CONFIDENTIAL

TITLE: HIGH-SPEED MULTIPLE-CHANNEL ANALOG TO
DIGITAL DATA ACQUISITION MODULE FOR
MICROPROCESSOR SYSTEMS

AUTHOR(S): C. DWAYNE ETHRIDGE

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration nor any of their employees, nor any of their contractors, subcontractors, or their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

SUBMITTED TO: L. Costrell
Center for Radiation Research
National Bureau of Standards
Washington, D.C. 20234

1976 Nuclear Science Symposium
and Scintillation and Semiconductor
Counter Symposium. The Institute
of Electrical and Electronic
Engineers; Nuclear and Plasma Science
Society, New Orleans, Louisiana,
October 20-22, 1976.

By acceptance of this article for publication, the publisher recognizes the Government's (license) rights in any copyright and the Government and its authorized representatives have unrestricted right to reproduce in whole or in part said article under any copyright secured by the publisher.

The Los Alamos Scientific Laboratory requests that the publisher identify this article as work performed under the auspices of the USERDA.

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

An Affirmative Action/Equal Opportunity Employer

ABSTRACT

HIGH-SPEED MULTIPLE-CHANNEL ANALOG
TO DIGITAL DATA-ACQUISITION MODULE
FOR MICROPROCESSOR SYSTEMS

C. Dwayne Ethridge,
Los Alamos Scientific Laboratory
of the University of California

Intelligent data acquisition and instrumentation systems established by the incorporation of microprocessor technology require high-speed analog to digital conversion of multiple-channel input signals. Sophisticated data systems or sub systems are enabled by the microprocessor software flexibility to establish adaptive input data procedures. These adaptive procedures are enhanced by versatile interface circuitry which is software controlled.

ADDRESS:

C. Dwayne Ethridge
University of California
Los Alamos Scientific Laboratory
P. O. Box 1663 MS 447
Los Alamos, New Mexico 87545

TELEPHONE:

Area Code 505 667-7161
FTS Number 643-7161

MASTER

HIGH-SPEED MULTIPLE-CHANNEL ANALOG
TO DIGITAL DATA-ACQUISITION MODULE
FOR MICROPROCESSOR SYSTEMS*

by

C. Dwayne Ethridge

University of California
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87545

I. INTRODUCTION

Intelligent data acquisition and instrumentation systems established by incorporation of microprocessor technology require analog to digital conversion of multiple-channel analog-input signals. Sophisticated data systems or subsystems are enabled by the microprocessor software flexibility to establish adaptive input data procedures. These adaptive procedures are enhanced by versatile interface circuitry which is software controlled.

This discussion involves a twelve-bit analog to digital conversion design to establish an eight-bit microprocessor interface for 16 single-ended channels (8 differential-ended) operating at rate of 100KHz. The design is expandable to 256 single-ended channels or 128 differential channels.

Data acquisition systems for research and development projects require flexibilities to incorporate adaptive features as the project progresses. The basic function of the specific microprocessor data acquisition system for which this Analog to Digital Converter (ADC) was designed is to examine, optimize, and provide accurate repeatable setup of the parameters of an Electron Beam Welder.

* Work performed under the auspices of the USERDA.

The microprocessor unit MPU controlling this system is the Intel 8080A, an 8-bit parallel central processing unit. The MPU is fabricated on a single LSI chip utilizing n-channel silicon MOS technology. The instruction times vary from 2 μ s to 9 μ s when the MPU is supplied a 2MHz two-phase clock.

II. ANALOG MULTIPLEXER AND CONVERSION MODULE

The design incorporates the Analogic^{*} MP6912 Subminiature High-Speed Data-Acquisition System Module. The module contains a complete 12-bit, high-speed, successive-approximation analog to digital converter, a high-speed sample and hold, a precision reference, a fast high-stability buffer amplifier, digital-output buffer amplifiers, and a 16-channel multiplexer for both single-ended and differential inputs. Typical data-acquisition system module characteristics are listed in Figure 1. Other pin-compatible modules available are Analogic 6812, Burr-Brown SDM850 and SDM851, Datel MDAS-16, Data Translation DT5710, and Micro Network MN7000.

III. MAPPED MEMORY ADDRESSING

The ADC subsystem is designed for mapped memory operation as opposed to I/O device operation. Mapped memory operation treats the microprocessor outputs for ADC initialization commands as instructions writing data into specific memory locations. Input instructions of ADC data are likewise handled as memory read instructions. Figures 2 and 3 designate the hexadecimal memory address assignments for writing ADC initialization commands and reading ADC data.

Herein all 16-bit addresses and 8-bit data specific values are defined with four characters and two characters of hexadecimal nomenclature respectively. Hexadecimal nomenclature defines the sixteen characters required to describe the sixteen possible states of 4 binary bits. Ten of the character set are

^{*}Reference to a company or product name here or elsewhere in this report does not imply approval or recommendation of the product by the University of California or the U. S. Energy Research and Development Administration to the exclusion of others that may be suitable.

identical to the Binary-Coded-Decimal characters 0 through 9. The remaining 6 hexadecimal states, illegal BCD characters, are assigned the characters A through F.

The ADC occupies eight contiguous memory locations. Basic ADC operation requires five write locations (ADC2 through ADC6) and three read memory locations (ADC0 through ADC2.) One additional write location (ADC7) is required for the ADC subsystem expanded from 16, single-ended analog input channels to 17 through 256, single-ended analog input channels.

IV. ADDRESS BUS DECODING

The ADC Subsystem functional block diagram of Figure 4 inputs the 16 address bus lines to the address decoding block. For this low volume application design, specific memory address locations were allocated and fully decoded to minimize software training and programming. Redundancy in addressing created by typically small programs can be used to reduce the address decoding function. Minimized production systems may take advantage of software programs less than 32K for example by using the upper address bit to enable the ADC. Decoding of the minimum number of address bits as a function of the number of addresses needed can minimize the address decoding function.

V. PROGRAMMABLE PERIPHERAL INTERFACE I/O PORT

The address decoding block generates the Port Select signal which addresses the Intel Silicon Gate MOS 8255 Programmable Peripheral Interface I/O Device in conjunction with address lines A_0 and A_1 . The 8255 is a general purpose programmable I/O device which has 24 I/O pins. The 24 pins may be individually programmed in two groups of twelve pins and used in three major modes of operation.

In the first mode (Mode 0), each group of twelve I/O pins may be programmed in sets of 4 to be input or output. In Mode 1, the second mode,

each group may be programmed to have 8 lines of input or output. Of the remaining four pins, three are used for handshaking and interrupt control signals. The third mode of operation (Mode 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking. The I/O device also includes bit set and reset capability.

The ADC is designed for I/O device operation in Mode 1. I/O Port pin assignments are listed in Figure 5. The eight bits of Port A are programmed as input of the ADC module bit 11 (MSB) through bit 4.

Port B is also programmed as input with the ADC module input of bit 3 through bit 0 (LSB.) The remaining four bits of Port B input one of three values as a function of program control bit set and reset capability of bits 6 and 7 of Port C. The software commands inputs of zero, ADC module channel address (1 of 8 or 1 of 16) or expander channel address (also 1 of 8 or 1 of 16.)

Port C is also programmed to manage the ADC module end of conversion signal to a generate program interrupt or input buffer full data for programmed I/O software. When a program interrupt system is incorporated, the software must enable the interrupt utilizing the I/O device bit set and reset capability. The software program may monitor the condition of Port C by reading contents of the eight bits.

Interrupt level is selected initially and implemented with a jumper selector. The interrupt level may be any one of the available eight levels as a function of overall program organization and requirements.

The I/O Port inputs are read into the MPU by reading hexadecimal addresses ADC0, ADC1, and/or ADC2. Figure 2 defines the specific address assignments.

MPU software control of the I/O Port is accomplished by writing into hexadecimal addresses ADC2 and ADC3. Figure 3 defines the specific address assignments.

VI. ANALOG INPUT CHANNEL SELECTION AND MULTIPLEXING

The specific analog channel to be converted to digital form is selected by the eight-bit data word written into memory address ADC7. The data word is controlled by two four-bit parallel bidirectional bus drivers as shown in Figure 4. The direction of data flow is managed by the 8080A Data Bus Input signal DBIN. DBIN active routes data from the ADC onto the MPU data bus. Otherwise, data is gated onto the ADC. The bus drivers are tri-state devices and are enabled by the 8080A Hold signal HLDA applied to the chip select inputs.

The eight-bit data word specifies 1 of 256 channels. For the expanded input channel configurations, the eight-bit word is separated into two significant four-bit bytes. The upper four-bit byte is loaded into the Analog Multiplexer and Conversion Module. (Reference Section II and Figure 4.) The lower four-bit byte is loaded into the expander circuitry to select the 1 of 16 expansion channels of the expansion modules.

A system of only 16 channels, for example, would write an eight-bit data word with the upper four-bit byte containing don't care information. The lower four-bit byte containing the channel selection data is then loaded into the Analog Multiplexer and Conversion Module. The function block diagram, Figure 4, illustrates the jumper selection of the four-bit byte channel address by control of the QUAD 2 to 1 multiplexer 74LS157.

I/O Port bits C₇ and C₆ are controlled by the software program to return the actual expander address or the ADC channel address. Figure 4 shows

bit C_7 STROBE and C_6 SELECT of the Quad 2 to 1 multiplexer 74LS157.

VII. CONVERSION DESCRIPTION AND SOFTWARE DESCRIPTION

The functional block diagram of the MPU multiple-channel, twelve-bit ADC is illustrated in Figure 4. This figure is the basis for discussion related to Conversion Description and Software Description. Two methods of conversion are available to software control. One method is random access of any analog input channel. The second method is sequential access of the channels. The end of conversion may be recognized in either method by programmed I/O or an interrupt system.

In general, the following operational procedure must be followed.

Port A mode selection control data is written into memory address ADC3.

Port B mode selection control data must also be written into memory address ADC3. Port C bit 7 must be set up as well as Port C bit 6.

In method one, random access of any analog input channel, the channel selection data word is then written into memory address ADC4. When an interrupt system is used, the software must then enable the ADC interrupt before continuing with the main program. When the ADC interrupt occurs, the ADC interrupt service routine must then read the ADC output by reading address ADC0 and ADC1 for the upper and lower eight-bit data bytes respectively.

An example of random accessing of input channels is listed in Figure 6. Port setup is accomplished by instructions P through P + 9. The input channel multiplexer and conversion initiation is accomplished by instruction C through C + 8. The readout of the ADC conversion is then accomplished by instructions R through R + 7.

The second method of conversion, sequential access of the channels is slightly more complex. Port setup is accomplished by instructions P through P + B. The first channel may be selected by writing anything into memory

address ADC5. This selection and conversion initiation is accomplished by instructions C through C + 9. Advancing the channel number and reinitiation of the conversion is accomplished by instructions A through A + A after reading the previous conversions with instructions R through R + 7.

These two examples are lengthy and redundant in some aspects with a goal of clarifying the basic ADC operation. Since the ADC addressing is mapped memory, the software programmer is restricted only by the MPU instruction set. The actual number of instructions required for a specific application will vary as a function of the individual programmer and peculiar requirements of the application.

VIII. INPUT/OUTPUT PARAMETER VARIATION CAPABILITY

The ADC design includes a flexibility to enable variation of input parameters without hardware redesign. This capability is established through a series of jumper selections. These selections allow redefinition of input parameters as projects develop definitive requirements.

Input configurations may be selected from the following three types:

1) 16 single-ended inputs, 2) 8 differential inputs, or 3) 16 pseudo-differential inputs. The three types are also available with the expansion capability.

The input voltage full scale range may be selected from the following eight ranges: 1) 0 to +10V, 2) 0 to + 10.24V, 3) 0 to + 5V, 4) 0 to + 5.12V, 5) - 10V to + 10V, 6) - 10.24V to + 10.24V, 7) - 5V to + 5V, or 8) - 5.12V to + 5.12V. All channels will have the same input voltage full scale range.

The conversion time, or throughput rate, can be trimmed to optimize the accuracy/throughput rate tradeoff. This trimming is enabled by jumper addition of on board potentiometers. One potentiometer increases or decreases the time allowance for multiplexer and sample-and-hold settling. The second potentiometer adjusts the A/D clock rate to increase or decrease the bit conversion time tradeoff. Refer to Figure 1 for the listings of adjustable ranges.

The Analog Multiplexer and Conversion Module (Section II) provides internal trimming potentiometers to allow calibration of zero offset and full scale range. Additional calibration is achieved by jumper selection of two on board ADC potentiometers.

Four output digital selections are provided by jumper selection. The four codes are as follows: 1) unipolar binary, 2) 2's complement, 3) offset binary, or 4) 1's complement.

The twelve-bit output may be decreased by short cycling the successive approximate A/D conversion to a lower number of bits of converted data. This decrease allows a flexibility to trade off output resolution versus conversion cycle time which is decreased for fewer output bits. This flexibility is provided by jumper selection.

IX. SUMMARY

Intelligent data acquisition and instrumentation systems established by incorporation of microprocessor technology require analog-to-digital conversion of multiple-channel analog-input signals. This high-speed (100KHz) multiple-channel (256) analog-to-digital data-acquisition module designed for eight-bit microprocessor systems fulfills this requirement. A twelve-bit or less analog-to-digital conversion design establishes an eight-bit microprocessor interface.

Sophisticated data systems or subsystems are enabled by microprocessor flexibility to establish adaptive input data procedures. The ADC operation sequence of channel selection, conversion initiation, and end of conversion identification is entirely software controlled. The software program can select random access of all input channels or sequence consecutively through all input channels. The software can check for end of conversion with programmed I/O techniques or enable the interrupt capability.

These adaptive procedures are enhanced by versatile interface circuitry which is software controlled. The incorporation of the programmable peripheral interface device provides software control of the data which the program needs. The circuitry provides capability to read only data, data and multiplexer channel address, or data and expander channel address. Jumper selection ADC input configurations, full-scale ranges, setting time allowance, conversion bit times, and output code formats further enhance the adaptability of the high-speed multiple-channel analog-to-digital data acquisition module for microprocessor systems.

MODEL	CHARACTERISTICS			CHANNEL SETUP TIME			ANALOG-TO-DIGITAL CONVERSION	
	Accuracy % FSR	Thruput Rate KHz	Cycle Time μ s	Input Mux Settling Time μ s	Sample and Hold Acquisition Time μ s	Total Delay Time μ s	NSEC/Bit	12-Bit Conversion Time μ s
6912-								
-30KHz	± 0.025	30	33.3	(NOTE 1)	—	16.7	(NOTE 1)	16.7
-60KHz	± 0.025	60	16.6	(NOTE 1)	—	8.3	(NOTE 1)	8.3
		150	6.6	—	—	MIN=2.8	320 MIN	3.8
-100KHz	± 0.025	100	10	3.5	3.5	5*		5
	± 0.015	60	16.6	—	—	MAX=9	625 MAX	7.6

$$* \quad 3.5^2 + 3.5^2 \quad 5$$

NOTE 1. 6912 - 30 KHz and -60KHz Models do not have External Adjustment of Delay Time or Bit Conversion Time.

FIGURE 1. DATA-ACQUISITION SYSTEM MODULE CHARACTERISTICS

FIG. 2 HEXADECIMAL READ ADDRESS ASSIGNMENTS BYTE ADDRESS XY

X \ Y	0	1	2	3
ADC	PORT A	PORT B	PORT C	ILLEGAL OPERATION

FIG. 3 HEXADECIMAL WRITE ADDRESS ASSIGNMENTS BYTE ADDRESS XY

X \ Y	0	1	2	3	4	5	6	7
ADC	ILLEGAL OPERATION: PORT A IS ASSIGNED TO INPUT DATA ONLY	ILLEGAL OPERATION: PORT C IS ASSIGNED TO INPUT DATA ONLY	PORT C	PORT A,B,C CONTROL	LOAD ANALOG CHANNEL ADDRESS	CLEAR ANALOG CHANNEL ADDRESS	INCREMENT ANALOG MULTIPLEXER CHANNEL ADDRESS	INCREMENT ANALOG EXPANDER CHANNEL ADDRESS

Figure 4. MPU MULTIPLE-CHANNEL 12-BIT ADC BLOCK DIAGRAM

FIGURE 5. I/O PORT PIN ASSIGNMENTS

PORT A	A ₇	ADC BIT 11	(MSB)
	A ₆	10	
	A ₅	9	
	A ₄	8	
	A ₃	7	
	A ₂	6	
	A ₁	5	
	A ₀	4	

PORT B	B ₇	3		
	B ₆	2		
	B ₅	1		
	B ₄	0	(LSB)	
		C ₇ = "1", C ₆ = X	C ₇ = "0", C ₆ = "1"	C ₇ = "0", C ₆ = "0"
	B ₃	"0"	8 } ADC	8 } EXPANDER
	B ₂	"0"	4 } CH	4 } CH
	B ₁	"0"	2 } ADDR	2 } ADDR
	B ₀	"0"	1 }	1 }

PORT C	C ₇	PORT B ₃₋₀	CONTROL (SEE PORT B DESCRIPTION)		
	C ₆				
	C ₅			IBF _A	"1" = PORT A INPUT BUFFER FULL (PORT A DATA LATCHED)
	C ₄ *			STB _A	"0" STROBES DATA INTO PORT A
	C ₃			INTR _A	"1" = PORT A INTERRUPT IF INTE _A HAS BEEN ENABLED
	C ₂ *			STB _B	"0" STROBES DATA INTO PORT B
	C ₁			IBF _B	"1" = PORT B INPUT BUFFER FULL (PORT B DATA LATCHED)
	C ₀			INTR _B	"1" = PORT B INTERRUPT IF INTE _B HAS BEEN ENABLED

* $\overline{STB}_A$ and $\overline{STB}_B$ signals are the $\overline{EOC}$ END of conversion signal from the ADC. Therefore $\overline{INTR}_A$ and $\overline{INTR}_B$ are identical except for the $\overline{INTE}_A$ and $\overline{INTE}_B$ which are enabled and disabled individually by software control.

FIGURE 6. PORT AND CHANNEL CONTROL SOFTWARE FOR 12 BIT DATA FROM RANDOM ACCESS CHANNEL CONVERSION

INSTRUCTION ADDRESS	INSTRUCTION MNEMONIC	MICROPROCESSOR OPERATION	ANALOG TO DIGITAL CONVERTER SUBSYSTEM OPERATION
P(ort)	LXI H,	(HL) = ADC3	—
P + 1	C3		—
P + 2	AD		—
P + 3	MVI M,	(ADC3) = B0	Command Port A and Port C Bits ₇₋₃
P + 4	B0		to 8255 Mode 1: Port A Strobed
P + 5	MVI M,	(ADC3) = 86	Command Port B and Port C Bits ₂₋₀
P + 6	86		to 8255 Mode 1: Port B Strobed
P + 7	DCR H	(HL) = (HL) - 1 = ADC2	—
P + 8	MVI M,	(ADC2) = 0F	Set Port C } Command Zero Data Bit 7 = 1 } To Port B Bits ₃₋₀
P + 9	0F		
[ADC 8255 I/O Port Initialization is Complete Except for Analog Input Channel Selection and Interrupt Enable]			
C(hannel)	LXI H,	(HL) = ADC4	—
C + 1	C4		—
C + 2	AD		—
C + 3	MVI M,	(ADC4) = XY	Load XY into Analog Channel Address
C + 4	XY		Register and Initiate Conversion
C + 5	DCR H	(HL) = (HL) - 1 = ADC3	—
C + 6	DCR H	(HL) = (HL) - 1 = ADC2	—
C + 7	MVI M,	(ADC2) = 0D	Set Port C } Enable Port A Bit 4 = 1 } Interrupt
C + 8	0D		
[ADC Input Channel is Selected and Conversion is Initiated]			
R(ead)	LDA	(A) = (ADC1)	Read the lower 4-bits of the
P + 1	C1		Analog-to-Digital
P + 2	AD		Conversion
P + 3	MOV E,A	(E) = (A)	—
P + 4	LDA	(A) = (ADC0)	Read the upper 8-bits of the
P + 5	C0		Analog-to-Digital
P + 6	AD		Conversion
P + 7	MOV D,A	(D) = (A)	—
[Analog to Digital Conversion Data is Now Located in Register Pair DE]			

FIGURE 7. PORT AND CHANNEL CONTROL SOFTWARE FOR 12 BIT DATA FROM SEQUENTIAL CHANNEL CONVERSION

INSTRUCTION ADDRESS	INSTRUCTION MNEMONIC	MICROPROCESSOR OPERATION	ANALOG TO DIGITAL CONVERTER SUBSYSTEM OPERATION
P(ort)	LXI H,	(HL) = ADC3	—
P + 1	C3		—
P + 2	AD		—
P + 3	MVI M,	(ADC3) = B0	Command Port A and Port C Bits ₇₋₃
P + 4	B0		to 8255 Mode 1: Port A Strobed
P + 5	MVI M, -	(ADC3) = 86	Command Port B and Port C Bits ₂₋₀
P + 6	86		to 8255 Mode 1: Port B Strobed
P + 7	DCR H	(HL) = (HL) - 1 = ADC2	—
P + 8	MVI M,	(ADC2) = 0E	Set Port C
P + 9	0E		Bit 7 = 0
P + A	MVI M,	(ADC2) = 0D	Set Port C
P + B	0D		Bit 6 =
[ADC 8255 I/O Port Initialization is Complete Except for Analog Input Channel Selection and Interrupt Enable]			
C(hannel)	LXI H,	(HL) = ADC5	—
C + 1	C5		—
C + 2	AD		—
C + 3	MVI M,	(ADC5) = ZZ = DON'T CARE	Clear Analog Channel Address
C + 4	ZZ		Register and Initiate Conversion
C + 5	DCR H	(HL) = (HL) - 1 = ADC4	—
C + 6	DCR H	(HL) = (HL) - 1 = ADC3	—
C + 7	DCR H	(HL) = (HL) - 1 = ADC2	—
C + 8	MVI M,	(ADC2) = 0D	Set Port C
C + 9	0D		Bit 4 = 1
[ADC Input Channel "0" is Selected and Conversion is Initiated]			

FIGURE 7. (CONTINUED)

INSTRUCTION ADDRESS	INSTRUCTION MNEMONIC	MICROPROCESSOR OPERATION	ANALOG TO DIGITAL CONVERTER SUBSYSTEM OPERATION
R(ead) R + 1 R + 2	LDA C1 AD	(A) = (ADC1)	Read the Lower 4-bits of the Analog-to-Digital Conversion and ADC Channel Address
R + 3	MOV E,A	(E) = (A)	—
R + 4 R + 5 R + 6	LDA C0 AD	(A) = (ADC0)	Read the Upper 8-Bits of the Analog-to-Digital Conversion
R + 7	MOV D,A	(D) = (A)	—
[Analog-to-Digital Conversion Data and Channel Address is Not Located in Register Pair DE]			
A(dvance) A + 1 A + 2	LHI H, C6 AD	(HL) = ADC6	— — —
A + 3	MVI M,	(ADC6) = ZZ = DON'T CARE	Increment Analog Channel
A + 4 A + 5 A + 6 A + 7 A + 8	ZZ DCR H DCR H DCR H DCR H	(HL) = (HL) - 1 = ADC5 (HL) = (HL) - 1 = ADC4 (HL) = (HL) - 1 = ADC3 (HL) = (HL) - 1 = ADC2	Address Register and Initiate Conversion — — — —
A + 9 A + A	MVI M, 0D	(ADC2) = 0D	Set Port C Bit 4 = 1 } Enable Port A Interrupt

[ADC Input Channel Address is Incremented, the New Channel is Selected and Conversion is Initiated]