Report No. 246 COO-1469-0069

AN ANALOG INPUT/OUTPUT SYSTEM FOR THE ILLIAC II

by

Alton Benjamin Otis, Jr.

September 22, 1967

DEPARTMENT OF COMPUTER SCIENCE · UNIVERSITY OF ILLINOIS · URBANA, ILLINOIS

DISCLAIMER

Report No. 246

AN ANALOG INPUT/OUTPUT SYSTEM FOR THE ILLIAC II*

by

Alton Benjamin Otis, Jr.

September 22, 1967

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

* This work was supported in part by Contract No. US AEC AT(11-1) 1469 and was submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering, August, 1967.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The analog input/output (I/O) facility of the University of Illinois Illiac II digital computer was designed primarily for the processing of audio or other analog signals lying in the frequency band 0 to 20,000 Hz. The facility consists of an audio tape recorder/reproducer, which is the source or receiver of analog signals, and an analog I/O interface that accomplishes both analog-to-digital (A/D) and digital-to-analog (D/A) signal transformation. Control of the audio tape machine and transfer of digitized analog signal information to or from the Illiac II are two additional functions of the analog interface, and the latter operation may be activated only by command from a program in the Illiac II.

By using the Illiac II in conjunction with the analog I/O facility it is possible to perform detailed Fourier analysis of complex audio wave forms, such as those produced in speech or music, and to re-synthesize these wave forms. Simpler processes such as time and amplitude transformations of existing audio signal information or generation of artificial signals are also possible.

Systems similar to the Illiac II facility are in use at several locations, notably Stanford University and Bell Telephone Laboratories. The fields of research being investigated at the University of Illinois and these locations generally fall into two categories: Namely, 1) Analysis and synthesis of speech and musical instrument tones and 2) generation of electronic music.

I.1. The Analog I/O Interface

The primary purpose of the analog I/O interface is to perform transformations between analog and digital signals. During the input operation, the interface produces digital values (called samples) within uniformly spaced time intervals (called sample periods). Each digital value represents the amplitude of the incoming audio signal at the beginning of the corresponding time interval. The Illiac II simultaneously reads each sample into its memory and thereby obtains a stair-step digital approximation in time of the actual analog signal. Analog output is performed by an inverse process in which the Illiac II periodically supplies the interface with digital samples (representing a stair-step approximation to the actual signal), which the interface converts to analog voltage levels. The resulting stair-step output voltages are filtered to produce the desired smooth waveform.

The transformations performed by the interface are executed by A/D and D/A converters. The D/A converter consists of a flip-flop register with the output of each flip-flop connected to a Kirchoff voltage adder. When a binary number is stored into the register by the Illiac II, the voltage at the output of the voltage adder becomes numerically proportional to the value of the binary number.

The A/D converter is more complex since it uses a process of successive approximations to obtain the result. While there are many different A/D conversion techniques[1] the most common method is to systematically alter the input to an internal D/A converter until the output of the D/A converter equals the value of the input voltage. The binary number that produces this result is the required digital value.

I.1.1. Precision of Representation

The precision to which the stair-step digital approximation represents the actual analog signal is determined by the resolution of the interface in both amplitude and time.

I.1.1.1. Amplitude Resolution

Amplitude resolution is a function of the number of binary digits (bits) used to represent the analog signal amplitude at each sampling period. As the number of bits used to represent a number increases, the fractional error in its value caused by an uncertainty of $\pm 1/2$ in the least significant bit (lsb) decreases. Since a similar uncertainty also exists in the analog signal amplitude due to additive noise, in practice it is only necessary to use enough bits in the digital representation so that the uncertainty in the lsb is below the noise level of the analog signal. To compute the "signal-to-noise (S/N) ratio" for a

---

[1] Stephenson, B.W., Analog-Digital Conversion Handbook, Digital Equipment Corp., Maynard, Mass.

digital sample value, consider a sample represented by n bits plus sign. The maximum value is $\pm 2^n-1$ and the "noise" value is $\pm 1/2$. The S/N ratio for the sample is thus

$$S/N = 20 \cdot \log_{10} \frac{2^n-1}{1/2}$$

$$\approx 20 \cdot \log_{10} 2^{n+1}$$

$$\approx 6 \cdot (n+1) \quad \text{decibels} \qquad \text{(I-1)}$$

In the Illiac II system 12 bits (13 bits including the sign) are used for conversion. This is equivalent to a S/N ratio of 78db, which far exceeds the present 65db S/N ratio of the tape recorder/reproducer and allows excellent amplitude representation. In addition, the 13 bit samples are optimally packed 4 per word inside the Illiac II memory since the word length for this computer is 52 bits.

I.1.1.2. Time Resolution

Time resolution is a function of the length of the sampling period or the sampling rate (sampling rate = 1/sampling period) of the conversion system used. If an analog signal is presumed to consist of frequencies in the band of 0 to f Hz then the sampling theorem[2] requires that a sampling rate of at least 2f be used in order to resolve all frequencies within the band. When the sampling rate is 2f, frequencies greater than f are also mapped into the range 0 to f by the sampling process and thus cause a distortion in the

---

[2] Susskind, A.K., Notes on Analog-Digital Conversion Techniques, MIT Press, Cambridge, Mass.

digital representation. In this mapping process, called "frequency foldover", the frequencies in the ranges nf to (n+1)f (n=1,2...) are linearly mapped onto the range 0 to f if n is even and onto the range f to 0 if n is odd.

The Illiac II system is capable of operating at a sampling rate of 40,000 Hz which permits accurate representations of audio signals in the range of 0 to 20,000 Hz. Sampling frequencies of 10,000 Hz, 20,000 Hz, and 30,000 Hz are also available for lower bandwidth signal processing.

I.1.1.3. Over-ranging

With a system which uses digital representations for analog signal sample values it is necessary that the amplitude and spectral frequencies of the signals to be processed lie within certain specified limits. When these limits are exceeded, over-ranging in both frequency and amplitude may distort the signal. The results of such actions are 1) frequency foldover when the bandwidth of the signal exceeds 1/2 the sampling frequency and 2) amplitude clipping of waveforms when the maximum amplitude limits are exceeded. In the case of analog input, the first type of over-ranging can be corrected by applying low pass filtration and the second type by amplitude attenuation of the analog signal. These operations restrict the signal so that it conforms to the limits imposed by the finite sampling rate and the finite digital word length of the I/O interface system. In the case of analog output it is up to the programmer to in-sure that the signal does not exceed the over-ranging limits in either frequency or amplitude.

I.2. Digital Data Storage

Digital samples are transferred to or from the Illiac II one at a time and at a rate determined by the sampling frequency of the analog I/O interface. While the samples could be actively stored and processed in core memory, this is not usually feasible because of the limited capacity of the Illiac II memory (8192 words or 32,768 samples) for any of the sampling frequencies used.

As an alternative, the samples may be stored on one of the three available secondary storage devices (namely, magnetic drum, disk, or tape) in "records" of about 1000 to 4000 samples. With such a scheme, overall signal processing is accomplished in three steps: Namely, 1) analog input onto the secondary storage device, 2) processing of the data or creation of artificial data, and 3) analog output from the secondary storage device. While the use of secondary storage is more time consuming than a direct "in core" processing approach, it provides for a much more flexible system.

I.2.1 Selecting the Storage Device

The selection of a secondary storage device depends on the type of processing to be done and is determined by 1) the maximum sampling rate to be used and 2) the maximum number of samples to be transferred. We will assume that the maximum time required for the storage device to "get to" a record of samples (the access time) is t_a seconds and that the time to transfer the record into core memory is t_b seconds. If the sampling rate is f samples/second and the record size is n samples then the inequality

$$t_a + t_b < n/f \tag{I-2}$$

must be satisfied for the storage device to be usable. Also, assuming that equation I-2 is satisfied, the maximum storage capacity of the device must be at least as great as that required by the processing application.

For the magnetic drum[3] t_a= 16.7 ms (the time for one revolution), t_b= 2.08 ms, and n is fixed at 1024. It follows that the maximum usable value of f is greater than 40,000 so that the drum can be used at any of the available sampling rates. On the other hand, the drum is limited by its storage capacity of approximately 260,000 samples, which allows only 6.5 seconds of audio signal storage at the 40,000 Hz sampling rate.

The magnetic disk[4], an IBM 1302 disk file, has a total capacity of over 20 million samples, which would allow more than 8.5 minutes of continuous audio signal storage at the 40,000 Hz sampling rate if this rate could be maintained.

---

[3]Gillies, D.B., Special Registers and Interrupts for the Illiac II, University of Illinois, DCS, Urbana, Illinois.

[4]Willard, R.E., Programming Aspects of the Disk File Channel, University of Illinois, DCS, Urbana, Illinois.

The access time, however, may range from 33.3 ms to 0.18 sec with this device, depending on the location of the record, while the record transfer time is approximately 25.6 ms. Since $n = 1024$, the maximum value of f is less than 20,000 making the disk unusable at the higher data rates where its total capacity is required. Therefore, for very large storage requirements, magnetic tape must be used.

The Illiac II has available two magnetic tape channels[5] and has the capability for transferring records with lengths equal to some multiple of 1024 samples. As is explained below, both tape channels can be used instead of one in order to reduce the effect of the record access time for one unit.

Each tape drive (IBM 729 model VI) has a maximum access time of $t_a = 10$ ms (the inter-record gap time) and a record transmission time of $t_b = n/40{,}000$ seconds assuming use of the 800 bits/inch Illiac II binary format. The inequality stated by I-2 thus reduces to:

$$f < \frac{40{,}000\ n}{n+400} \qquad \text{(I-3)}$$

indicating that for $n > 1200$, a sample rate of 30,000 samples/second can be used, but that for $f = 40{,}000$, n must be infinite.

Both tape channels can be used to achieve the 40,000

[5] Pisterzi, M.J., Programming Aspects of the Magnetic Tape Interplay Channels, University of Illinois, DCS, Urbana, Illinois.

samples/second rate by overlapping the time for data transmission on one channel with the record access time on the other channel. With such a system, sample records which are adjacent in time are recorded alternately on two different tapes in such a way that if the records were numbered 1, 2, 3 ... in time, then all odd-numbered records would be on one tape and all even-numbered records on the other.

The storage capacity of a single 2400 foot reel of magnetic tape is approximately 8 million samples which allows more than 4.5 minutes of continuous audio signal storage at 30,000 samples/second with a single tape system or over 6.7 minutes at 40,000 samples/second with a two tape system.

I.2.2. Data Transfer Errors

Errors occurring in the transfer of information between drum or tape and core memory fall into two categories: Namely 1) parity errors and 2) wrong-length-record errors (tape only). Parity errors generally indicate an error in one or more bits of information and can be ignored as "noise" for most applications. Wrong-length-record errors, however, indicate a loss of one or more words (4 samples) of information during reading. This kind of error is generally due to an initial mispositioning of the tape and can be corrected by subsequent backspacing and re-reading. During analog I/O such operations cannot be performed in real time. Therefore, the error (which often produces an audible "click") might not be ignorable, depending on the application.

For the two tape system, a wrong length record introduces a far worse secondary effect. If the error was caused by a digital tape dropout (loss of information) instead of mispositioning, the tape channel may consider the dropout to be the end of the record. In this case the remainder of the record may be interpreted (by the tape channel) to be another record, thereby destroying the sequencing between the two tapes. Unless further information is provided in each record, such as a sequence number, this error is sufficient cause for termination of the analog output operation since the remainder of the analog signal would be hopelessly distorted.

I.2.3. Programming Analog I/O

Programs to perform continuous analog I/O with magnetic drum or tape require the use of a multiple buffering scheme so as to provide a continuous flow of samples between the Illiac II and the analog interface while discrete records are being transferred to or from the secondary storage device. To indicate the general procedure, consider a program using a single tape as its secondary storage device with records of 512 words (or 2048 samples). Two buffers, each 512 words long, are set aside in core memory. During an analog input operation the program reads incoming samples from the analog interface, packs them 4 per word, and stores each filled word in buffer 1. When buffer 1 is full the program sends

I/O commands to the tape channel (a direct-memory access device) to cause the contents of buffer 1 to be written onto tape. Once the commands are given, the program then proceeds to fill buffer 2 (as it did buffer 1) while the tape I/O channel independently writes buffer 1 onto tape. After buffer 2 is filled, the program checks the I/O channel to make sure the previous transfer is complete and then commands the channel to write buffer 2 onto tape. The program repeats the process by filling buffer 1 again and so forth, allowing the continuous stream of input samples to be separated into discrete records. Analog output is performed by the inverse process.

A two tape system uses a scheme similar to that for a single tape with the exception that four buffers are required to maintain the desired data flow instead of two. Therefore, core storage equivalent to four times a single record length must be reserved for the analog I/O operations.

It should be mentioned that the <u>key</u> to implementation of a program for analog I/O to or from digital tape, is the direct-memory-access capability of the tape channels. Without this capability the tape drives and the computer's central processor could not work independently in order to maintain a continuous flow of data. The same type of channels are also provided for the magnetic drum and disk units allowing either to be used, if required, in place of a single tape in the procedure described above.

II. A DESCRIPTION OF THE ILLIAC II ANALOG I/O SYSTEM

II.1. Operation of the Analog I/O Interface

The basic structure of the analog I/O interface and its interconnection with the Illiac II is shown in Figure 1. All I/O operations performed by the interface are controlled with a program in the Illiac by means of a double flip-flop register known as "special register 58" or SR58. A second identical register ("special register 54" or SR54) is used to transfer the digital sample data to and from the interface.

Each double register is 13 bits in length and actually consists of a 13 bit output "side", into which the program can set or store information from memory, and a 13 bit input "side", that the program can read into memory. The interface, however, can store information into the input side, and read information from the output side of each of the two registers, allowing samples to be transferred through SR54 and control information to be transferred through SR58.

Proper sequencing of information transfers is provided by four control signals that are associated with each of the special registers. Two of these (called ASN and FULL) are generated in the special register control section of Illiac II and are sent to the interface. The other two (called EMI and RFULL) are generated by the interface and sent to the special register control.

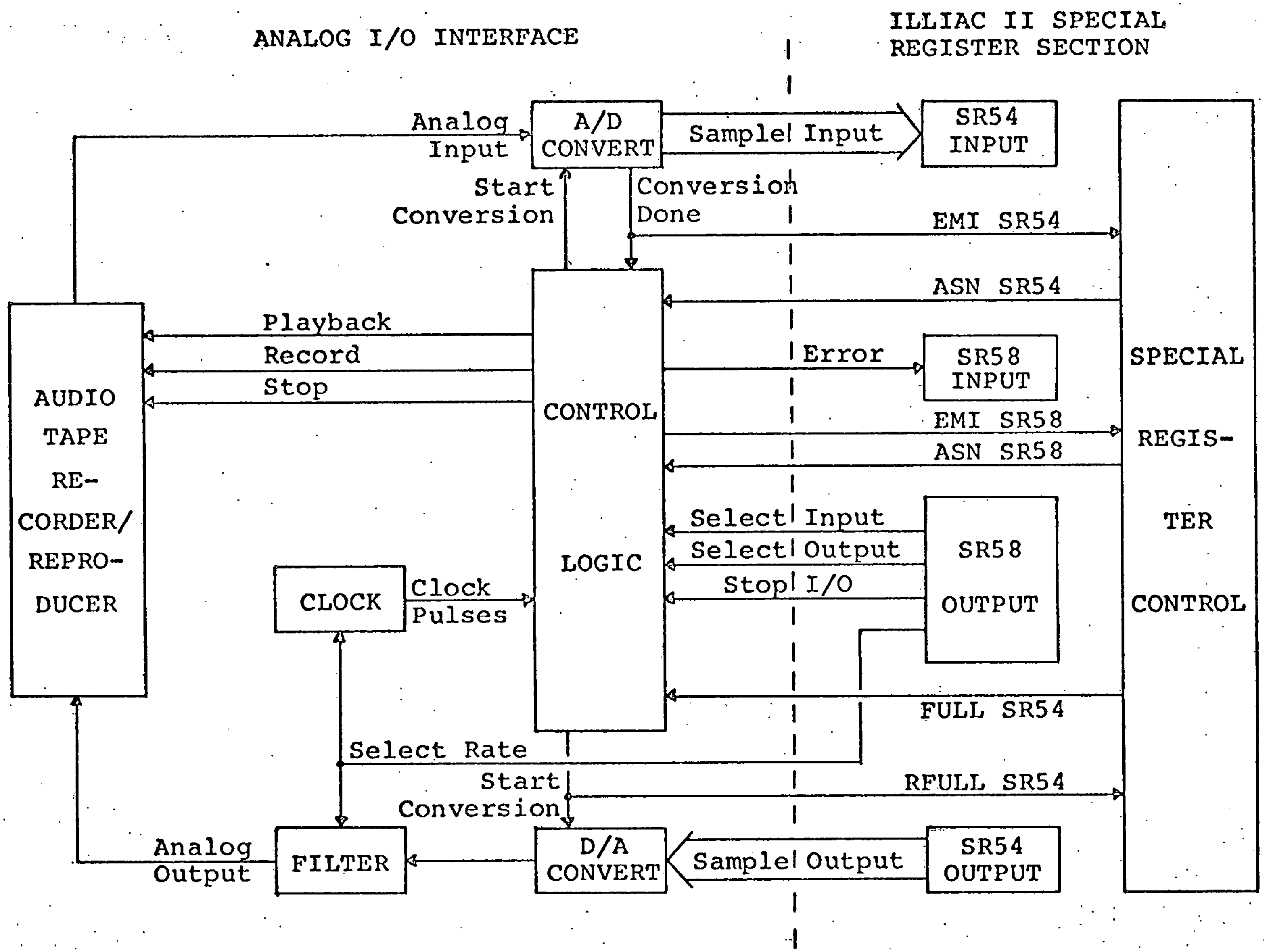


Figure 1. The Analog I/O Interface

The signals ASN and EMI are used to control the input sides of the registers. The signal ASN occurs when a program in the Illiac reads the input side of either register and is used to tell the interface that new data may be stored. The signal EMI causes the new data to be stored and "interrupts"[6] the Illiac. This latter action is used by the program to determine when new data is available. The signals FULL and RFULL perform similar control functions for the output sides of the registers. The signal FULL occurs when a program stores new data into the output side of either register, and is used to tell the interface that new data is available. The signal RFULL resets FULL and tells the special register control section that new data may be stored. When FULL occurs, further attempts by the program to store new data into the special register will cause the Illiac to stop the program until RFULL is sent. In this manner the previous data cannot be destroyed until used by the interface.

The interface itself consists of 1) the A/D and D/A converters, 2) control logic to sequence the I/O operations, 3) a clock pulse generator to provide the proper sampling rate, 4) filters to smooth the analog output, and 5) the audio tape recorder/reproducer. Program control of the interface is performed by setting bits in the output side of

---

[6]Gillies, D.B., Special Registers and Interrupts for the Illiac II, University of Illinois, DCS, Urbana, Illinois.

SR58. The control functions allow the program to either start or stop analog I/O, and to simultaneously select one of four sampling rates and its associated analog output filter. (The RFULL signal for SR58 is continuously sent to the Illiac allowing the program to change the output side of this register at any time).

During analog I/O, the interface control logic checks to make sure that the program is keeping up with the sampling rate selected (i.e., that it is reading samples in or sending them out fast enough). If either the program lags behind or the A/D converter does not finish conversion in time, the input side of SR58 is set to indicate an error condition. When the program checks the error condition flag by reading SR58, the ASN that is sent back to the interface control logic, resets the error condition and causes SR58 to be cleared.

II.1.1. Analog Input

Analog input is started by setting the bit in SR58 which produces the signal "Select Input". When "Select Input" goes on, the control logic starts the audio tape machine in the playback mode and begins A/D conversion and error checking. The correct sampling rate is determined by setting two bits in SR58 which produce the signal "Select Rate" that in turn controls the clock frequency.

At the beginning of each sample period, the clock sends a pulse to the control logic, which in turn checks for errors (as previously described) and sends a "Start Conversion" signal to the A/D converter. At the completion of conversion a "Conversion Done" signal is returned by the A/D converter and is sent simultaneously to the control logic for error checking and to the Illiac as the EMI signal for SR54. The latter action indicates that a new sample is ready. The ASN signal from SR54, which indicates that the new sample has been read, goes back to the control logic for error checking.

II.1.2. Analog Output

Analog output is started by setting the bit in SR58 which produces the signal "Select Output". When "Select Output" goes on, the control logic starts the audio tape machine in the record mode and begins D/A conversion and error checking. The sampling rate is selected in the same manner as for analog input.

At the beginning of each sample period, the clock pulse causes the control logic to send a "Start Conversion" signal to the D/A converter which then reads and converts the value stored in the output side of SR54. The same signal is sent to the Illiac as RFULL for SR54. This action allows the program to store a new value. When a new value has been stored, the FULL signal that is returned from SR54 is sent to the control logic for error checking.

II.1.3. Termination of Analog I/O

Analog I/O is terminated when the program resets the output side of SR58 so that both "Select Input" and "Select Output" are turned off and "Stop I/O" is turned on. When both "Select Input" and "Select Output" are turned off, transmission of the clock pulses through the control logic is prevented, and all conversion and I/O operations from the interface are halted.[7] The "Stop I/O" signal is required by the control logic in order to automatically stop the audio tape machine after it has been started in the record or playback mode by the previous analog I/O operation.

Detailed information concerning the interface "hardware" is given in Appendix A.

II.2. Programming the Analog I/O Interface

The programming of analog I/O is performed by use of the ASN and SSR instructions, which allow a program to read data from the input side and store data into the output side of a special register. Thus, the program can transmit samples via SR54 and control the interface and check for errors via SR58.

The major problem that arises in programming is to keep in step with the interface. During output it is only

[7] If both "Select Input" and "Select Output" are turned on, the interface will attempt to perform both input and output but the audio tape machine will be set to record.

necessary for the program to execute a store (SSR) instruction for SR54 before the end of each sample period, thereby allowing the Illiac to guarantee synchronization by holding up the program until the end of each period. Since this cannot be done during analog input, a third special register, SR14, is available in the Illiac to tell the program when an EMI has been sent to certain of the other special registers (including SR54 and SR58).

By use of SR14, the program can tell when the analog I/O interface has sent an EMI to SR54 indicating the end of conversion and the availability of a new sample. In this manner proper synchronization can be maintained to prevent loss of samples.

II.2.1. Using the Digitized Samples

Samples sent to and from the interface are 13 bit binary numbers in 2's complement notation. The values range from +4095 for the positive voltage maxima to -4095 for the negative voltage maxima with zero equaling zero volts.

Due to the 13 bit quarter word structure of the 52 bit Illiac II words, the samples are conveniently packed 4 per word for compact storage. For computational purposes, the samples may be easily converted to normalized floating point and inversely. Single instructions perform the conversions automatically when the range is restricted to absolute values of less than 1.0 in the floating point form.

Detailed information concerning the programming of the interface is given in Appendix B.

II.3. The Analog I/O Programming System

The analog programming system was written to make the Illiac II analog I/O facility available to those users who are unfamiliar with machine language programming. The system consists of a program that performs continuous analog I/O using digital magnetic tape as the secondary storage medium, and a set of subroutines that allow these tapes to be read and written by user programs written in FORTRAN.

The first version of the analog programming system used both tape channels and two tapes in order to allow analog I/O at the 40,000 Hz sampling rate. Use of the system indicated that its major drawback was frequent occurrences of wrong-length-record errors, which necessitated premature termination of analog output operations. As a result, a second version was written using a single tape and long records. This bypasses the need for program termination when a wrong length record is encountered. The new system allows use of all available sampling rates except for 40,000 Hz.

The original system, consisting of the analog I/O program (called ADSYS) and the tape I/O subroutines for FORTRAN (called ADPAK), is available to users requiring the 40,000 Hz sampling rate. A complete description of the use of the programs is given in reference 3 of the bibliography. The programs in the new system (called ADSYS2 and ADPAK2, respectively) are described below.

II.3.1. ADSYS2

ADSYS2 is a machine language program to perform analog I/O to and from a digital magnetic tape supplied by the user. The user's tape consists of a set of sample files each containing one or more records of digital samples in a format that can be used by the ADPAK2 subroutines. The sample records in each file constitute the digital representation of a continuous analog signal beginning at the start of a file and terminating at the start of the next sequential file. In this manner, more than one continuous analog signal may be stored on a tape so that the handling of large numbers of signals is made convenient. (The original version allowed only one continuous signal per pair of reels). In addition to the sample records, each file contains a file label record and each tape contains a tape label record as a means of uniquely identifying the tape and the files stored on it.

In actual operation analog I/O is not performed directly onto or from the user's tape because the records used are too short to maintain a 30,000 Hz sampling rate (See Section I.2). Instead an intermediate tape (called the conversion tape) is used on which long records are stored. To perform input the user specifies to ADSYS2 the sample rate and the total time of input and then requests that the operation proceed. When the input is finished, the digitized signal is stored on the conversion tape as a single file of long records. If the user wishes to save the information he may then request ADSYS2 to copy the conversion tape into a file on the user's file tape, thereby making it available for processing. Output is performed inversely by first requesting a file to be copied from a file tape onto the conversion tape as a single file of long records and then requesting analog output. In either case, the copying operations properly convert the record sizes so that no samples are lost.

By use of the conversion tape it is possible to provide users with access to the 30,000 Hz sampling rate. In addition, sample records stored on the conversion tape are not destroyed (during a single loading of the program) until over-written by a copy or input operation. Thus it is possible to check the digitized analog input samples by doing analog output and performing a visual or aural check on the converted signal before transferring the samples to the file tape. It is also possible to retry faulty analog output operations without additional copying.

II.3.1.1. Using ADSYS2

All operations performed by ADSYS2 are determined by a set of commands which are communicated to the program via a teletypewriter console connected on-line to the Illiac II. The six basic operations specified by command are

1) labeling of a new file tape, 2) copying a file onto the conversion tape, 3) copying the conversion tape onto a file tape, 4) analog input, 5) analog output, and 6) internal generation of sine and sawtooth tones for analog output. Each command is input to the program by typing the name of the operation to be performed followed by a string of job parameters. The job parameters are used to tell ADSYS2 such information as the sampling rate, the time limit for I/O, the file to be copied into or from, which tape drive the file tape is on, and so forth.

When a command is given, the system verifies all the input parameters and, if tape copying or labeling was specified, generates or checks the tape and file labels on the user's file tape and then performs the operation. For analog I/O or tone generation a message is then sent to the on-line teletypewriter indicating the system is ready to perform the requested operation. The operation, however, does not proceed until a switch is set on the Illiac II console. This gives the operator time to mount any required audio tape.

Once started, an operation continues until it is completed, (in the case of copying and tape initialization) or it continues until an end condition is detected (in the case of analog I/O). End conditions may occur if the console switch is reset, the specified time limit is exceeded, or for other reasons. Upon termination, ADSYS2 sends a final message to the on-line teletypewriter indicating that the operation has been completed. It is then prepared to accept the next command. All tape errors occurring during analog I/O are ignored; however, tape errors detected on the file tape during copying operations cause the record to be re-read or re-written several times before being ignored. Analog interface errors are counted and output with the final message but are otherwise ignored. Detailed information regarding the tape record formats and the command language is given in Appendix C.

II.3.2. ADPAK2

ADPAK2 is a collection of machine language subroutines that allow programs written in FORTRAN (or machine language) to read and write the file tapes used by ADSYS2. The subroutines are divided into two groups with four subroutines in each group. One group is used to input an existing file of samples while the other is used to create a new file of samples. In addition, a ninth subroutine provides the means for creating a tape label on a previously unused file tape.

The subroutines in the input/output groups are used to "activate" and "deactivate" files on the file tape, to transfer sample records between core memory and tape, and to skip over sample records or generate sample records containing all zeros. A file is activated by first checking the tape label and then searching the tape and checking each file label until the requested file is found. If the requested file is not found and the file is being activated for input, the program terminates. If, however, the file is being activated for output, a new file is created at the end of the last file on the tape and the program continues. Once activated, the file remains active until it is deactivated. During this period, sample records may be read or written and record skipping or zero record generation may be performed.

A file is deactivated by writing the proper file termination marker (in the case of an output file only) and/or rewinding the tape on which the file is located. Deactivation occurs automatically when the end of an active file is reached during the transfer of sample records or can be requested by calling an appropriate subroutine. Attempts to read or write sample records when there is no active input or output file, respectively, are ignored.

The samples transferred between core memory and tape are written or read in records of 1024 samples each. ADPAK2 performs a conversion so that all input samples are stored as normalized floating point numbers with absolute values less than 1.0. On output, the inverse conversion is performed and samples whose absolute values are equal or greater than 1.0 will cause accumulator overflow.

All tape errors cause the associated read or write operation to be retried a maximum of 10 times. If the error persists on the 10th retry, it is ignored, and a flag is returned to the calling program indicating the error. Detailed information regarding the required calling sequences is given in Appendix D.

III. TWO EXAMPLES OF USAGE OF THE ILLIAC II ANALOG I/O FACILITY

III.1. Current Status

The analog input/output system on the Illiac II is currently being used by the departments of computer science, music, and electrical engineering. This work includes Fourier synthesis of musical instrument tones and various experiments in music generation and processing. In the latter category are the processes of time-rate changing and synthetic reverberation, which will be discussed in detail below. Future plans include research in speech analysis and digital processing of analog data from experimental apparatus.

III.2. Time-Rate Changing

Time-rate changing is a process in which the duration of a given audio signal is compressed or expanded in time without changing the audible frequencies. In a musical context this means that the tempo of the music is changed while the pitches remain unaltered. This process differs from the more common time-change process accomplished by playing back a recorded signal at a rate different from that at which the signal was recorded. In this case the audible frequencies, as well as the tempo, vary in direct proportion to the playback rate.

Time-rate changing can be performed by a special analog apparatus using a variable speed capstan drive and a cylindrical revolving head with four gaps equally spaced about its circumference. In operation the capstan and tape head speeds are adjusted to maintain the same gap-to-tape speed as was used to record the sound. Since the tape covers slightly less than one-fourth of the revolving head's circumference only one gap is in contact with the tape at any time. Thus, since during playback the tape is actually moving faster or slower with respect to the axis of the head than it was when recorded, a small segment of the tape is either deleted or doubly reproduced as one head leaves and another comes in contact with the tape surface. Those segments which are reproduced form a piece-wise continuous signal that is shorter or longer in duration than the original, but, due to the proper gap-to-tape speed, the audible frequencies are the same as those of the original. The distortion which results from such a process depends on 1) the rate at which the segments are deleted or doubled (called the "segmentation rate"), 2) the degree of continuity between adjacent reproduced segments, and 3) the percentage by which the durations are changed (called the "degree of rate-changing").

The segmentation process is essentially equivalent to a time modulation. As a result minimum distortion in the signal will occur when the segmentation rate is as far as possible below the lowest spectral frequency component of the signal. However, as this rate is decreased, the individual reproduced segments can be audibly distinguished as a "choppy-like" sound pattern. Therefore, it is necessary to select a segment length shorter than can be recognized by ear but longer than the period of the lowest frequency to be processed. For normal musical material, segmentation rates of 20 to 30 segments per second seem to provide the best compromise of these factors. (This may depend on the kind of music being processed).

The degree of rate-changing is in a sense the percentage by which the sound source is modulated, and the distortion introduced will be a function of the degree of rate-changing. The audible effects of this distortion range from near inaudiblity at small percentages (less than 5%) to a strong "warbling" of the sound at the segmentation rate. This latter condition results from imperfect matching of the amplitude values of the original signal at the boundaries between reproduced segments. If "perfect" boundary matching could be established, the rate changing process would be greatly improved. However, this would not only require matching the amplitudes but all of the derivatives of the signal at the boundary points. Schemes for matching the first few derivatives are described in the following section. It must be noted, however, that there is no possible way of achieving perfect rate-changing without considerable a priori knowledge of the signal, and even with such knowledge the algorithm for "optimal" rate changing may not be unique.

III.2.1. Computer Simulation of Time-Rate Changing

The time-rate changing experiment was designed to examine the aural effect of satisfying various types of boundary conditions between segments produced by the process described in III.2. To perform such a task with an analog system would have required a great deal of experimentation with various mechanical setups and head configurations. On the other hand, a digital simulation of the process allowed boundary conditions to be modified by simple program changes.

Simulating the analog rate changer was one of the simplest tasks to perform with the Illiac II analog-digital facility. In operation, musical selections were digitized and stored on digital tape using ADSYS. A 40KHz sampling rate was selected for maximum fidelity. Then, using ADPAK, the rate changing program would input into memory arrays of 1600 samples each, one at a time. This resulted in a segmentation rate of 25 segments per second. The specified percentage of samples was then deleted (for time compression) or doubled and joined (for time expansion) onto the end of a segment according to the appropriate boundary matching algorithm. The resulting modified segments were written by ADPAK onto the output tape which was then converted by ADSYS into the rate-changed audio signal.

Two basic schemes of boundary matching were investigated. With the first scheme the program was designed to match the signs of the first derivatives of the signal at zero axis crossings nearest the boundaries. In this case the program examined samples beyond the end of the rate-changed segment and beyond the beginning of the next segment until the required boundary match was obtained. As a result, the actual segmentation rate varied slightly from segment to segment depending on the signal. This caused a time-jitter distortion in addition to the distortion due to segmentation, which is reduced by this process.

With the second scheme the end of each segment was "blended" with the beginning of the next according to some distribution function. The actual algorithm for this type of time-rate changing is precisely stated as follows:

For time compression of a segment composed of samples S_i, for i = 1 to N, with n samples to be deleted, the resulting samples S_i', for i = 1 to N-n, are given by

$$S_i = \begin{cases} S_i & [i = 1 \text{ to } N-2n] \\ P_k \cdot S_i + (1-P_k) \cdot S_{i+n} & [i = N-2n+1 \text{ to } N-n; \\ & \quad k = i-(N-2n)] \end{cases} \qquad \text{(III-1)}$$

where P is a distribution function with values in the range 0 to 1 for k = 1 to n. For time expansion, where n samples are to be added to each segment, the resulting samples S_i', where i = 1 to N+n, are given by

$$S'_i = \begin{cases} S_i & [i = 1 \text{ to } N-n] \\ P_k \cdot S_i + (1-P_k) \cdot S_{i-n} & [i = N-n+1 \text{ to } N; \\ & \quad k = i-(N-n)] \\ S_{i-n} & [i = N+1 \text{ to } N+n] \end{cases} \qquad \text{(III-2)}$$

The process is shown schematically in Figure 2.

The distribution function, P, can vary from no blending (P_k = 1 or 0 for all k) to more complex functions that match the amplitudes and derivatives of the signal at the boundaries of the blending region. As an example, the function

$$P_k = 1-(k-1)/n \quad (k = 1 \text{ to } n) \qquad \text{(III-3)}$$

will match boundary amplitudes properly while the function

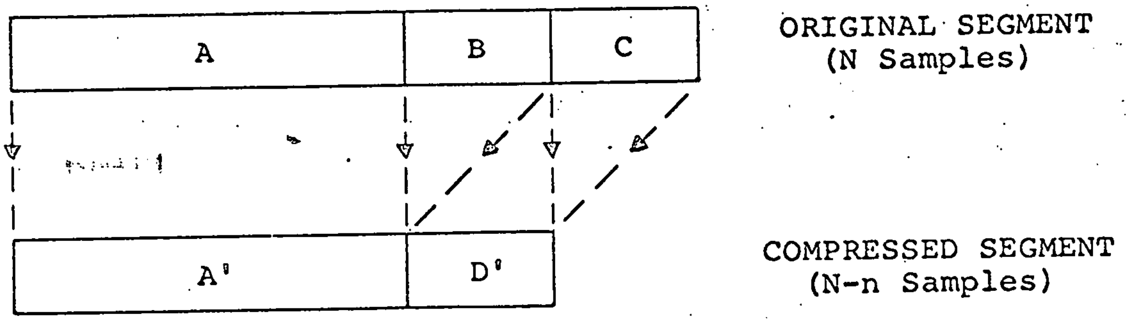
$$P_k = [1+\cos(\pi \cdot (k-1)/n)]/2 \quad (k = 1 \text{ to } n) \qquad \text{(III-4)}$$

will match both amplitudes and first derivatives properly. In general, if the ith derivative of P is zero at k = 1 and k = n for i = 1 to J, then the first J derivatives of the signal will be matched at the boundaries of the blending region. As more derivatives are matched, the boundary discontinuities are decreased but, at the same time, the derivatives of P in the middle of the blending region become increasingly larger. Since this factor also adds to the overall distortion, it is necessary to compromise and select a P that produces the best overall blending of the signal segments.

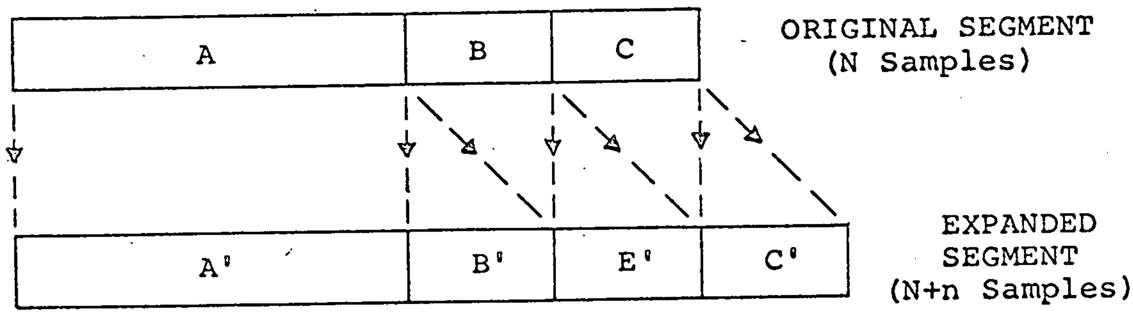
III.3. Synthetic Reverberation

Reverberation is a term applied to the collection of

TIME COMPRESSION



TIME EXPANSION



$$A = S_1 \text{ to } S_{N-2n}$$

$$B = S_{N-2n+1} \text{ to } S_{N-n}$$

$$C = S_{N-n+1} \text{ to } S_N$$

$$A' = A = S'_1 \text{ to } S'_{N-2n}$$

$$B' = B = S'_{N-2n+1} \text{ to } S'_{N-n}$$

$$C' = C = S'_{N+1} \text{ to } S'_{N+n}$$

$$D' = P \cdot B + (1-P) \cdot C = S'_{N-2n+1} \text{ to } S'_{N-n}$$

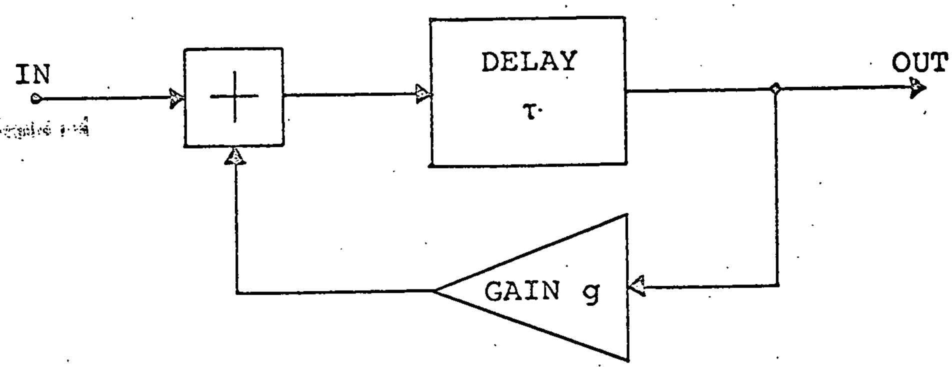
$$E' = P \cdot C + (1-P) \cdot B = S'_{N-n+1} \text{ to } S'_N$$

Figure 2. Time-Rate Changing Scheme

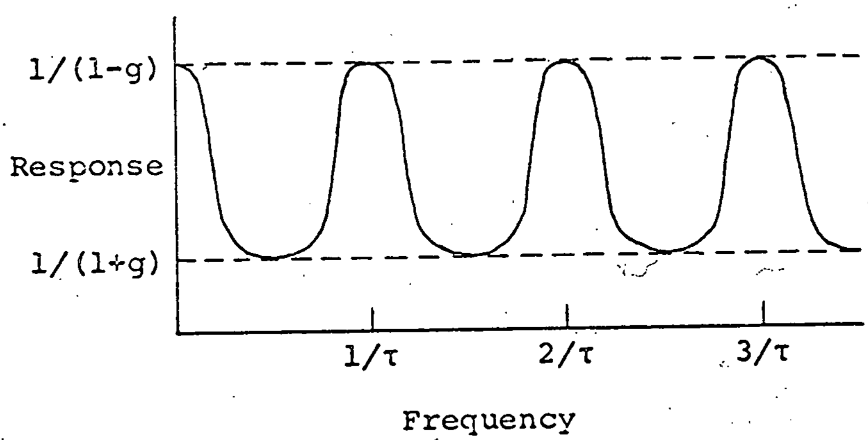
echoes which are added to an audio signal by sound reflections in an acoustical environment. By artificial means it is possible to simulate or supplement the reverberation produced by a concert hall. Several means of producing synthetic reverberation by electro-mechanical analog systems are currently available, but these are generally deficient in one or more aspects. In an attempt to circumvent these deficiencies, a detailed study of reverberation was carried out by M. R. Schroeder at Bell Telephone Laboratories.

Schroeder's analysis determined that the two main problems which characterize existing reverberation equipment are 1) non-uniform frequency response, and 2) insufficiently large random echo density. One cause of non-uniform frequency response in reverberators results from the use of the simple ideal reverberator circuit shown in Figure 3a. The response of this configuration is given in Figure 3b. While an increase in pseudo-random echo density is achieved by a series connection of such reverberators (set for different values of gain, g, and delay, τ), the frequency response of the final output signal is still seriously distorted.

As a result Schroeder developed an all-pass reverberator (Figure 4a). This unit has a uniform gain of 1 for all frequencies and allows one to cascade as many units as necessary to provide the required echo density

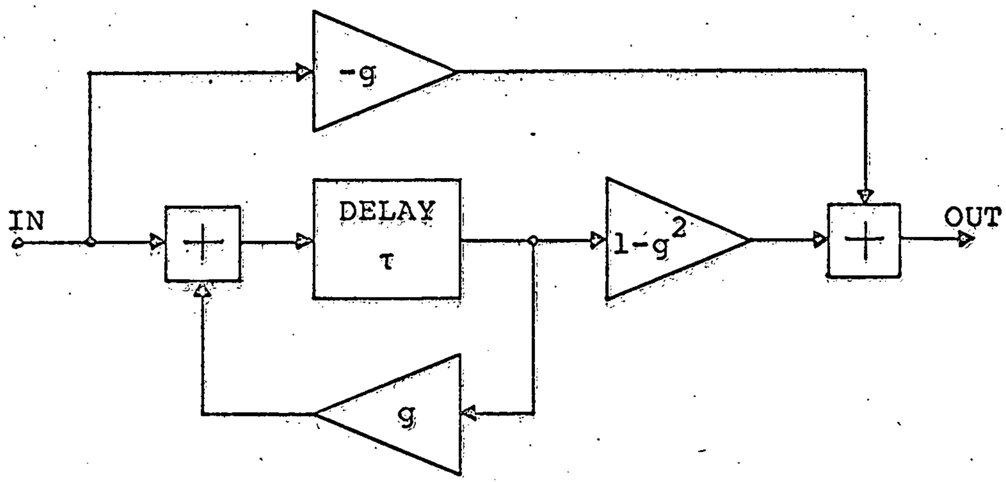


(a) Basic Circuit

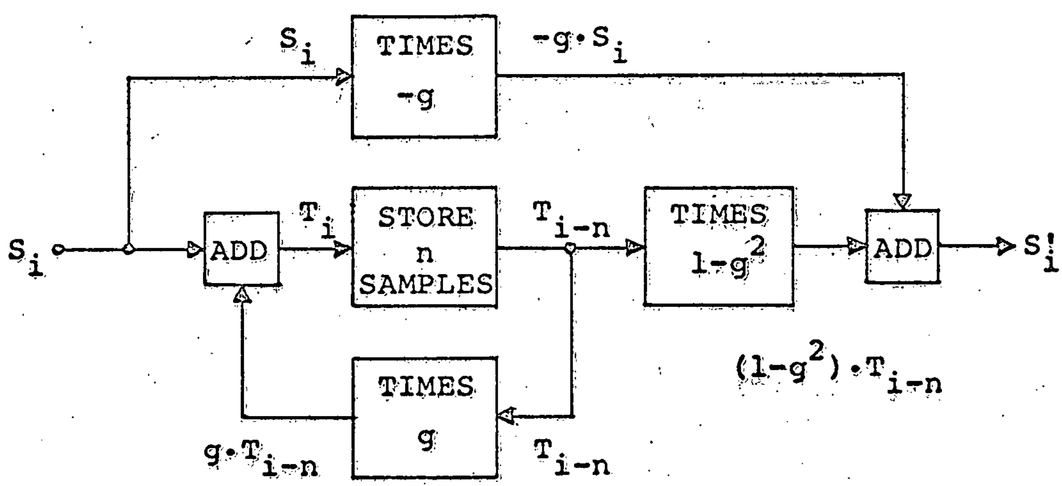


(b) Frequency Response

Figure 3. The Simple Reverberator



(a) Basic Circuit



(b) Digital Simulation

Figure 4. The All-Pass Reverberator

without altering the frequency response. Extensive experimentation by Schroeder indicated that the series connection of five all-pass reverberators with feedback gains of $\pm.7$ and selected delays between 5 and 100 msec will produce a reverberation effect that is comparable with that of a conventional concert hall.

III.3.1. Computer Simulation of Reverberation

Using the Illiac II analog-digital system, it was possible to artificially reverberate musical passages by simulation of Schroeder's all-pass reverberator. The overall logical scheme was similar to that used for the time-rate change experiment. Only the actual computational process was different. Simulation of the all-pass reverberator is performed by use of a table, treated circularly, which represents the delay element τ. If the maximum delay is to be m samples then the delay table must be m locations long. As each sample S_i is input to the reverberator, the corresponding output sample S_i' is produced by the following operation:

$$S_i = (1-g^2)\cdot T_{i-n} - g\cdot S_i \qquad \text{(III-5)}$$

where T_{i-n} is an entry in the delay table corresponding to the signal input to the delay element (see Figure 4b) at time τ seconds, or n samples, ago ($n \leq m$). For each new input sample, the corresponding entry in the delay table T_i is reset by the operation:

$$T_i = g \cdot T_{i-n} + S_i \qquad \text{(III-6)}$$

This provides the feedback around the delay element τ, as indicated in the figure. Note that the index for T is interpreted modulo m, the table length, to be compatible with the circular table interpretation. The simulation of a cascaded series of such reverberators is achieved by using the output of one pass of the simulator as input to the next sequential pass and so forth.

APPENDIX A. THE ANALOG I/O INTERFACE: EQUIPMENT, SIGNAL, AND LOGIC DESCRIPTIONS

The reader should be familiar with Section II.1 of this thesis before attempting to read the following information.

A.1. Interface Equipment

The A/D converter is an 8 bit bipolar unit manufactured by ADCOM (Model 208C). The input range maxima are ± 2.5 volts corresponding to sample values of +4064 and -4095 read through SR54. A change in the least significant bit is equivalent to a voltage change of ± .01954 volts corresponding to a digital value change of ± 32.[8] The maximum conversion rate is 100,000 samples per second.

The D/A converter is a 13 bit bipolar unit manufactured by Texas Instruments (Model 854). The output range maxima are ± 10 volts corresponding to sample values of +4095 and -4095 stored in SR54. Changing the least significant bit produces a voltage change of ± .00244 volts. The maximum conversion rate is 250,000 samples per second.

The tape recorder/reproducer is manufactured by Scully (Model 280). This is a half track, two channel unit that accepts 1/4" audio tape at speeds of 7-1/2 and 15 inches per second. The signal to noise ratio using Scotch 201 magnetic tape is better than 60 db. The frequency response is flat

---

[8]This device is to be replaced by a 13 bit A/D converter in the near future.

within $\pm$2db from 35Hz to 18KHz at 15 inches per second. Only the left tape channel (closest to the deck surface) is presently used for analog I/O.

A.2. Audio Signal Conventions

The tape recorder/reproducer has been calibrated for Scotch 201 recording tape. The record and playback level controls are correspondingly marked so that when set appropriately the following conventions apply:

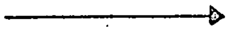
1) A maximum amplitude 1KHz sine tone output from the Illiac II ($\pm$ 4095) will produce a 0 db record level on the tape.
2) A 1KHz sine tone recorded at 0 db on the tape will be input to the Illiac as a 50% maximum value sine tone.

A.3. Digital Signal Conventions

The digital signals within the analog I/O interface and those sent to and from the Illiac are equal to one of two voltage levels at any particular time: 0 volts stands for logical 1 and -5 volts for logical 0. In certain portions of the circuitry the level signals are converted into short duration pulses for triggering and gating purposes.

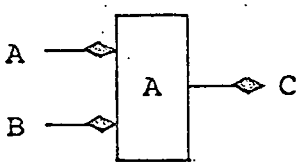
A.4. The Analog I/O Interface Circuitry

The complete logical diagrams of the clock, filter, and control sections of the analog I/O interface are shown in Figures 6 and 7. Definitions of the logic symbols

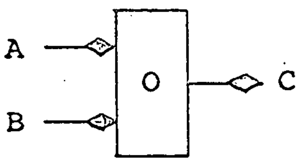


Pulse Signal
(logical 1 for 500 nsec after being logical 0 for > 700 nsec)

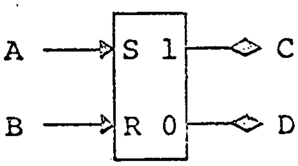
Level Signal
(remains logical 1 or 0 for > 700 nsec)



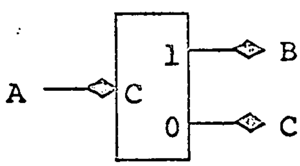
AND Gate
(C = 1 when A and B both = 1)



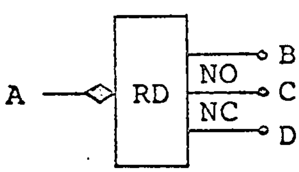
OR Gate
(C = 1 when either A or B or both = 1)



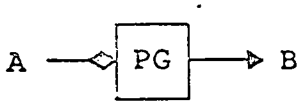
Set/Reset Flip-Flop
(C = 1, D = 0 after A = 1; C = 0, D = 1 after B = 1; output changes are delayed by 700 nsec)



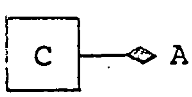
Complementing Flip-Flop
(B and C exchange values 700 nsec after A changes from 0 to 1)



Relay Driver and Relay
(C and D are connected when A = 0; C and B are connected when A = 1)



Pulse Generator
(B occurs when A changes from 0 to 1)



Crystal Clock
(A is a square wave at the clock rate)

Figure 5. Logic Symbol Definitions

used are given in Figure 5. The reader should be familiar with the logic conventions before reading further.

A.4.1. The Clock and Filter Circuits

The clock consists of a 30KHz and a 40KHz crystal oscillator providing a square wave output. The signal from the 40KHz oscillator is fed to a two stage counter which produces outputs of 20KHz and 10KHz. The four signals are then sent to a two stage relay tree that selects one of the four as the desired clock frequency. Selection is determined by the state of bits 2 and 3 (weight 1024 and 512) in the output side of SR58. A one stored in either bit will cause the associated relay driver to turn on. The square wave output of the relay tree is then sent through a pulse generator which produces 500 nanosecond pulses for use by the interface.

The filter circuit consists of two low pass R-C networks used with the 40KHz and 30KHz clock frequencies and two low pass L-C filters used with the 20KHz and 10KHz clock frequencies (manufactured by TT Electronics, Inc.). The frequency response characteristics of the R-C networks roll off with a -6 db/octave slope above 20KHz and 15KHz respectively. The L-C filters are characterized by -3 db cutoff frequencies of 8.5 KHz and 4KHz with 60 db attenuation frequencies of 17KHz and 9KHz, respectively. The output of the D/A converter is fed simultaneously through

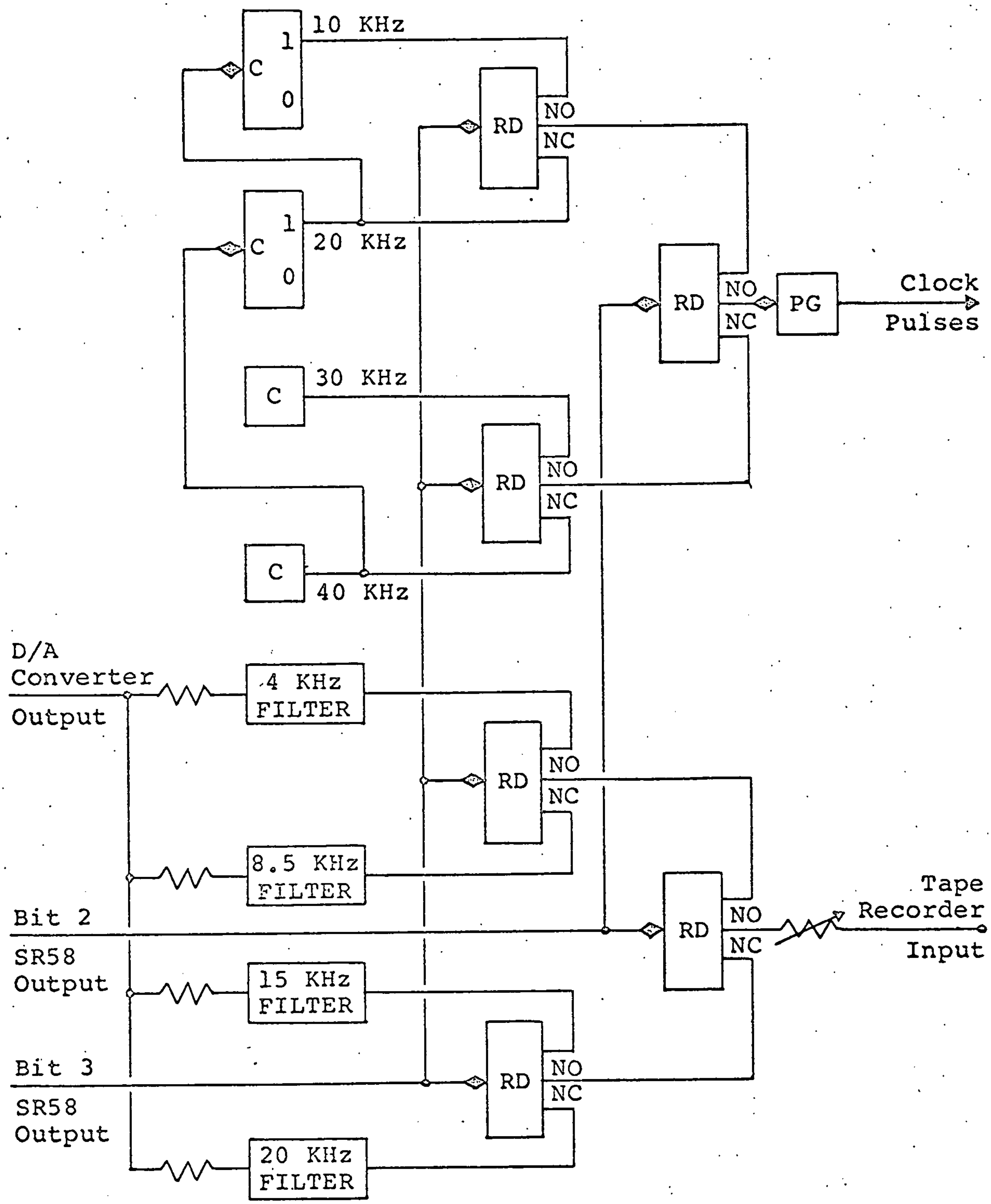


Figure 6. Clock and Filter Circuits

individual coupling resistors to the inputs of all four filters. The filter outputs enter a relay tree, identical to that used to select the clock frequency, and are terminated by an attenuater which supplies the signal to the tape recorder. Since the output impedance of the D/A converter is very low (≈ 0.1 ohms) the series resistors at the inputs to the filters provide both the proper source resistances for the filters and a decoupling amongst the four of them.

The clock and filter circuits are connected and operating at all times independent of what analog I/O is being performed. The actual connections within the circuits, corresponding to one of the four clock frequencies, are dependent solely on the setting of bits 2 and 3 of SR58.

A.4.2. The Control Circuits

The action performed by the control circuitry is determined completely by bits 4, 5 and 7 (weights 256, 128 and 32) of the output side of SR58. In the quiescent state (all bits in SR58 equal to zero) no action is performed by the control circuits and the tape recorder is under complete manual control.

For analog input bit 5 of SR58 is set to a one. This gates the clock pulses into the A/D section of the circuitry and closes the START relay so as to hold the tape recorder in the playback mode. At the beginning of each sample cycle the A/D clock pulse clears the "End of Conversion" (CNDN)

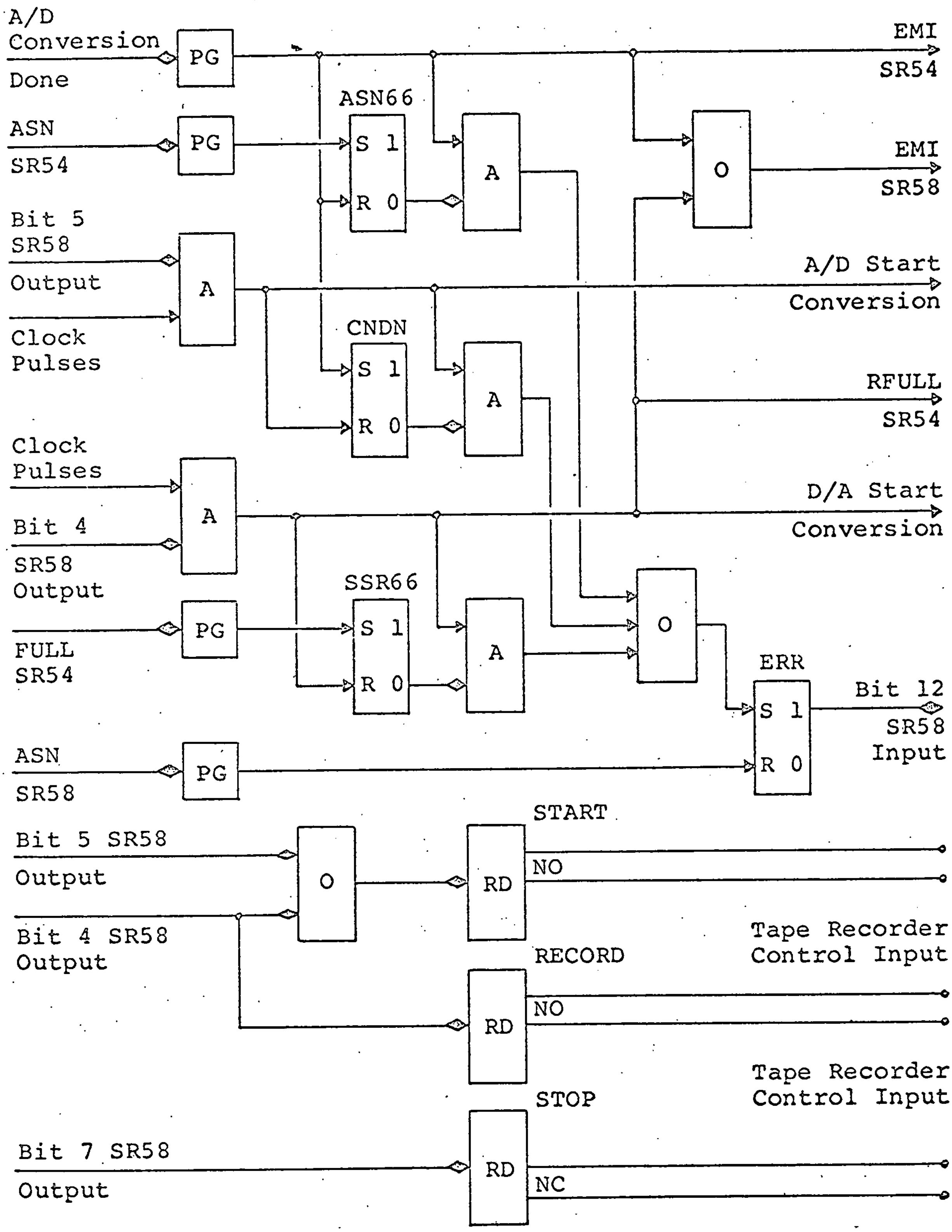


Figure 7. Control Circuits

flip-flop and signals the A/D converter to begin conversion. If (CNDN) was not set to 1 during the previous sample cycle the "Error" (ERR) flip-flop will be set to 1 at this time. When "Conversion Done" occurs, CNDN is set to 1, and a pulse is sent to the EMI inputs of SR54 and SR58. This causes the outputs of the A/D converter and the flip-flop ERR to be gated into the input registers. After the contents of SR54 are read by a program in the Illiac II, the ASN signal for SR54 is sent to the interface which, in turn, sets to 1 the "ASN Done" (ASN66) flip-flop. If this flip-flop is not set by the time the next "Conversion Done" pulse occurs ERR will be set to 1. The ASN66 flip-flop is reset by the "Conversion Done" pulse, while ERR is reset by the ASN signal from SR58.

For analog output bit 4 of SR58 is set to 1. This gates the clock pulses into the D/A section of the control circuitry and closes the START and RECORD relays so as to hold the tape recorder in the record mode. At the beginning of each sample cycle the D/A clock pulse is sent to the D/A converter causing it to gate the contents of the output side of SR54 into its own internal register thus producing a new analog value. The D/A clock pulse is also sent to the RFULL input of SR54 and to the EMI input of SR58, enabling the program in the Illiac II to store a new sample in SR54, and causing the flip-flop ERR to be gated into the input side of SR58, respectively. When the output side of SR54 has been reset by the program, the signal FULL is sent to the interface which in turn sets to 1 the "SSR Done" (SSR66) flip-flop. If SSR66 is not set to 1 at the beginning of each D/A sample cycle, ERR is set to 1. SSR66 is reset by the D/A clock pulse and ERR is reset by the ASN signal from SR58.[9]

Analog input or output is terminated by resetting bits 4 and 5 of SR58 to zero. When this is done, the tape recorder is returned to manual control but is left running in the state previously selected (playback or record). To stop the recorder, bit 7 of SR58 is set to 1 for about 0.2 seconds, and then reset to 0 thereby momentarily activating the STOP relay. If bit 7 remains set to 1, the recorder is held in the stop position by the STOP relay and cannot be manually controlled.

[9]Note that while ERR is reset when SR58 is read, the input side of SR58 is reset only when ERR is gated into it (i.e., at the next occurrence of "Conversion Done" during input or the D/A clock pulse during output).

APPENDIX B. THE ANALOG I/O INTERFACE: MACHINE LANGUAGE PROGRAM CONTROL

The reader should be familiar with Section II.2 of this thesis and the Illiac II symbolic machine language, NICAP[10], before attempting to read the following information.

B.1. Analog Input

Input conversion of an analog signal recorded on audio tape is initiated by the following sequence of instructions:

(1) ATN 128+F

(2) SSR 58

(3) (Delay loop of 1 second)

(1) and (2) set the output side of SR58 to the value 128+F where F is a previously defined constant. The 128 (bit 5) selects analog input and starts the tape recorder in the playback mode. The bits defined by F select one of the four sampling rates. The delay loop (3) allows the tape recorder to stabilize but is not absolutely required. The allowed values for F are:

0	for	40,000	samples	per	second
1024	"	30,000	"	"	"
512	"	20,000	"	"	"
1536	"	10,000	"	"	"

At the end of each sample conversion, bit 2 (weight 1024)

[10] Gear, C.W., Illiac II Manual, University of Illinois, DCS, Urbana, Illinois.

of SR14 is set to 1. This indicates that the sample is ready to be read. If the corresponding mask bit in SR22 is set and "interrupts are enabled", the program will be interrupted at this point.[11] Otherwise, this bit must be tested by a program loop. Experimentation has shown the shortest test-and-read loop to be the following:

```
      FLD
(1)   ASN   14
(2)   CAM   N,-1025
(3)   CJF   N
(4)   ASN   54
(5)   CAM   M
(6)   CNN
(7)   SSR   14
```

(1) and (2) store in modifier N the contents of SR14 minus 1025. If bit 2 is not on, modifier N will contain -1025 and (3) will transfer control back to (1). However, when bit 2 is set, modifier N will contain -1 allowing control to pass onto (4). (4) and (5) store into modifier M the sample value while (6) and (7) reset SR14 to zero.

Analog input is terminated by the sequence

```
(1)   ATN   32
(2)   SSR   58
(3)   (Delay loop ≈ 0.2 sec)
```

---

[11] Gillies, D.B., Special Registers and Interrupts for the Illiac II, University of Illinois, DCS, Urbana, Illinois.

(4) SSR 58

(1) and (2) reset the output side of SR58 to 32 (bit 7 = 1) which terminates analog I/O and stops the tape recorder. (3) is required to allow time for the relays in the tape recorder to function while (4) resets the output side of SR58 to zero allowing manual control of the recorder.

B.2. Analog Output

Analog output to the audio tape recorder is initiated by the sequence of instructions

(1) ATN 256+F

(2) SSR 58

(3) (Delay loop of 1 second)

(1) and (2) set the output side of SR58 to the bit configuration 256+F. The 256 (bit 4) selects analog output and starts the tape recorder in the record mode. The bits defined by F select one of the four sampling rates and its associated filter. The delay loop (3) is provided to allow stabilization of the recorder but, again, it is not required. The allowed values for F are:

0 for 40,000 samples per second with a 20 KHz low pass filter

1024 for 30,000 samples per second with a 15 KHz low pass filter

512 for 20,000 samples per second with a 8.5 KHz low pass filter

1536 for 10,000 samples per second with a 4 KHz low pass filter

Each output sample is transmitted by the sequence

(1) ATN M

(2) SSR 54

(1) and (2) store the number M or the contents of modifier M into SR54. Since the program is prevented from executing the SSR 54 instruction until a new sample is requested by the interface, no other signal need be tested. It is only necessary that the program be capable of executing this sequence at a rate exceeding the sample conversion rate. The Illiac II will then stop the program before each execution until the interface is ready to receive a new value.

Analog output is terminated by the same sequence required by analog input. It should be noted that when either bit 4, 5, or 7 (weights 256, 128, and 32 respectively) is set to 1 in the output side of SR58, the recorder cannot be operated manually. Therefore, it is necessary to make sure SR58 is set to 0 after analog I/O is terminated.

B.3. Error Checking

The error flag, set as described in Section II.1, is read by the sequence:

(1) ASN 58

(2) CAM M

(1) and (2) store in modifier M the input side of SR58 which has the value 0, if no errors have been detected, or

1, if one or more errors have been detected. Execution of (1) also causes the error flag to be reset to 0 immediately and the input side of SR58 to be reset at the beginning of the next sample period. Note that if (2) is replaced by the instruction ADM M, modifier M becomes an error counter.

B.4. Floating Point Conversion

The digitized samples may be easily converted to or from normalized floating point by the following sequences:

(1) ATN Mx

(2) CAD 10,3,0

(3) STF FY

Sequence (1), (2) stores the sample in modifier Mx into the accumulator as a signed floating point number with a magnitude $\leq$ 0.9997558 (the value of $\sum_{i=1}^{12} 2^{-i}$). The instruction (3) shifts the accumulator and stores it in fast register FY so that the first modifier of FY (M0, M4, M8, or M12) contains those bits of the original number with weights $2^{0}, 2^{-1}, \ldots, 2^{-12}$. If the accumulator had a magnitude of $\leq$ 0.9997558 (as produced by the sequence (1), (2)) the result in the first modifier would be the equivalent sample value. Accumulator values with magnitudes $\geq$ 1.0 will cause overflow to be set when (3) is executed. Thus sample values in the range -4095 to +4095 may be mapped directly into the normalized floating point indicated above and conversely.

Information regarding the programming to tape, disk and drum I/O is given in bibliography references 2, 4, and 8.

APPENDIX C. ADSYS2 TAPE FORMAT AND COMMAND LANGUAGE DESCRIPTIONS

The reader should be familiar with Section II.3 of this thesis before attempting to read the following information.

C.1. File Tape Format

All records written on the user's file tape must be 256 Illiac II words in length. The first record on the tape is the tape label record, and the first word of the tape label record contains the 6 character BCD name "ANALOG" in quarter words 0, 1, and 2 as a means of identifying the tape as a proper file tape. Quarter word 3 contains a binary "tape number" that may be used to uniquely identify the tape. The only restriction on the tape number is that it should lie in the range 0 to 4095. (Note that the tape number in no way refers to the logical number of the tape unit on which it is mounted for processing.) The remaining 255 words of the tape label record, as well as the entire contents of any record following up to the first tape mark (end-of-file), may be specified by the user and used for further labeling information if desired.

Following the tape label record (and any additional records), are an unspecified number of data files. At the beginning of each data file there is a tape mark followed by a file label record or another tape mark. The first tape mark serves to separate the data files while the second tape mark, in place of a file label record, indicates that there are no more data files beyond this point.

The first word of each file label record serves to uniquely identify the file. Quarter word 0 contains a binary number while quarter words 1, 2 and 3 are required to be zero. The binary number is used to define the file and may have any value from 0 to 4095. The files need not be numbered in increasing numeric order; however, no two files on the same tape should have the same file number.

Following the file label record are an unspecified number of data records which constitute the file. A file is terminated by the tape mark at the beginning of the next sequential file. Samples in the data records are stored one per quarter word, 1024 per record. Sequencing of the samples in time is from quarter word 0 to 3, word 0 to 255 with the records sequenced in the order that they are read or written.

C.2. Conversion Tape Format

The conversion tape is copied from or onto the file tape using multiple block records and is used for the actual analog I/O. All records on the conversion tape are 1024 Illiac II words long. The first record is preceded by a tape mark (written at the beginning of the tape) and the last record is followed by two tape marks. There are no labels to be checked, and thus all the records are data records. Sequencing of the samples in time is from quarter word 0 to 3, word 0 to 1023 with the records sequenced in the order they are read or written. (Note: The conversion tape is to be considered a scratch tape and should not be used to save data between runs. Furthermore, processing of the multiple block records may not be performed during normal batch processing.)

C.3. The ADSYS2 Command Language

ADSYS2 begins execution by printing on the console teletype

ADSYS2 LOADED

and then proceeds to rewind and initialize the conversion tape on channel 5, unit 0. When this has been accomplished, ADSYS2 begins job processing until a request is made by the operator to terminate execution. Each job is initiated by ADSYS2 printing the message

SELECT JOB

followed by two bell rings indicating the operator is to type in a command. Information typed in at any other time is ignored. In describing the commands symbolic parameters will be used to represent the actual numeric values that are input. In certain cases these parameters may be eliminated by deleting them from the end of the list causing ADSYS2 to use a default value instead. The parameters and their default values are as follows:

ln This is the logical number of the tape unit containing the user's file tape. This unit may not be connected to channel 5, and is therefore limited to the values 6, 7, 8, or 11. This parameter may not be deleted if used by a command.

tn This is the tape number of a specified file tape. ($0 \leq tn \leq 4095$) If tn is typed in as a non-zero number, the tape specified will be checked for, or written with, a tape label record containing this number. If tn is deleted or equal to zero the tape number will not be checked or a zero tape number will be written. Note that in all cases where checking is performed the first word of the tape label record must contain the BCD name "ANALOG" regardless of whether or not the tape number is verified.

fn This is the file number of a specified file. ($0 \leq fn \leq 4095$) If fn is deleted, fn = 0 will be assumed.

r This is the sampling rate desired for conversion. The allowed values are 1, 2, 3, or

4 for 10KHz, 20KHz, 30KHz or 40KHz respectively. If r is deleted or zero, r = 4 will be assumed. Note that ADSYS2 will attempt to process any of the four possible sample rates. However, for rates exceeding 35KHz, the run will be terminated immediately after starting due to the insufficient data I/O rate capability of the digital tape units.

t This is the maximum time limit for conversion in seconds. ($0 \le t \le 999$) If t is deleted or set to zero, no time limit is imposed.

f This is the frequency in hertzians of a tone to be generated internally by ADSYS2. ($100 \le f \le 4000$) If f is deleted or set to zero f = 1000Hz will be assumed.

The commands which may be input to ADSYS2 are defined as follows:

(1) COMMAND: LABEL ln tn

ACTION: Rewind the tape on logical tape unit ln and write a tape label record with the tape number tn followed by two tape marks. All previous information on the tape will be effectively destroyed. When the job is completed, reply with the message "JOB DONE".

(2) COMMAND: LOAD ln fn tn

ACTION: Rewind the file tape on logical tape unit ln and check the tape label record for validity. Then search the tape until the file numbered fn is found. Copy the data records from the file tape onto the conversion tape four at a time until a tape mark is read, then rewind both tapes. The previous contents of the conversion tape are destroyed. When the job is completed, reply with the message "JOB DONE". (Note that if the number of data records on the file tape are not a multiple of 4, the last 1, 2, or 3 records before the terminating tape mark will not be copied onto the conversion tape).

(3) COMMAND: SAVE ln fn tn

ACTION: Rewind the file tape on logical tape unit ln and check the tape label record for validity. Then search the tape until the file numbered fn is found. If file fn is not found, create a new file labeled fn at the end of the last file on the tape.

Copy the data records from the conversion tape onto the file tape until a tape mark is read. Then, rewind the conversion tape, write two tape marks on the file tape, and rewind the file tape. The previous contents of the specified file and all following files will be destroyed. When the job is completed, reply with the message "JOB DONE".

(4) COMMAND: A/D r t

ACTION: Rewind and write a tape mark at the beginning of the conversion tape; then reply with the message "READY". The previous contents of the conversion tape will be destroyed. When the operator turns bit 0 of SR28 from OFF to ON, begin analog input at the sampling rate r for the time t. After each data record is written on the conversion tape, check for a possible termination condition. If one is found stop analog input, write two tape marks on the conversion tape and rewind it, then reply with the message "XXX END YYY ZZZ". XXX is the type of termination and may be one of the following:

OPR   if the operator sets bit 0 of SR28 to the OFF position.

TIME   if the input time exceeds that specified by t.

EOT   if the physical end of the conversion tape is sensed.

ERR   if the tape data transfer rate is below the specified sampling rate.

YYY is the actual time expended in the conversion in seconds (if >999, YYY will equal 999) and ZZZ is the number of records written in which one or more conversion errors occurred. ZZZ is not printed if it is zero.

(5) COMMAND: D/A r t

ACTION: Rewind and skip the tape mark at the beginning of the conversion tape. Then reply with the message "READY". When the operator turns bit 0 of SR28 from OFF to ON, begin analog output at the sampling rate r for the time t. After each record is read from the conversion tape, check for a possible termination condition. If one is found, stop analog output, rewind the conversion tape, and then reply with the message "XXX END YYY ZZZ". This code is identical to that which occurs for A/D except that EOT is replaced by EOF if a tape mark indicating the end of the input file is sensed.

(6) COMMAND: SINE f t

ACTION: Prepare to generate a sinusoidal full-scale analog output of frequency f at a sample rate 40,000 Hz; then reply with the message "READY". When the operator turns bit 0 of SR28 from OFF to ON, begin analog output of the required tone. At the end of each cycle, check for a possible termination condition. If one is found, stop analog output and then reply with the message "XXX END YYY". This code is identical to that which occurs for A/D except that XXX is restricted to either OPR or TIME, and no checking is done for conversion errors.

(7) COMMAND: RAMP f t

ACTION: Identical to SINE except a sawtooth tone is produced.

(8) COMMAND: END

ACTION: Output the message "BATCH REENTRY" and then terminate ADSYS2 by returning to the batch processing stream.

C.4. Error Messages

The following error messages may be output by ADSYS2 in response to a job request. In all cases the job will not be run.

(1) MESSAGE: INVALID PARAMETERS

MEANING: The job name or one of the input parameters is illegal.

(2) MESSAGE: TAPE ERRORS

MEANING: Repeated tape errors were detected during tape or file label processing. (This could indicate an improper file tape).

(3) MESSAGE: BAD TAPE LABEL

MEANING: The tape label record did not contain the BCD name "ANALOG" in the first word or the label record was not there (i.e., a tape mark was written instead).

(4) MESSAGE: WRONG TAPE NUMBER

MEANING: Tape label number checking was requested and the indicated tape did not contain the given tape number.

(5) MESSAGE: BAD FILE LABEL

MEANING: The last three quarter words of the first word of a file label were not all zero.

(6) MESSAGE: FILE NOT FOUND

MEANING: The file requested to be loaded onto the conversion tape was not found on the file tape.

C.5. Other Operating Information

During analog input the first 1 second of analog tape is bypassed without being input to allow for speed stabilization. When input is terminated another 1 second is bypassed before the recorder is stopped. The same procedure is followed during analog output except that the bypassed tape is erased.

For the commands which wait until the operator sets bit 0 of SR28 to begin, the operator may instead turn on bit 1 causing the job to be deleted and the message "JOB NOT RUN" to be printed.

If an error is made in typing a command, typing the character % will delete the line and send out two bell rings indicating a new command is requested.

The total time charged is the sum of the times given by the messages "XXX END YYY ZZZ" plus 3 minutes for overhead.

APPENDIX D. ADPAK2 SUBROUTINE CALLING SEQUENCES

The reader should be familiar with section II.3 of this thesis and with the use of FORTRAN and/or NICAP in the Illiac II batch processing system before attempting to read the following information.

D.1. Program Requirements

ADPAK2 is 472 locations long and uses the first 256 words of common storage as a tape I/O buffer. The common locations may be used by the program between calls to ADPAK2 subroutines. Any call to an ADPAK2 subroutine destroys the contents of the accumulator and fast registers 0 and 1. Fast registers 2 through 7 are left unchanged.

The logical tape units used may be on either channel 4 or 5 but must be units that are normally allowed to the user. Uncorrectable tape errors occurring during reading or writing of data records will not cause program termination.

D.2. FORTRAN Calling Sequences

(1) STATEMENT: CALL ADLBL (I,J)

ARGUMENTS: Input - I,J; output - none

ACTION: Rewind logical tape I and write a tape label record with the tape number J followed by two tape marks. All previous data files on tape I are effectively destroyed.

(2) STATEMENT: CALL ADBGI (I,N,J)

ARGUMENTS: Input - I,J,N; output - none

ACTION: Rewind logical tape I and check the tape label record for validity. If $J \neq 0$, check to see that the tape number is the same as J; otherwise, do not check the tape number. After the tape label is checked, search the tape for a file numbered N and position the tape to read the first data record from the file (the input file is now active;) then, return to the calling program.

(3) STATEMENT: CALL ADBLI (A,K)

ARGUMENTS: Input - none; Output - A(1024),K

ACTION: If the input file is not active, set K = 0 and return immediately to the calling program. Otherwise, read the next data record from the input file. If this is not a tape mark, convert the 1024 samples to floating point and store them in the array A from A(1) to A(1024). Set K = -1 or 1 if uncorrectable tape errors were or were not encountered, respectively; then, return to the calling program. If the data record is a tape mark rewind the input file tape, (the input file is now inactive) set K = 0, and then return to the calling program.

(4) STATEMENT: CALL ADSKI (M,K)

ARGUMENTS: Input - M; Output - K

ACTION: If the input file is not active set K = 0 and return immediately to the calling program. Otherwise, skip over the next M records in the input file. If no tape mark is detected, set K = -1 or 1 if uncorrectable tape errors were or were not encountered, respectively; then, return to the calling program. If a tape mark is read, rewind the input file tape, (the input file is now inactive) set K = 0, and then return to the calling program.

(5) STATEMENT: CALL ADEDI

ARGUMENTS: None

ACTION: If the input file is not active, return immediately to the calling program. Otherwise, rewind the input file tape, (the input file is now inactive) then return to the calling program.

(6) STATEMENT: CALL ADBGO (I,N,J)

ARGUMENTS: Input - I,J,N; Output - None

ACTION: Rewind logical tape I and check the tape label record as for ADBGI, then search the tape for a file numbered N. If the file is found, position the tape to write the first data record of the file. Otherwise, write a new file label record after the last file on the tape; (in either case the output file is now active) then, return to the calling program.

(7) STATEMENT: CALL ADBLO (A,K)

ARGUMENTS: Input - A(1024); Output - K

ACTION: If the output file is not active, set K = 0 and return immediately to the calling program. Otherwise, convert the 1024 floating point samples in the array A from A(1) to A(1024) into a data record and write this onto the output file as the next sequential data record. If the end-of-tape indicator is not sensed, set K = -1 or 1 if uncorrectable tape errors were or were not encountered, respectively; then, return to the calling program. If the end-of-tape indicator is sensed, write two tape marks and rewind the output file tape. (The output file is now inactive.) Set K = 0, and then return to the calling program.

(8) STATEMENT: CALL ADSKO (M,K)

ARGUMENTS: Input - M; Output - K

ACTION: If the output file is not active, set K = 0 and return immediately to the calling program. Otherwise, write M records of all zeros onto the output file. If the end-of-tape mark is not detected set K = -1 or 1 if uncorrectable tape errors were or were not encountered, respectively; then, return to the calling program. If the end-of-tape indicator is sensed, write two tape marks and rewind the output file tape. (The output file is now inactive.) Set K = 0, and then return to the calling program.

(9) STATEMENT: CALL ADEDO

ARGUMENTS: None

ACTION: If the output file is not active, return immediately to the calling program. Otherwise, write two tape marks and rewind the output file tape; (the output file is now inactive) then, return to the calling program.

D.3. NICAP Calling Sequences

NICAP programs may call any of the FORTRAN callable subroutines as follows:

```
CALL   NAME
DECQ   W,X,Y,Z
```

where name is the subroutine name, and W,X,Y, and Z are the addresses of the input parameters required. For example, to call ADBGI in NICAP, the sequence

```
CALL   ADBGI
DECQ   I,N,J,0
```

is used where I,N, and J are the addresses of the logical tape number, file number, and tape number, respectively.

In addition, there are four other subroutines available only to NICAP programs, The calling sequences for these are:

```
CAM    0,A
CALL   ADBI1    (or ADBO1)

CAM    0,M
CALL   ADSI1    (or ADSO1)
```

ADBI1, ADBO1, ADSI1 and ADSO1 are the same as ADBLI, ADBLO, ADSKI, and ADSKO, respectively, where A is the address of the first location of the array and M is the number of records to skip. On return modifier zero equals the value (not the address) of K.

D.4. Error Exits

If an error is detected by an ADPAK2 subroutine, a message will be printed on the user's output, and SYSERR will be called. The form of the message is:

*** ADPAK2 ERROR - XXX ***

where XXX is one of the following:

(1) MESSAGE: ADLBL or ADBGO ATTEMPTED ACCESS OF INPUT TAPE

MEANING: A call to ADLBL or ADBGO specified a logical tape unit which contains a currently active input file tape.

(2) MESSAGE: ADLBL or ADBGI ATTEMPTED ACCESS OF OUTPUT TAPE

MEANING: A call to ADLBL or ADBGI specified a logical tape unit which contains a currently active output file tape.

(3) MESSAGE: TAPE ERRORS OR END-OF-TAPE DETECTED DURING LABEL WRITING

or

TAPE ERRORS DETECTED DURING LABEL READING

MEANING: Uncorrectable tape errors or an end-of-tape indication occurred during tape or file label record processing.

(4) MESSAGE: INCORRECT TAPE LABEL

MEANING: The tape label record of a file tape did not have the BCD name "ANALOG" in the first three quarter words of the first word.

(5) MESSAGE: INCORRECT TAPE NUMBER

MEANING: The tape number in the tape label record did not agree with that specified in a call to ADBGI or ADBGO.

(6) MESSAGE: INCORRECT FILE LABEL

MEANING: The last three quarter words of the first word of a file label record were not all zero.

(7) MESSAGE: REQUESTED FILE NOT FOUND

MEANING: The file number given in a call to ADBGI was not found on the specified input file tape.

In addition to the above, accumulator overflow or specification of an illegal logical tape unit number may cause the system to terminate the job. Accumulator overflow will occur if any of the parameters I,N,J, or M (specified in D.2 and D.3) exceeds 4095, or if any sample in the array A has an absolute value that is equal to or greater than 1.0.

BIBLIOGRAPHY

1. Gear, C.W., Illiac II Manual; Use of the New Illinois Computer Installation, University of Illinois, Department of Computer Science, Urbana, Illinois, March 1963.

2. Gillies, D.B., Levin, M.S., Special Registers and Interrupts for the Illiac II, File No. 517, University of Illinois, Department of Computer Science, Urbana, Illinois, December 15, 1964.

3. Otis, A.B., Jr., The Analog-Digital Programming System, File No. 726, University of Illinois, Department of Computer Science, Urbana, Illinois, February 23, 1967.

4. Pisterzi, M.J., Programming Aspects of the Magnetic Tape Interplay Channels, File No. 588, University of Illinois, Department of Computer Science, Urbana, Illinois, March 16, 1964.

5. Schroeder, M.R., Natural Sounding Artificial Reverbertion, Monograph 4309, Bell Telephone Laboratories, Inc., New York, N.Y., November 1962.

6. Stephenson, B.W., Analog-Digital Conversion Handbook, Digital Equipment Corp., Maynard, Mass., 1964.

7. Susskind, A.K., Notes on Analog-Digital Conversion Techniques, The MIT Press, Cambridge, Mass., 1957.

8. Willard, R.E., Nuspl, S.J., Programming Aspects of the Disk File Channel, File No. 620, University of Illinois, Department of Computer Science, Urbana, Illinois, August 2, 1965.