root of the sum) that will be saved in the appropriate location in the magnitude array (the current real, imaginary and magnitude data samples must have the same array index). Whenever, the processor accomplishes these operations it will return to the main routine. Finally, the main routine will direct the processor to the communication subroutine to transfer the output data samples of the magnitude array (the amplitude spectrum) to the PC by the same mechanism that was explained previously in the communication and data transferring subsection.

3.2.2. PC Programs

The PC display software represents a visual basic program resident on the PC. This program will integrate the analyzer work by creating a user interface environment between the analyzer hardware and the user of the analyzer. When the program is first executed the platform of **fig.4** will be displayed.

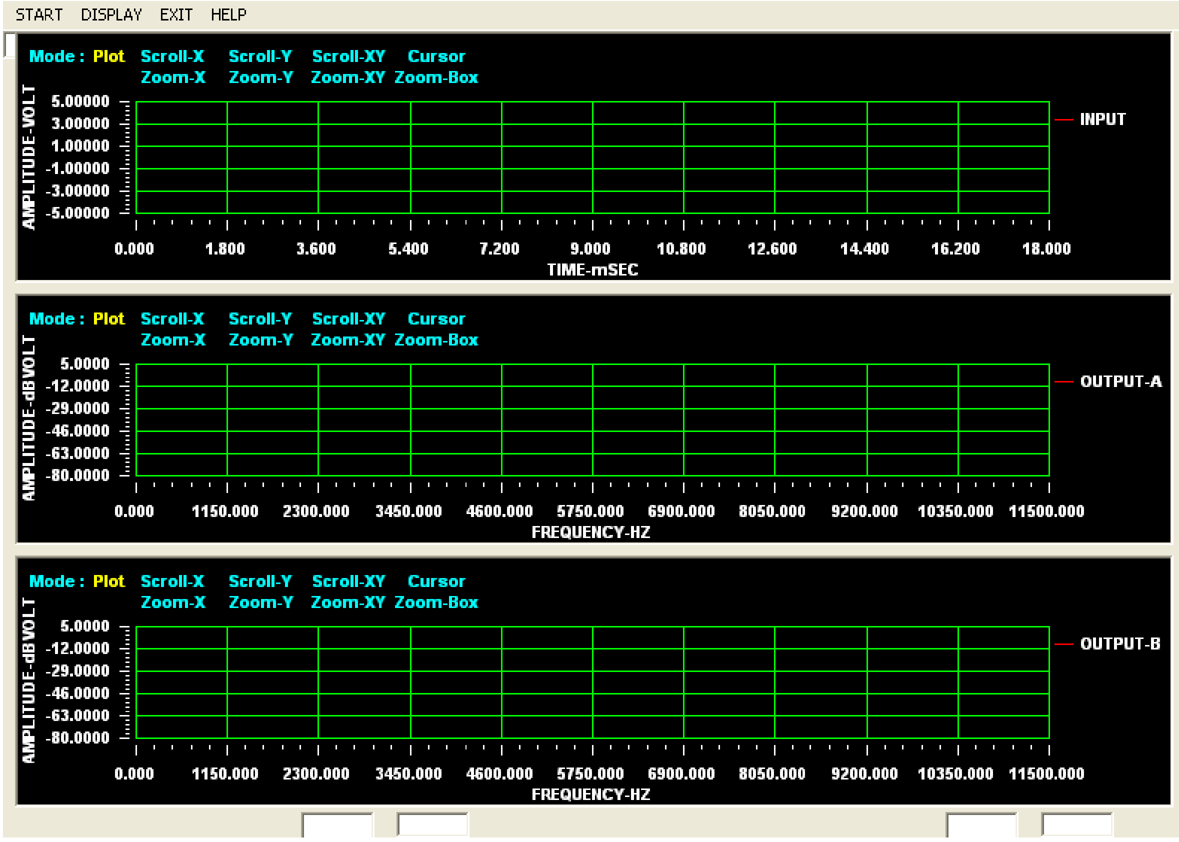


Fig.4 Displaying and user interfacing platform.

A menu is displayed requesting the user to select an option:
(1) START option, when selected a button named as RUN will appear.
(2) DISPLAY option, when selected a sequence of buttons will appear such as: CURRENT >>, NEXT >, OLDEST << and PREVIOUS <.
(3) EXIT option, when selected a button named as END will appear.
(4) HELP option, when selected a button named as ABOUT will appear.

Also, there are three displaying windows in the platform labeled as input, simulated, and actual. The first window is to display the acquired time record (form the analogue input signal) by the analyzer hardware, the second window is to display the spectrum of this record computed by the visual basic program and the third window is to display the spectrum computed by the analyzer hardware. The operation mechanism of the visual basic program is simple and can be explained as follows: To start the analyzer cycle the RUN button in the start option must be clicked. When the user does so, the visual basic program will send the go command code to the analyzer main program and then it takes the control (in fact, the analyzer main program is asking for that as was explained in the main program section). The main program will direct and manage its subroutines to get the spectrum of the underlying time – domain input signal as had been described above. After that, the visual basic program will manage the operation of two subroutines the communication subroutine, processing subroutine and the displaying subroutine. The communication subroutine will operate automatically in a manner proportion with the actions sequence of the analyzer main program. After the analyzer main program took the control the visual basic communication subroutine expects it will receive the time record data samples, so it will prepare a file in the hard disk to save the received data and when data samples begin transferring it will collect them in the file. After the data samples collected the visual basic program will begin to compute the checksum of the collected data samples in the same manner that the communication subroutine in the main program has computed it. Now, it will compare the computed checksum with that sent with the data samples. If they are equal, the visual basic program will pass a go command code to the communication subroutine in the main program, otherwise, it will pass a retransmit command code. However, whenever the visual basic program communication subroutine ends this situation it will be ready to receive the hardware output data samples (amplitude spectrum samples) and the same scenario will be repeated and so on. The PC resident visual basic main program can be simulated in the following:

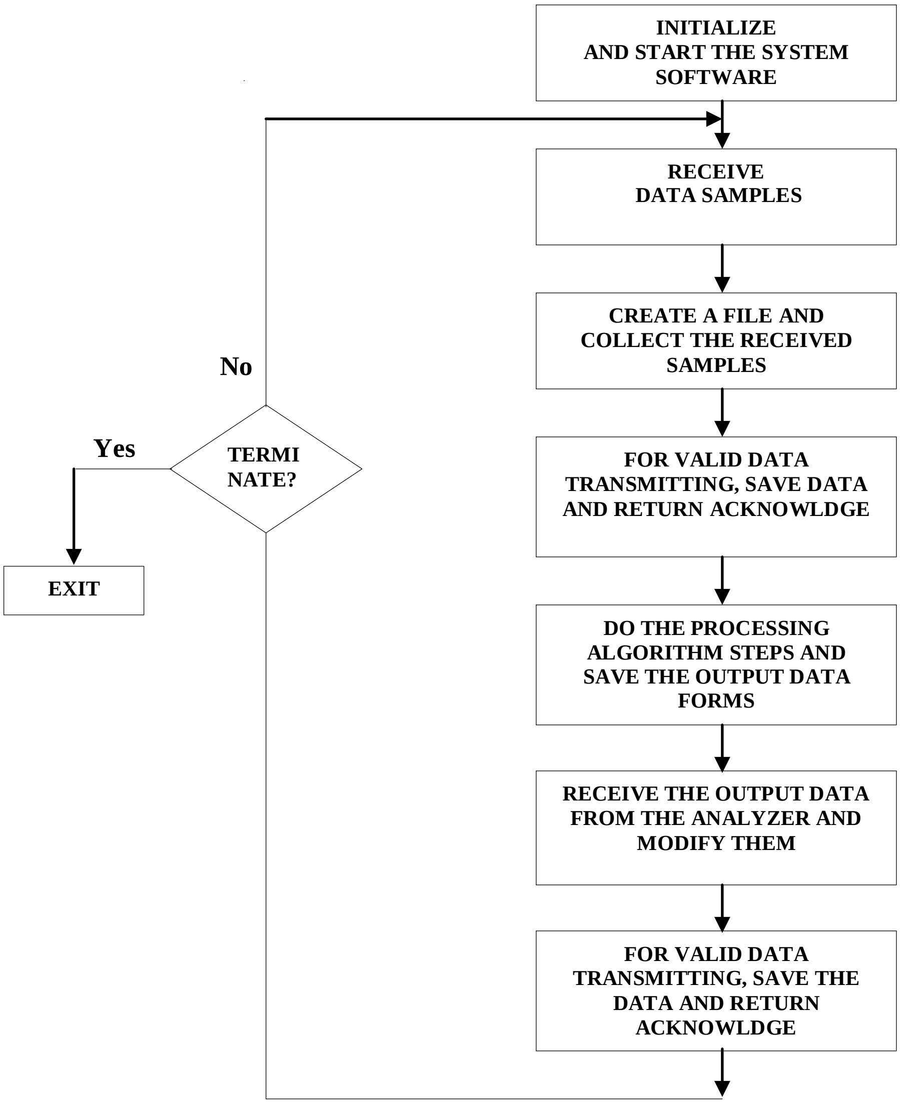


Fig.5 The PC resident vb-program flow chart.

During the waiting period that the visual basic program will live in to receive the data output from the hardware it will direct the operation flow to the visual basic data processing subroutine (spectrum estimation routine) which has been built according to the analyzer processing spectrum estimation algorithm (data processing algorithm). Needless, to say that the Pentium computer is very fast as compared with the analyzer speed. However, visual basic processing subroutine will end its tasks and save the output data (spectrum values) in a file in the hard disk, and then the visual basic program will return to the communication subroutine and to receive the output from the analyzer hardware. It is necessary to say that the file creation process for saving data was done by a professional means of data base programming so as to keep the actions flow as efficient and quick as possible.

The displaying subroutine works to perform the user requests. If any button of the display option was selected before the run button or the system still in its first cycle, a small window will appear with acknowledgement of there is no record to be displayed. Any way, if the system operates and the input and output data transferring to the PC is continuous and display option was selected, then you can see some things on the displaying windows. If the CURRENT>> button is clicked the most recent time record and its spectrums (two same spectrums, one from the visual basic program and the other from the analyzer hardware) will be displayed, if the OLDEST<< button is clicked the oldest time record and its spectrums will be displayed and finally, the PREVIOUS< button and NEXT> button will work to display the previous or the next time record and its spectrums with respect to the displaying one at this moment on the screen. Each displaying window has options: scroll x, scroll y, scroll xy, zoom x, zoom y, zoom xy, zoom box and cursor option to read and display the x-axis and the y-axis of a selected point.

It is necessary to clarify our contribution about displaying two identical spectrums for any acquired time record as follows:

Firstly, it's recommended to convert the processing algorithm into a high level language before writing it in assembly language (for SXA51, the particular assembler) to verify the same performance. Therefore, the two identical spectrums means that the two machines (the PC and the Microcontroller) are programmed to give processing results with same accuracy. However, after the tracking process to reach to the assembly program with the aimed processing accuracy was succeeded, so the computed spectrum by the PC can be removed. In fact, it is so easy to remove it, by just ignore the calling instruction for the particular subroutine, but at the same time we can say that the computing of this spectrum doesn't slow the system because after a time record is uploaded to the PC two commands will be released at the same time for the microcontroller and for the PC to compute the spectrum of the acquired time record. Needless to say that the Pentium IV PC is faster than the microcontroller, therefore the remaining of the computed spectrum by the PC doesn't matter as long as the PC is a part of the system.

The PC is used just for display, so as a future work we can easily improve the system, to be a complete portable kit, by adding a display unit with good resolution, LCD for example, just likes that's used in the portable oscilloscope.

4. Analyzer Features and Operation Test

The analyzer performance testing process includes supplying a known spectrum signal like sinusoidal, square and triangle waveforms with different frequencies from a function generator to the analyzer target hardware. Then the obtained spectrums can be displayed and analyzed.

It is necessary to remember the main features of the accomplished spectrum analyzer prototype. The spectrum analyzer consists of a microcontroller – based board connected to an IBM – PC or compatible, and a PC – resident software package for displaying the signal spectrum and integrating the target board operation. The main operation features of the analyzer are as follows:

- Single analogue input channel, with a 12 – bit ADC;
- Sampling rate is less than 23 kHz;
- Estimates spectra of signals in the frequency range 0 – 7kHz using the radix – 2 FFT algorithm;
- Serial communication with a host personal computer via an RS232 port;
- Displaying the signal and its spectrum on the PC using the host's graphics facilities.
- The spectrum display resolution of the FFT analyzer Δf is less than 11 Hz;

One thing must also be remembered is that the analyzer can be used to analyze the stationary, deterministic and periodic signals. For such class of signals the analyzer can deliver a consistent estimated spectrum.

One test will be demonstrated and discussed in this section; it represents the practical results obtained from the analyzer target board and PC resident software package and displayed by the PC on the software designed window. Appendix-E of reference [4] will demonstrate some of the practical results obtained from other tests.

The test was carried out by applying *a sine wave* to the analyzer and recording the spectrum. The amplitude of the applied signal was *5 volt* and its frequency was *1 kHz*. **Fig.6** shows the display window on the PC screen with the results of this test. It consists of three separate display parts: the first part, which is entitled as *INPUT*, represents the current acquired time record from the underlying input signal that to be processed. The second part, which is entitled as *OUTPUT-A*, represents the spectrum of the particular acquired time record computed by the PC as was cleared in the software structure chapter. The third part, which is entitled as *OUTPUT-B*, represents the spectrum of the same acquired time record computed by the target hardware board.

Fig.6 The time record and its spectrum of 1 kHz sine wave.

Each display part is supported with scroll and zoom options to make the results reading analysis process as easy and accurate as possible.

Conceptually, the expected spectrum of a pure sinusoid consists of two impulses located at -/+ *f Hz*, where *f* is the time – domain signal frequency, the *–ve frequency* is redundant, so it will be discarded as was pointed out in theory and software structure chapters *(single sided obtained spectrum)*.

However, as was pointed out from the theory chapter that the spectrum of a time – domain sequence *x[n]* is obtained by sampling its *Fourier transform* $X(\exp(j2\pi f))$ at $f = (\Delta f.k)$ *Hz*, where Δf represents the *spectrum resolution* or the width of the *spectrum frequency bins* and *k* is the *index of each frequency bin* in the spectrum begins from *zero till N-1(where N is the FFT size or the number of the spectrum samples).* Also that the *first half* of the *spectrum (or FFT) samples* for $k = 0$ *to* $k = (N/2) - 1$ correspond to the positive frequency axis from $f = 0$ *to* $f = fs/2$ excluding the point *fs/2*, while the second half for $k = (N/2)$ *to N-1* corresponds to the negative frequency axis from $f = - fs/2$ *to* $f = 0$ excluding the point $f = 0$ *(which is discarded).*

According to the instrument specifications the time record length *(N/fs)* is fixed at *89.04347 ms or (N = 2048 and fs = 23 ksample/sec).* So, Δf *is 11.23047 Hz* and *the spectrum frequency bins locate at (11.23047).k Hz, k = 0 to 1023 (for single sided spectrum).*

It is clear from the first part of **fig.6** that the acquired time record is a pure sinusoid. It is also clear that the delivered spectrums from the PC *(OUTPUT-A)* and from the target hardware board *(OUTPUT-B)* are the same.

This means that the high level language spectrum computation module of the software package, that resident on the PC, and the low level language (machine language) spectrum computation module of the software package, that resident on the target hardware are of the same operation and performance on the two machines. **Fig.6** also shows that the obtained spectrums consist of single strong harmonic as was expected, spurious frequencies and the noise floor. The amplitudes of the spurious frequencies and the noise floor are 40 dB below the strong harmonic amplitude. This gives an indication about the performance of the data acquisition module of the system. By the way the dB volt amplitude measurement is obtained by comparing all the amplitudes of the spectrum with the maximum amplitude in the same spectrum.

Fig.7 shows the accurate reading of the amplitude and the frequency of the harmonic of interest by using the reading options on each displaying part.
However, the harmonic is *999.512 Hz frequency and 0 dB amplitude*.

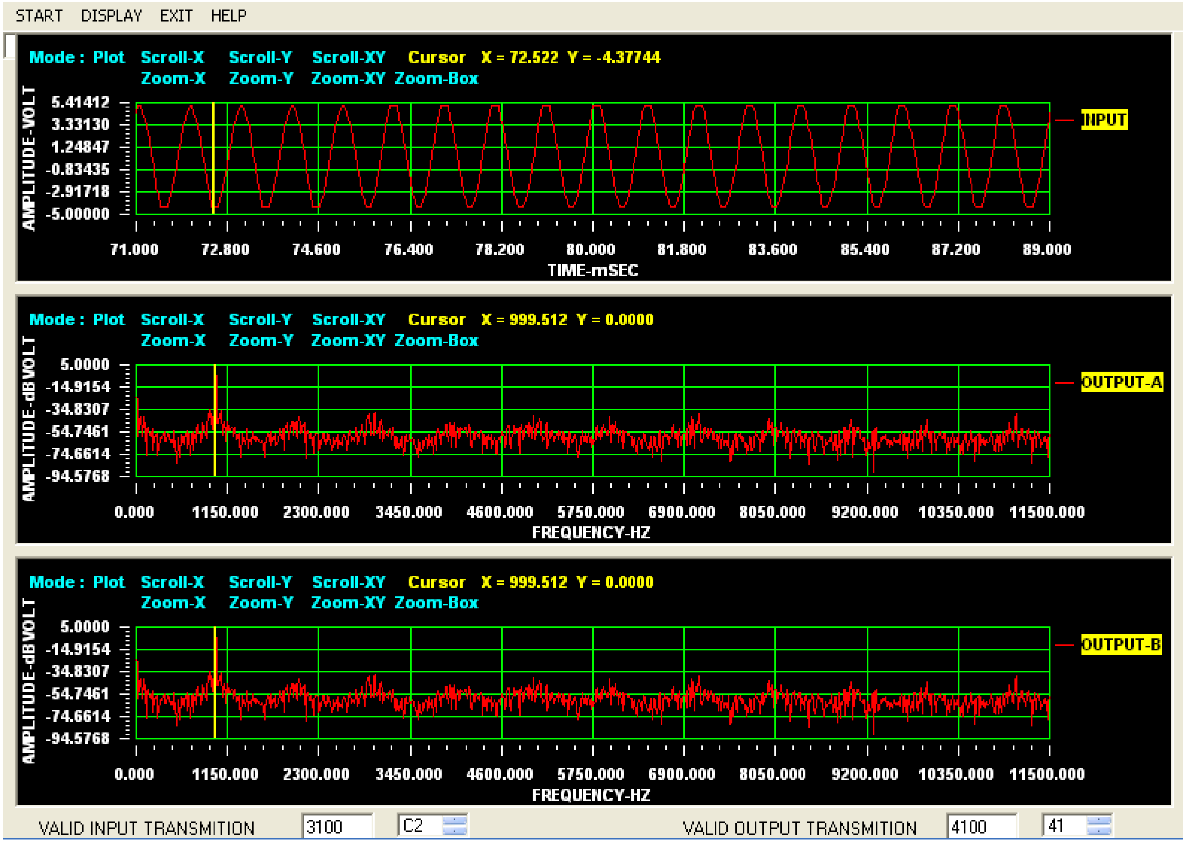


Fig.7 The amplitude and frequency of the spectrum strong harmonic.

Conceptually, it is known that the fundamental frequency of the input signal is *1kHz*, but the reading shows that the harmonic frequency is *999.512 Hz* which represents *the spectrum frequency bin of index 89*. Here, it can be said that the signal harmonic matches *the nearest frequency bin in the spectrum*. In the appendix-I of this paper, we demonstrated the interested samples (which represent the region that contain the alone fundamental harmonic) of the time and frequency records but not all the records samples (1024 samples) because of the size limitation.

That is to say that there is no *frequency bin* in the spectrum can represent the signal harmonic frequency exactly. In other words, a *leakage* has occurred in the spectrum. The interpretation, cause and effects of this *phenomenon* were well explained in the theory chapter.

However, *sine waves* with different frequencies *(almost the overall analyzer band was covered)* were supplied to the analyzer, individually, and the obtained *results readings* were analyzed and studied by the same manner that the *1 kHz sine wave spectrum* was analyzed by.

5. CONCLUSIONS

A microcontroller – based spectrum analyzer is designed and implemented for laboratory instrumentation applications. The analyzer design includes the basic hardware and software requirements for most traditional FFT analyzers. The analyzer hardware was described as a structure consists of modules;

these modules were designed to be flexible and modular. The analyzer software also was described as a structure which consists of routines; some of these routines are system-dependent, just like the data acquisition and communication routines, which were defined and implemented to be the backbone of a complete application program. In this way, the required effort and time to build the complete application software was greatly reduced and could be incrementally upgraded.

The work, also, includes the necessary built – in facilities in the hardware and in the software of the FFT analyzer to allow down – loading / up – loading, testing and modifying of application programs. This is carried out using the interchangeable part of program and data of the analyzer, and the communication ability with the PC – computer. The PC – computer incorporates software development tools and the software library as well as the software for communication and processing with the analyzer.

As an FFT – based measurement it is concluded that there are many issues to be considered when signals pure analyzed and measured from the plug-in devices. Unfortunately, it is easy to make incorrect spectral measurements. This can be overcome by understanding the basic computations involved in FFT – based measurement, knowing how to prevent aliasing, properly scaling and converting to different units, choosing and using windows correctly, and learning how to use FFT – based functions for network measurement are all critical to the success of analysis and measurement tasks.

Finally, the microcontroller plays a good role in providing the processing ability for spectrum computation of stationary signals that we deal with in the power and power electronic fields and optimizing the cost and size of the system as compared with the DSP processors.

6. REFERENCES

[1] Larry D. Jones, and A. Foster Chin, **"Electronic Instruments and Measurements"**, John Wiley & Sons, 1983.

[2] Bayard – Alpert, SRS Application Notes, 132 Application Notes, Application Note #1, The SR760 FFT Spectrum Analyzer, **"About FFT Spectrum Analyzer"**, Sr-systems 2001.

[3] Dale Grover, and John R. Deller, **"Digital Signal Processing and the Microcontroller"**, Prentice-Hall, Inc., 1999.

[4] Omar H. Hamad Al-Jubouri, **"Microcontroller – Based Spectrum Analyzer"**, University of Baghdad, College of Engineering, Electrical Engineering Department, April 2004.

[5] Kenneth J. Ayala, **"The 8051 Microcontroller, Architecture, Programming and Application"**, West Publishing Co., 1997.

[6] Emmanuel C. Ifeachor, and Barrie W. Jervis, **"Digital Signal Processing, a Practical Approach",** Addison-Wesley Publisher Co., Inc., 1998.

[7] Sanjit K. Mitra, **"Digital Signal Processing, A computer-Based Approach"**, McGraw-Hill Co., Inc., 1998

7. BIBLIOGRAPHY

[1] T. Carey, **"Introduction to FFT Analysis**, application note", IFR 2309 FFT Spectrum Analyzer, IFR Americas, Inc., Ifr-systems 2001.

[2] Amar Palacherla, **"Implementation of Fast Fourier Transform"**, Microchip Technology Inc., 1997.

[3] Dimitris G. Manolakis, Vinay K. Ingle and Stephen M. Kogon, **"Statistical and Adaptive Signal Processing, Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing"**, McGraw-Hill Companies, Inc., 2000.

[4] Alan V. Oppenheim, Ronald W. Schafter, and John R. Buck, **"Discrete Time Signal Processing"**, Prentice-Hall, Inc., 1999.

[5] Larry D. Jones and A. Foster Chin, **"Electronic Instruments and Measurements"**, John Wiley & Sons, 1983.

[6] Michael Cerna and Audrey F. Harvey, **"The Fundamentals of FFT-Based Signal Analysis and Measurement"**, National Instruments, July 2000.

8. APPENDIX-I

Sam -ple No.	Time Record Samples	Sam -ple No.	Frequency Record Samples	Sam -ple No.	Time Record Samples	Sam ple No.	Frequency Record Samples
0	4.997559E+000	0	-1.873647E+001	43	4.997559E+000	43	-7.078037E+001
1	3.747559E+000	1	-2.624883E+001	44	4.997559E+000	44	-6.365289E+001
2	3.122559E+000	2	-5.882431E+001	45	4.997559E+000	45	-8.321598E+001
3	1.872559E+000	3	-4.935031E+001	46	4.997559E+000	46	-6.008918E+001
4	6.225590E-001	4	-4.374442E+001	47	3.747559E+000	47	-6.240997E+001
5	-6.274410E-001	5	-4.231280E+001	48	3.122559E+000	48	-7.160165E+001
6	-1.877441E+000	6	-5.671040E+001	49	1.872559E+000	49	-7.708832E+001
7	-3.127441E+000	7	-5.968631E+001	50	6.225590E-001	50	-5.769765E+001
8	-3.752441E+000	8	-4.506967E+001	51	-6.274410E-001	51	-6.458065E+001
9	-4.377441E+000	9	-4.228024E+001	52	-1.877441E+000	52	-5.888700E+001
10	-4.377441E+000	10	-5.083171E+001	53	-3.127441E+000	53	-6.629690E+001
11	-4.377441E+000	11	-7.567719E+001	54	-3.752441E+000	54	-5.911370E+001
12	-3.752441E+000	12	-5.280342E+001	55	-4.377441E+000	55	-5.977924E+001
13	-3.127441E+000	13	-5.158810E+001	56	-4.377441E+000	56	-5.456587E+001
14	-1.877441E+000	14	-5.310258E+001	57	-4.377441E+000	57	-7.023880E+001
15	-6.274410E-001	15	-5.745908E+001	58	-3.752441E+000	58	-6.089640E+001
16	6.225590E-001	16	-5.957225E+001	59	-3.127441E+000	59	-5.601204E+001
17	1.872559E+000	17	-5.834197E+001	60	-1.877441E+000	60	-7.629806E+001
18	3.122559E+000	18	-5.328027E+001	61	-6.274410E-001	61	-5.625283E+001
19	4.372559E+000	19	-7.142524E+001	62	6.225590E-001	62	-5.600176E+001
20	4.997559E+000	20	-6.084682E+001	63	1.872559E+000	63	-7.103735E+001
21	4.997559E+000	21	-5.754857E+001	64	3.122559E+000	64	-5.591402E+001
22	4.997559E+000	22	-5.965593E+001	65	4.372559E+000	65	-5.819975E+001
23	4.997559E+000	23	-5.693432E+001	66	4.997559E+000	66	-6.386923E+001
24	3.747559E+000	24	-7.635503E+001	67	4.997559E+000	67	-5.452954E+001
25	3.122559E+000	25	-6.826861E+001	68	4.997559E+000	68	-4.841319E+001
26	1.872559E+000	26	-6.219299E+001	69	4.997559E+000	69	-5.668297E+001
27	6.225590E-001	27	-6.099401E+001	70	3.747559E+000	70	-5.691867E+001
28	-6.274410E-001	28	-5.949426E+001	71	3.122559E+000	71	-5.668018E+001
29	-1.877441E+000	29	-6.030814E+001	72	1.872559E+000	72	-6.439889E+001
30	-3.127441E+000	30	-5.753728E+001	73	6.225590E-001	73	-5.795745E+001
31	-3.752441E+000	31	-5.475534E+001	74	-6.274410E-001	74	-4.680313E+001
32	-4.377441E+000	32	-6.250329E+001	75	-1.877441E+000	75	-4.976460E+001
33	-4.377441E+000	33	-6.057101E+001	76	-3.127441E+000	76	-5.357193E+001
34	-4.377441E+000	34	-6.359385E+001	77	-3.752441E+000	77	-5.566862E+001
35	-3.752441E+000	35	-5.595388E+001	78	-4.377441E+000	78	-6.104382E+001
36	-3.127441E+000	36	-5.747597E+001	79	-4.377441E+000	79	-4.759890E+001
37	-1.877441E+000	37	-5.865975E+001	80	-4.377441E+000	80	-4.349609E+001
38	-6.274410E-001	38	-6.465760E+001	81	-3.752441E+000	81	-4.275117E+001
39	6.225590E-001	39	-6.236360E+001	82	-3.127441E+000	82	-3.597241E+001
40	1.872559E+000	40	-6.203112E+001	83	-1.877441E+000	83	-4.819513E+001
41	3.122559E+000	**Fundamental Harmonic**		84	-6.274410E-001	84	-3.938277E+001
42	4.372559E+000	42	-5.432337E+001	85	6.225590E-001	85	-4.390677E+001
86	1.872559E+000	86	-5.315138E+001	138	4.997559E+000	138	-6.508000E+001
87	3.122559E+000	87	-4.077068E+001	139	4.372559E+000	139	-6.385951E+001
88	4.372559E+000	88	-5.717050E+000	140	3.122559E+000	140	-6.394428E+001
89	4.997559E+000	**89**	**3.293777E-011**	141	1.872559E+000	141	-6.258723E+001
90	4.997559E+000	90	-9.210440E+000	142	1.247559E+000	142	-6.517404E+001

91	4.997559E+000	91	-4.713401E+001	143	-6.274410E-001	143	-7.726246E+001
92	4.997559E+000	92	-4.703065E+001	144	-1.877441E+000	144	-6.508245E+001
93	4.372559E+000	93	-3.928467E+001	145	-3.127441E+000	145	-6.372617E+001
94	3.122559E+000	94	-4.214313E+001	146	-3.752441E+000	146	-6.408963E+001
95	1.872559E+000	95	-4.454453E+001	147	-4.377441E+000	147	-5.654506E+001
96	6.225590E-001	96	-3.671652E+001	148	-4.377441E+000	148	-5.852366E+001
97	-6.274410E-001	97	-5.723565E+001	149	-4.377441E+000	149	-6.249993E+001
98	-1.877441E+000	98	-4.618958E+001	150	-3.752441E+000	150	-5.471474E+001
99	-3.127441E+000	99	-4.877660E+001	151	-3.127441E+000	151	-6.692928E+001
100	-3.752441E+000	100	-5.203275E+001	152	-1.877441E+000	152	-6.015575E+001
101	-4.377441E+000	101	-6.298409E+001	153	-1.252441E+000	153	-6.185376E+001
102	-4.377441E+000	102	-4.659503E+001	154	-2.441000E-003	154	-6.193993E+001
103	-4.377441E+000	103	-5.173904E+001	155	1.872559E+000	155	-5.708831E+001
104	-3.752441E+000	104	-5.389490E+001	156	3.122559E+000	156	-6.738927E+001
105	-3.127441E+000	105	-5.278426E+001	157	3.747559E+000	157	-6.560045E+001
106	-1.877441E+000	106	-5.417643E+001	158	4.997559E+000	158	-6.245710E+001
107	-6.274410E-001	107	-5.187428E+001	159	4.997559E+000	159	-6.052150E+001
108	6.225590E-001	108	-5.304983E+001	160	4.997559E+000	160	-5.703553E+001
109	1.872559E+000	109	-5.889707E+001	161	4.997559E+000	161	-5.273771E+001
110	3.122559E+000	110	-5.460213E+001	162	4.372559E+000	162	-6.005663E+001
111	4.372559E+000	111	-5.177381E+001	163	3.122559E+000	163	-5.859360E+001
112	4.997559E+000	112	-6.151042E+001	164	2.497559E+000	164	-5.307770E+001
113	4.997559E+000	113	-5.793824E+001	165	1.247559E+000	165	-7.112561E+001
114	4.997559E+000	114	-6.082418E+001	166	-2.441000E-003	166	-5.581701E+001
115	4.997559E+000	115	-5.714719E+001	167	-1.877441E+000	167	-5.030834E+001
116	4.372559E+000	116	-5.944250E+001	168	-3.127441E+000	168	-5.284195E+001
117	3.122559E+000	117	-7.777445E+001	169	-3.752441E+000	169	-4.748558E+001
118	1.872559E+000	118	-5.618007E+001	170	-4.377441E+000	170	-6.232999E+001
119	6.225590E-001	119	-6.411050E+001	171	-4.377441E+000	171	-5.968141E+001
120	-6.274410E-001	120	-6.045932E+001	172	-4.377441E+000	172	-5.306939E+001
121	-1.877441E+000	121	-6.132428E+001	173	-3.752441E+000	173	-4.558540E+001
122	-3.127441E+000	122	-6.544662E+001	174	-3.127441E+000	174	-5.046612E+001
123	-3.752441E+000	123	-6.104099E+001	175	-1.877441E+000	175	-4.658402E+001
124	-4.377441E+000	124	-5.229524E+001	176	-1.252441E+000	176	-5.735479E+001
125	-4.377441E+000	125	-7.007608E+001	177	-2.441000E-003	177	-4.846329E+001
126	-4.377441E+000	126	-6.271700E+001	178	1.872559E+000	178	-4.620173E+001
127	-3.752441E+000	127	-6.298572E+001	179	3.122559E+000	179	-5.794666E+001
128	-3.127441E+000	128	-6.065785E+001	180	3.747559E+000	180	-6.041302E+001
129	-1.877441E+000	129	-6.592685E+001	181	4.997559E+000	181	-4.974636E+001
130	-6.274410E-001	130	-5.615884E+001	182	4.997559E+000	182	-4.277241E+001
131	6.225590E-001	131	-5.526116E+001	183	4.997559E+000	183	-4.815274E+001
132	1.872559E+000	132	-6.019612E+001	184	4.997559E+000	184	-5.686839E+001
133	3.122559E+000	133	-6.326794E+001	185	4.372559E+000	185	-5.469798E+001
134	3.747559E+000	134	-5.939622E+001	186	3.122559E+000	186	-5.008392E+001
135	4.997559E+000	135	-5.996194E+001	187	2.497559E+000	187	-5.020171E+001
136	4.997559E+000	136	-6.130967E+001	188	1.247559E+000	188	-4.798387E+001
137	4.997559E+000	137	-6.132050E+001	189	-2.441000E-003	189	-5.038269E+001

محللّ طيّف يعتمد على المتحكّم المتناهّي الدقة

عمر حسين حمد **د. إسماعيل عبدالله محمّد** **اميرة علي سلمان**
قسم الهندسة الكهربائية/كلية الهندسة قسم الهندسة الكهربائية/كلية الهندسة قسم هندسة السيطرة و الحاسبات
جامعة بغداد جامعة بغداد الجامعة التكنولوجية

الخلاصة:

يتضمن البحث تصمّيم و بناء وفحص محللّ طيّف يعتمد على المتحكّم المتناهّي الدقة كقاعدة للمعالجة. إن كلاً من البناء المادي و البرمجيات بُنيت لتحقيق الوظائف الرئيسية المطلوبة من هكذا نظام، والتي تصميمهما أشتمّل على ما متوّفر ومتاح من الأدوات لإنجاز وظائف ذلك الن ظام بحيث يكون ذا سماحية للتبديل أو التعديل من أجل تغيير معين في محيط التطبيق. إن هذه التقنية تعتمد أساساً على طرق التحليل الطيّفي التي تتخذ من تحليل فورير كقاعدة مع عمليات التكييف المسبق الضرورية و المطلوبة. يتكوّن البناء المادي للمحّلل من نوعيتين من الدوائر الإلكترونية:-

- دوائر إلكترونية أساسية وهي المعالج الدقيق ،الدوائر المنطقية المتعلقة به، وسائط الخزن، ودوائر الاتصال مع الحاسبة المركزّية.
- دوائر إلكترونية تتغير تبعا للتغير في التطبيق وهي دوائر الإدخال و الإخراج.

يتكوّن البناء البرمجي للمحلل من نوعيتين من البرامجيات:-

- برامجيات أساسية لا تعتمد على التطبيق وهي برنامج إدارة البرامج الأخرى، برنامج السيطّرة وحدات استحّصال البيانات و برنامج الاتصال مع الحاسبة المركزّية.
- برامجيات تتغير وفقا للتطبيق وهي برامج التطبيق و برنامج مسّوق وحدات الإدخال و الإخراج.

كذلك تمّ إعداد وتهيئة منظومة إعداد وتهيئة و تطوير برامجيات التطبيق باستخدام حاسّوب شخّصي (**IBM**) ومسيطر العمليات المقترح و على مستويين:

- مستوى الحاسّوب الشخصي.
- مستوى المحّلل.