AN ABSTRACT OF THE THESIS OF

RODOLFO BLANCO-DURAN for the degree of MASTER OF SCIENCE in INDUSTRIAL ENGINEERING presented on August 17, 1981

TITLE: APPLICATIONS OF MICROPROCESSORS IN WORK MEASUREMENT

Abstract Approved: Redacted for Privacy

Dr. Michael S. Inoue

The main objective of this thesis is to investigate the microprocessor's potentials to improve the productivity of Industrial Engineers engaged in work design and measurement. More exactly, the following four hypotheses were proposed and investigated:

(1) A Universal Time Data (UTD) structure could be used to accommodate a variety of industrial needs.

(2) The data files created according to the Universal Data Base structure should be accessed and edited by the analyst much like the word processing machines used by office clerks.

(3) The programs developed could be written, compiled and stored onto 4096 bytes of Read Only Memory available in the computer.

(4) The timer functions of microprocessors could be used not only to measure and record time, but also to real-time simulate activities and output signals in order to control light panels and a speaker.

The thesis is comprised of two papers. The first paper, entitled "Microprocessors in Work Measurement: A Productivity Tool for MTM Analysts," comprising Chapter II of this thesis, is a tutorial introduction to the use of microprocessors to work analysis and design using the MTM predetermined time method.

The second paper, entitled "Design of a Universal Time Data Structure for Work Measurement and Design," is included in Chapter III of this thesis, and specifically addresses the above four hypotheses.

The set of programs developed have the ability to make the microcomputer operate as a multi-activity stopwatch, create the UTD files that are compatible with the microcomputer's Text Editor, inform the user of the additional observations needed to reach a given confidence level and interval, perform real-time simulation of the files, and generate MTM files and simograms.

The future areas of research include: more extensive uses of the I/O ports to further assist Industrial Engineers, incorporation of work sampling program, more extensive time data tables (including MTM-2, MTM-1, robot action tables, etc.), testing of the practicality of the developed system under actual industrial work conditions. Matching microprocessor and video-tape capabilities presents another challenging dimension to Industrial Engineering productivity.

Applications of Microprocessors in Work Measurement

by

RODOLFO BLANCO-DURAN

A THESIS

Submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed August 17, 1981

Commencement June 1982

APPROVED:

Redacted for Privacy

Professor of Industrial Engineering
in charge of major

Redacted for Privacy

Head of Department of Industrial Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented August 17, 1981

Thesis typed by MARY SYHLMAN for Rodolfo Blanco-Duran

ACKNOWLEDGMENTS

Special thanks to my wife. Her love and encouragement provided a very singular escape for my many frustrations along the learning process.

Sincere thanks to Dr. Michael Inoue. The knowledge I have gained from him has truly improved my understanding and enriched my spirit. The valuable experiences we shared will be unforgetable.

Thanks to Dr. James L. Riggs for his help and encouragement throughout my studies.

Thanks to Mr. Nichols for providing the means to send out the survey conducted.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

A - FIRST ARTICLE

B - SECOND ARTICLE

LIST OF TABLES

APPLICATIONS OF MICROPROCESSORS IN WORK MEASUREMENT

I. INTRODUCTION

Work Measurement, commonly known as Motion and Time Study, has long been the backbone of Industrial Engineering. Recent awakening of public interests in productivity for socio-economic reasons is renewing our realization of just how important it is to account for time, and how we spend our human life at work.

Yet, the very productivity of the time spent by Industrial Engineers and technicians assigned to collect and analyze time study data is questionable. There is no doubt that the replacement of mechanical stopwatches by electronic digital stopwatches, the introduction of video tape machines, the adaptation of predetermined time systems, such as MTM and Work Factor, and other technological advances are helping Industrial Engineers become more productive. Let us ask "Is this enough?"

Many Industrial Engineers advocate a "wholistic" systems-oriented approach to problem-solving. "Global" optimization is stressed and the "efficiency experts" are disdained. But our own approach to Work Measurement has been piece-meal, fragmanted, and sporatic. What we save by using electronic stopwatches and computers, we often waste by manually transcribing the output from one device only to have it re-entered into another. We recommend integrating Management Information Systems, adopting a comprehensive Decision Support System, implementing an MRP data base...all the while running around with a stopwatch in our hand, or worse yet, trusting data developed by someone else to apply to our situation so that we are spared of the drudgery of time study.

The Purpose of This Study

The purpose of this study, therefore, is to investigate the possibility of designing a Universal Time Data (UTD) structure that can be used as the foundation upon which to build a set of compatible tools to enhance the productivity of Industrial Engineers and Technicians engaged in Work Measurement.

More specifically, we wanted to find out how far we could push the use of microelectronics, microprocessors in particular, to facilitate our tasks of creating work measurement data files, analyzing them, testing them in real-time, and documenting and saving them for future use.

The Development of the Study

Several tasks had to be undertaken. First it was necessary to acquire the basic skills, tools, and techniques necessary to undertake this project. This meant learning and understanding how microprocessors work., how they interact with other components of a computer system, etc. This also meant gaining the expertise to use machine language, assembly language, BASIC, monitor subroutines, machine language code editor (COED), etc. This learning phase included three months spent taking the undergraduate microcomputer course in Industrial Engineering, three more months to become proficient and understand how the microprocessors interacted with the environment through Versatile

Interface Adapters. This effort culminated in a presentation made at the Joint CORS (Canadian Operational Research Society), TIMS (The Institute of Management Science) and ORSA (Operations Research Society of America) National Meeting held in Toronto, Canada, in May 1981.

The second task was to fully appreciate and understand the nature of Industrial Engineering tasks in Work Measurement. After reviewing published materials, a survey form was constructed and given to nine Pacific Northwest companies. Six of them replied and the survey results are included in the Appendix C of this thesis. Industrial Engineers at Intel were consulted and the AIM 65 was demonstrated with some sample programs. Out of these initial efforts, the Universal Time Data (UTD) structure was designed.

Realizing the importance of MTM-based time studies in industry led us to examine this particular application in detail. MTM-3 was selected as the basis because of its simplicity and increasing popularity over other tools. Another consideration was the realization of the power of "real-time simulation" as an effective new tool for Industrial Engineers. For such a simulation, MTM-1 and MTM-2 elements were usually too small and awkward to be really useful. This phase of the research culminated in the paper titled "Microprocessors in Work Measurement. A Productivity Tool for MTM Analysts," and contained in Chapter II.

Along the path of the study, we encountered a series of "subgoals" or hypotheses we were testing. These were:

(1) A universal Data Base structure could be used to accomodate a variety of industrial needs.

(2) Data files created according to the UTD could be accessed and edited by the analyst, much like the way office clerks use word processing machines.

(3) The programs needed could be written, compiled and stored onto a limited size Read Only Memory (e.g., 4-K bytes ROM), so that the system is "turn-key" ready.

(4) The timer functions of microprocessors could be used not only to measure and record time data, but also to real-time simulate activities recorded in the Random Assess Memory (e.g., 4-K bytes RAM).

Real-time collection of data, by modifying the microcomputer system to act as a multi-activity stopwatch, added a new dimension to the utility of UTD. Times collected were in cumulative MUs (one-millionths of an hour), and a program had to be developed to convert the cumulative time data to individual time data as in the MTM usage.

The study of the IE applications of work measurement data brought in the "rating" or "speed" consideration. A new program was then developed to change the entire data file by multiplying them with a user specified constant. Thus, if the IE department wanted to simulate and create an 80% time UTD file, this could be done easily from a 100% UTD file stored on a casette tape. This feature could be useful in training, learning-curve verification, slow-motion analysis with videotape or movie film, memomotion analyses, etc. Also a program was designed for the creation of standard time data files when the analyst finishes the time study. This phase of the study is included in Chapter III of this thesis, under the title "Design of a Universal Time Data Structure for Work Measurement and Design."

Outcome of the Study

Overall, the project proved to be even more satisfying than we had ever expected initially. All major hypotheses turned out to be positively supported, and the resulting firmware surpassed the original expectations of how much features could be compressed into a 4K-Byte Read-Only-Memory (ROM).

There are still many unfinished tasks, and those are identified in Chapter IV, the Conclusion and Recommendation for Future Research Chapter. The backup data and specialized procedure for "burning" the Erasable Programmable ROM (EPROM) using the CO-ED editor are included in the appendices.

Two papers are expected to be submitted for publication to *MTM Journal* and *AIIE Transactions*, and a copy of each is forwarded to Rockwell International as a way to acknowledge the assistance the company has given to the Industrial and General Engineering Department by providing the hardware.

II. MICROPROCESSORS IN WORK MEASUREMENT

A Productivity Tool for MTM Analysts

Rodolfo Blanco, Professor, Industrial Engineering, Instituto Technologico De Costa, Rica, Apartado 159, Cartago, Costa Rica

Michael S. Inoue, Professor, Industrial and General Engineering, Oregon State University, Corvallis, Oregon 97331

An integrated data base structure for a microprocessor-based computer system to enter, edit, simulate, and document MTM-3 and other time elements is presented.

Work measurement is a cornerstone of all productivity studies, and MTM is the most widely used predetermined time work measurement technique. MTM's popularity is largely based upon its ability to improve the productivity of work measurement analysts.

The microprocessor is a low cost machine intelligence that Fortune magazine called the "crude oil of industrial progress." Produced by millions, a modern microprocessor packs as much computing power on a chip of silicon as a roomful of vacuum tubes did just a couple of decades ago. These chips are the brains behind microcomputers, digital watches, pocket calculators, word processing machines, TV games, cash registers and even home appliances. The microprocessor's popularity is largely based upon its ability to improve productivity of machines and their users.

It is perhaps time for us to look at a marriage of the two "productivity tools" and see how they can help us be ready for the increasingly automated factories of tomorrow.

Increasing Importance of Work Measurement

An increasing number of companies are conducting studies to investigate the viability of introducing robots and other automated production and materials handling devices. A thorough work measurement study is a prerequisite to an accurate study that helps prevent costly mistakes in capital expenditures. Tektronix in Beaverton, Oregon, for example, is using computer models to make comparison studies. In a small survey of six companies in the Pacific Northwest region in 1981, authors found that all except one company used work measurement techniques to justify NC machines and/or robot purchases.

Another reason for the increasing importance of Work Measurement is the industry's need to integrate machines and humans in such a way as to maximize the quality of work life. Rather than to make humans slave to machines, machines should be utilized to maximize human productivity. As is exemplified by the Quality Circle Movement, managers are realizing that it is no longer desirable to employ workers solely for their hands and feet but more importantly for their brains. Work Measurement measures time, and time is how we measure the length of human life. A productive company must also provide an opportunity for employees to grow and feel productive.

Work Measurement Tasks

Traditional Approach

In an article entitled, "Today's Computers and Work Measurement[43]," found in the January 1980 issue of The Journal of Methods-Time Measurement, Douglas M. Towne identified twelve areas of Work Measurement that were suitable for computer data processing by four types of computers: (A) Large-Central Computers, (B) Time-Shared Systems, (C) Mini-computers, and (D) Microcomputers. Not a single one of the four types of computers was recommended as applicable to all twelve areas of Work Measurement. Microcomputers were considered applicable to only seven areas and excelling over other types of computers in five of those areas. Though the article illustrated several application examples, they were all based upon different combinations of hardware and software.

A typical work measurement study requires, as shown in Figure 1:

(A) The subject to be studied: actual, filmed, or abstract man-machine system(s) being observed.

(B) The observer: human or automated monitor(s) equipped with appropriate time measurement equipment (e.g., stop-watch).

(C) Analysis techniques: MTM tables, sampling and other statistical data processing techniques, rating and allowance criteria, etc.

(D) Applications: productivity measurement, time standards, methods description, learning curves, simo-chart and man-machine scheduling of concurrent

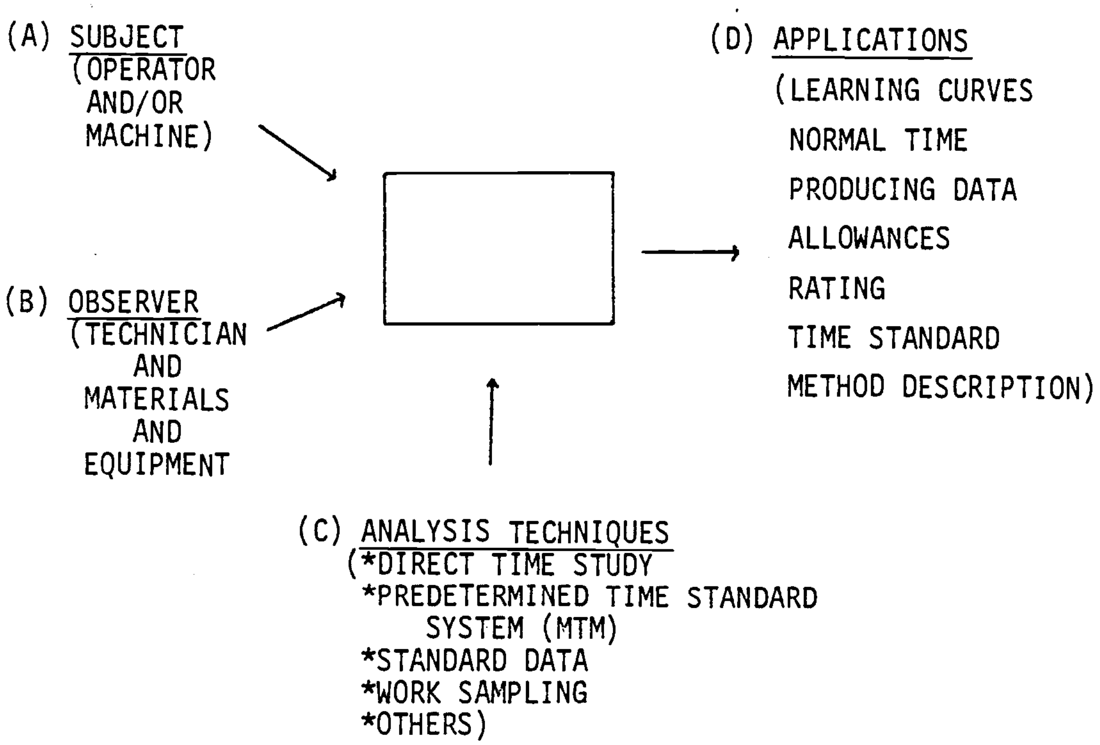


FIGURE 1. Work Measurement Scheme.

activities, line balancing, etc.

As Tom Mofett, an Industrial Engineer with the Weyerhaeuser Company in Springfield, Oregon commented, considerable time is wasted by analysts because of tools not suitable to make their jobs more productive.

MTM Methods

The predetermined time standard system, typified by the Methods-Time Measurement, may be the most significant "productivity aid" for work measurement analysts.

A typical MTM analysis starts out by identifying the operation to be analyzed by breaking that operation into micromovements. Once the whole operation has been analyzed into micromovements for the operator's right hand, left hand, and body motion, the analyst can use an MTM table to look up the basic time data to compute the normal time. Typically, the data must then be integrated with timing of machineries and other external events that affect the operation. A simo-chart may be used to synchronize body motions, and a man-machine or a gang chart may be used to synchronize the operations of men and machines.

Somewhere along the line, the analyst may want to use other techniques, such as direct timing by stop-watch, work sampling, memomotion study, etc., and integrate these observations in order to rate the operation and add allowances.

Many iterations of data input, data editing, analysis, simulation, documentation, and evaluation are repeated before the study is complete.

Yet, even the most carefully engineered time data will need checking and updating as methods, equipment, and operators change. Maintaining a viable standard time data base is a never-ending battle.

MTM-3

The MTM-3 is the fastest and most increasingly popular MTM method used to facilitate the creation and maintenance of accurate time data base. MTM-3 is said to be about seven times faster to apply than MTM-1 and twice as fast as MTM-2. Its accuracy is within ±5% with 95% confidence when used in observation of operations with cycle times of four minutes or longer.(11)

The MTM-3 Application card (Figure 2) illustrates its four basic elements (24):

1. HANDLE (H) Handle is a motion sequence with the purpose of gaining control over an object and placing the object in a new location.

2. TRANSPORT (T) Transport is when the hand, already controlling the object, moves the object to a new location, positioning it if necessary and releasing it.

3. STEP (SF) Step consists of the operator moving his/her foot or leg in a horizontal direction.

4. BEND (B) Bend is the operator's body movement of lowering the trunk followed by an arise.

There are operator's eye movements, machine operations, and other elements that cannot be directly identified by the MTM-3.

Integrated Approach

To fully take advantage of the simplicity and efficiency of MTM-3, an integrated approach is proposed. The approach is to structure an integrated time data format that is compatible to MTM-3 as well as to other work measurement techniques and to design a series of computer programs that will be automatically executed to assist the MTM analysts.

Figure 3 illustrates the integrated approach.

1. INPUT The analyst will have the options of:

(A) Directly entering the time data and short description of each micromovement into the computer Random Access Memory (RAM).

(B) Identifying MTM-3 symbols to have the computer look up the MTM-3 time value.

(C) Entering observed events in a chronological sequence using the cumulative time method of the stopwatch.

(D) Having the computer automatically collect time data in the chronological order as in C.

2. EDITING The data file created through any of the methods described by the input process should be accessible to the analyst so that he/she may perform any of the following functions:

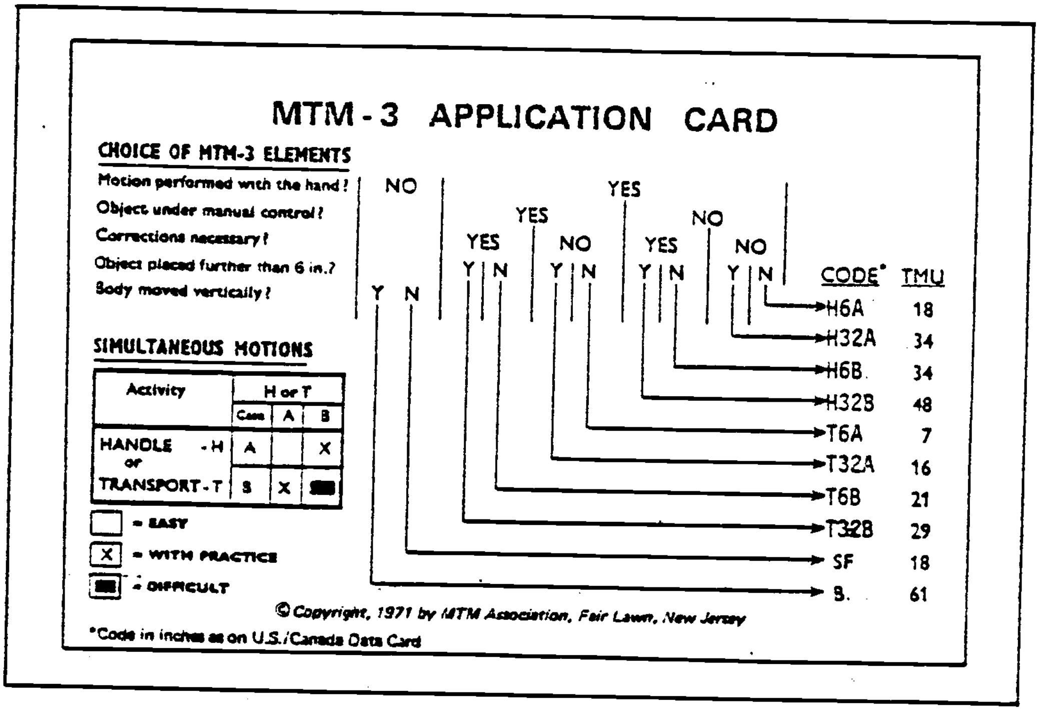
(T) Enter Text Editor at the top of its data file.

(L) List the data file as stored in the computer.

(I) Insert new micromovements, machine status, etc.

(K) Kill or delete one or more micromovements.

(D) Move down to the next micromovement description.



MTM-3 APPLICATION CARD

CHOICE OF MTM-3 ELEMENTS

Motion performed with the hand?
Object under manual control?
Corrections necessary?
Object placed further than 6 in.?
Body moved vertically?

CODE*	TMU
H6A	18
H32A	34
H6B	34
H32B	48
T6A	7
T32A	16
T6B	21
T32B	29
SF	18
B	61

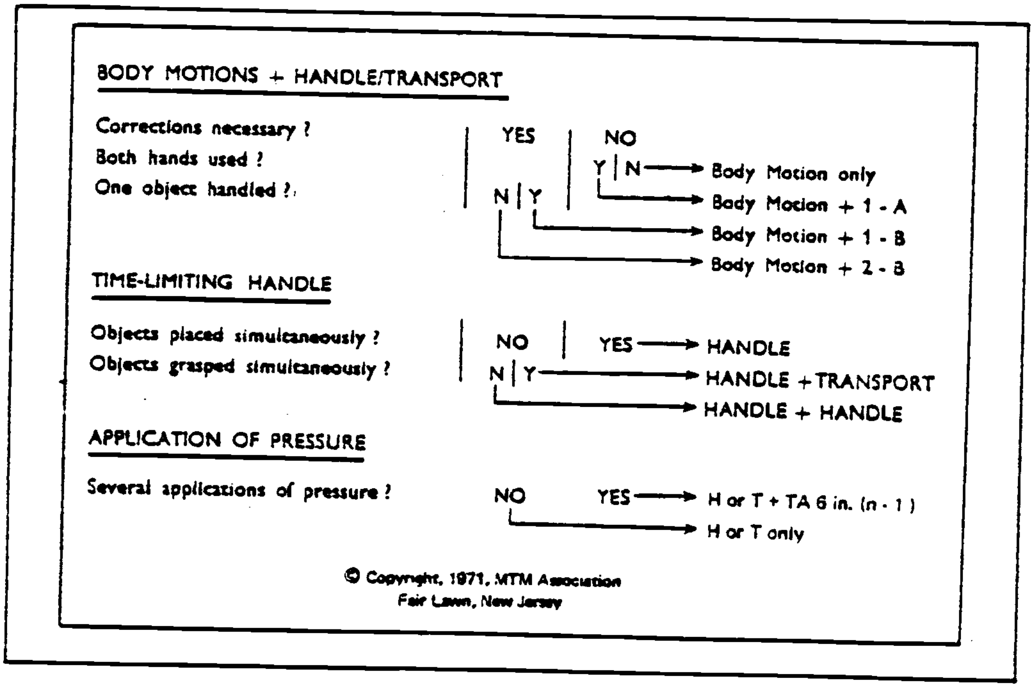
SIMULTANEOUS MOTIONS

Activity	H or T	
	Case A	B
HANDLE -H or TRANSPORT -T: A		X
B	X	■

□ = EASY
☒ = WITH PRACTICE
■ = DIFFICULT

© Copyright, 1971 by MTM Association, Fair Lawn, New Jersey

*Code in inches as on U.S./Canada Data Card



BODY MOTIONS + HANDLE/TRANSPORT

Corrections necessary?
Both hands used?
One object handled?

Body Motion only
Body Motion + 1 - A
Body Motion + 1 - B
Body Motion + 2 - B

TIME-LIMITING HANDLE

Objects placed simultaneously?
Objects grasped simultaneously?

HANDLE
HANDLE + TRANSPORT
HANDLE + HANDLE

APPLICATION OF PRESSURE

Several applications of pressure?

H or T + TA 6 in. (n - 1)
H or T only

© Copyright, 1971, MTM Association
Fair Lawn, New Jersey

FIGURE 2. MTM-3 Application Card.

(U) Move up to the previous micromovement description.

(B) Go to the bottom of the data file.

(F) Find the micromovement containing a specified chapter string.

(C) Change a character string in the line by another string.

(Q) Quit and exit the editor to return to the monitor program.

3. OUTPUT The data file created and edited by any of the above methods may then be subject to the following:

(1) Simogram: sorting data according to the type of subject (e.g., Left-Hand, Machine #1, Operator #3, e.g.) and producing a cumulative time listing for each.

(2) Simulation: real-time simulates the data file by allowing as many as four subjects to be displayed simultaneously on the computer display for the actual time corresponding to the duration of the time elements.

(3) Storage and Retrieval of Data Files: through the Text Editor it is possible to: (R) Read a data file that is stored in a casette-tape, and (L) List onto a casette-tape, for permanent storage, the current data file in the editor.

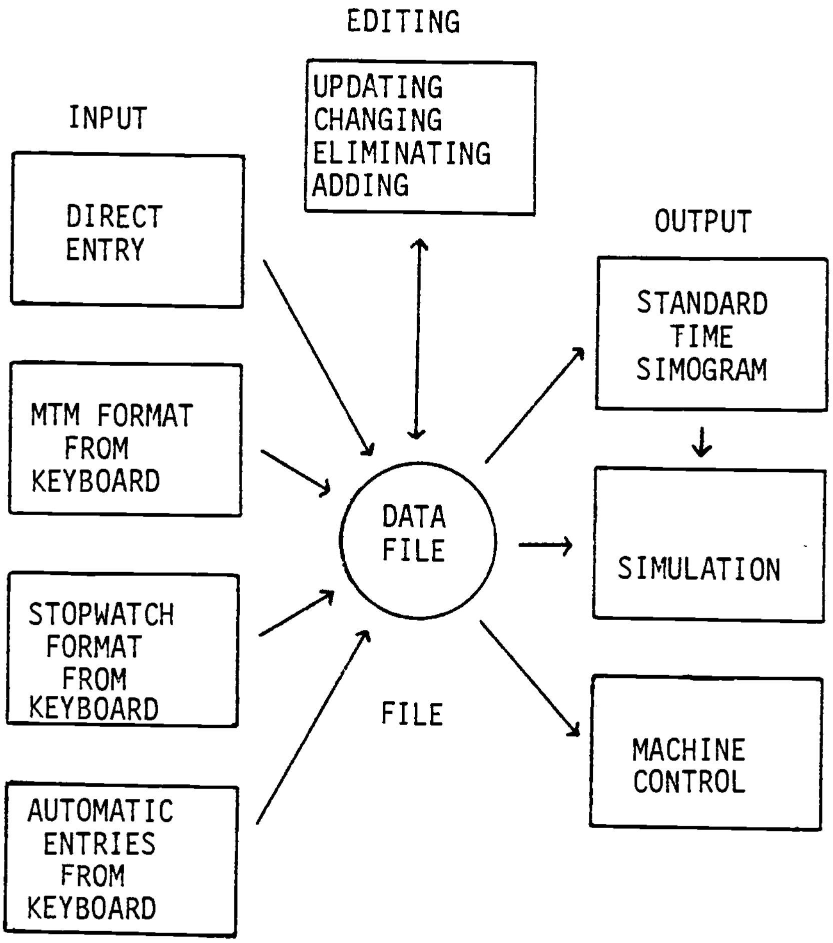


FIGURE 3. Integrated Approach

(4) Direct Control: directly control machineries and equipment or display the alternating on/off condition as specified on the four channel simulation program.

These activities were selected to enable broad applications of MTM-3 as well as other time measurement tools and activities to be integrated into one system. To design a microprocessor based computer system capable of accomplishing these tasks, we need to look at what a microprocessor is and what are the design limitations that accompany its usage.

Microprocessor-based Computer System

Design Criteria

A work measurement tool based on the microprocessor technology will have to meet several criteria. Some of the more important criteria that we have identified are:

1. EASE OF USAGE The system should be ready to go just as soon as its switch is turned on. Complicated booth-trapping and start-up procedure may be fine for a large-scale computer, but highly impractical for a tool to be used daily by an MTM analyst. There should be no need to insert disks, load programs, or perform other tasks that are not directly related to work measurement.

2. FLEXIBILITY The MTM analyst should always remain on top of the computer, and not be slave to it. It is therefore preferable that the user be able to always access the data files and modify them in any way that the application deems necessary. All data

should therefore follow a universally accepted format, the American Standard Codes for Information Interchange (ASCII).

3. COST The total cost of the system should be within reach of individual analysts. We have arbitrarily set this limit to be around $1,000.

4. DISPLAY AND PRINTING The analyst should not have to use pencil and paper to record data that are displayed by the computer system. There should be an on-board alphanumeric display and printer so that the entire system can be transported easily.

5. COMPUTER CAPABILITY It is desirable to have the system capable of understanding a user-oriented language, such as the Beginner's All-purpose Symbolic Instruction Codes (BASIC), so that the user can also use the system to write his/her own programs or perform immediate computations as on a scientific programmable calculator.

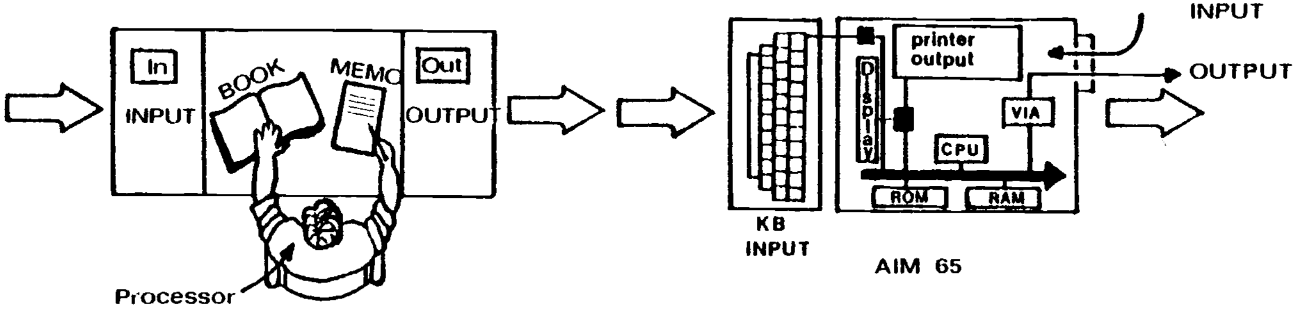
What is a Computer?

Before we can appreciate what a microprocessor is, we need to understand what a computer is. A computer is basically a machine-intelligence system that can accept data, remember them, process them according to a given set of instructions, and output the results in a form that can be used.

Figure 4[18] illustrates the relationship between a human analyst and a computer. The "in" basket containing raw work measurement data is comparable to the keyboard on a computer. The data are then in an electronically detectable format allowing them to flow through the system at the speed of light. The Central Processing Unit (CPU) is like the analyst following the MTM tables in a book to record the time data on his memo pad. As he performs mental calculations and logical reasoning, the analyst is likely to write and erase and rewrite over his memo pad. He can write on the top, bottom, or center of the memopad, just as he pleases. In a computer, a Random Access Memory (RAM) fills such a function. The book, on the other hand, is read, but not written over. An analyst does not want to change the MTM table values to fit the answer he wants. Such a memory in a computer system is known as a Read Only Memory (ROM). When the work is finished, the analyst may put his report in the "out" basket, post the results for others to see, or call up his boss to tell him/her what was found from the study. A computer may have a visual display, a line printer, a tone generator, or a telecommunication link to another machine, terminal, or storage device.

In short, a computer is a machine intelligence created in the image of a rational human being. Its purpose is to eliminate the mental drudgery that does not require human intervention.

A COMPUTER IS A MACHINE INTELLIGENCE CREATED IN THE IMAGE OF US, TO ELIMINATE THE MENTAL DRUDGERY OF OUR WORK.



ROM = READ ONLY MEMORY = BOOK OF RULES

RAM = RANDOM ACCESS MEMORY = MEMO PAD

CPU = CENTRAL PROCESSING UNIT = BRAIN

FIGURE 4. Relationships between a human analyst and a computer. (Artwork by Judy A. Witt)

Microprocessor

A microprocessor is really the brain of the computer on a silicon chip. The so-called IC or Integrated Circuit, contains layers of charged silicon films that act like thousands of transistors that are turned on or off dependent upon the signal received. A typical IC contains from 1,000 to 100,000 transistor equivalents. Each on-off switching is represented by the binary integer 0 to 1. A series of eight of these "bits" is called a "byte." A byte can represent an ASCII character or a letter. For example, "A" is 0100 0001 and "1" is 0011 0001 in bits. To simplify the notation, each half of a byte (called a "nibble") is given a numeric value from 0 (for 0000) to 15 (for 1111), except that the letters A through F are used for the numbers between 10 and 15. Thus, the letter "A" is $0100\ 0001_2$ or 41_h in this new notation called "hexadecimal". "1" is $0011\ 0001_2$ in binary and 31_h in hexadecimal. "/" is a symbol that is represented by $0010\ 1111_2$ or $2F_h$ in hexadecimal. Thus "M5A" in a computer looks like $4D\ 35\ 41_h$ in ASCII.

An integrated Circuit that is used strictly for remembering data such as these is known as a Random Access Memory. On the other hand, the IC that performs arithmetic and logic decisions and is capable of directing the flow of data from one address location in a RAM to another, is really a CPU (the MTM analyst in our previous analogy) and is called a microprocessor. An IC that interfaces the computer system with the outside world is known as the Versatile Interface Adapter or VIA as shown in Figure 4.

Microcomputer

A microcomputer is really a computer that is made up of IC chips: one microprocessor for a CPU, several RAMs, and one or more VIAs. Usually, it also has several ROMs where commonly used programs such as the monitor for the system, the Text Editor, and BASIC language interpreter reside. This is the reason why most microcomputers can be simply turned on and operated directly without having to load the monitor program or the language interpreter. A pocket calculator, a cash register, and a traffic controller may share the same microprocessor and RAMs. It is the program stored in ROM and the VIA interfacing that makes one system look differently from the other.

FIRMWARE

The electronic components such as the ICs are called the hardware. The computer programs written by the users are called the software. Finally, the programs that are written onto a ROM permanently are known as "Firmware", an obvious interpolation between the hard and the soft!

Advanced Interactive Microcomputer AIM 65

The microprocessor system that we have chosen to carry out this project (Fig. 5) is the Advanced Interactive Microcomputer AIM 65, manufactured by Rockwell International. Its Central Processing Unit is the microprocessor 6502 which has the capability of addressing 65,536 address locations in RAMs or ROMs, and perform either binary

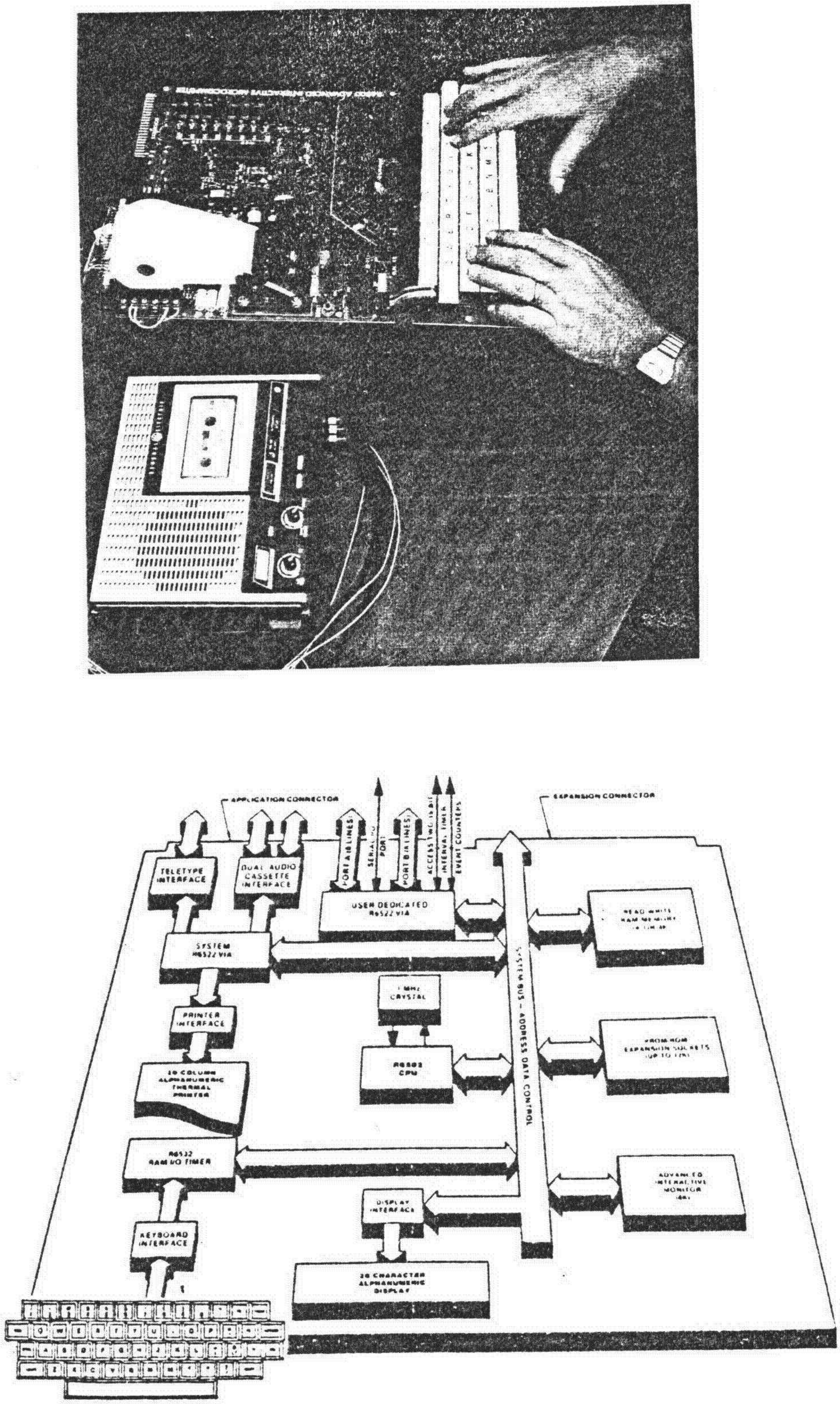


FIGURE 5. Rockwell International's AIM 65

or decimal arithmetic. For example, if the location 1000 on a RAM contained the binary number 0000 1001_2 and the location 1001 contained 0000 0001_2, a binary addition would yield $09_h + 01_h = 0A_h$ (numerically 10) or 0000 1010_2. The same arithmetic performed in the decimal mode would yield $09_h + 01_h = 10_h$ or 0001 0000_2.

The 6502 microprocessor also has 56 commands and 13 addressing modes, making it one of the most versatile central processing units. The same 6502 is used in Apple, Atari, Commodore, PET, and other microcomputer's industrial applications.

The AIM 65 contains 4,096 bytes (4KB) of RAM and 20K bytes available for plug in sockets of ROMs. These sockets are usually filled with ROMs containing the monitor, Text Editor, BASIC interpreter, and an assembler. The assembler is used by professional programmers to create efficient machine language codes of the type that most firmwares are made of.

The AIM 65 is what is known as a "single-board" computer since the entire system is on a circuit board. On a larger computer system one board may simply contain one CPU or a segment of one.

This single-board computer also has a 20-character alphanumeric display and a 20-character alphanumeric printer, as well as a typewriter-type keyboard. The original cost of the equipment for a stripped version with just 1KB RAMs and no BASIC or power supply was around $400. The complete unit cost us around $1,000.

Timer

Another feature of the AIM 65 is that it includes several VIAs. One VIA is especially dedicated for the user interface applications. The VIA, in order to coordinate computer activities with outside activities, includes two timers which can keep time separately. These 16-bit timers can be used to time operations, create musical tunes to be played over a speaker and to count frequency of input/output signals.

Hardware Selection

To summarize, the hardware selection of the AIM 65 was based upon several considerations.

1. 6502: The microprocessor's ability to handle decimal arithmetic made it more suitable for ASCII data.
2. VIA: The two timers available on the AIM 65 means that we can time simulation and generate "beep" tones for the analyst to identify the beginning and ending of each micromovement. Two 16-bit ports available on the VIA were considered adequate for direct observation or control of machineries.
3. FIRMWARE: The Text Editor, BASIC interpreter, and the system monitor are already available and their subroutines can be used by the work measurement programs. The Assembler slot will be replaced by the ROM containing the work measurement programs.

4. PRINTER AND DISPLAY: The alphanumeric display and printer were considered essential to work measurement analyses.

5. COST, SIZE, WEIGHT: The physical size and weight, as well as the low cost of the hardware were considered favorable factors.

Several manufacturers were contacted but no commercially available equipment appeared to match the offerings of the AIM 65 at a comparable price.

Integrated Work Measurement System

System Architecture

The microprocessor-based work measurement system proposed in this paper is called integrated for several reasons:

(A) It combines the MTM-3 predetermined time data with data generated using other sources such as MTM-2, stopwatch timing, and memo-motion study. Data files may contain both MTM and non-MTM elements.

(B) It follows a systematic approach from input to output without requiring the user to copy the data and transfer them from one system to another manually. The output is obtained after the most appropriate inputs are combined and the data file is edited and reviewed through real-time simulation.

Data Structure

To be compatible with the MTM-3, stopwatch-type observations, and ASCII based Text Editors, a data structure was formulated to accommodate the most flexibility. Each time data record is composed of seventeen bytes. Each observation is recorded as shown in Figure 6:

Byte 1: Record Type (F for Foreign, T for Comment, S for the Start of an element, E for the End of an element).

Bytes 2 and 3: Subject being observed. A number between 00 and 99 in order to identify the operator, right or left arm, body, machine, etc.

Bytes 3 through 6: Element's ID or description, such as H32A, IDLE, or RUN.

Byte 7: (Reserved for future use)

Byte 8 through 16: Observed time in MU's where 1 MU = 1/10 TMU = $1/10^6$ hour.

Byte 17: $0D_h$ used as the end of the record mark.

The eight bytes representation of time data allows up to 100 hours of operations to be recorded. The last byte is used to denote the end of the element observation. Under this format, every observation is taken as a pair consisting of an element description and its element time. This format is the same regardless of whether the MTM format is used or a stopwatch format is used. Figure 7 is an example of a "mixed" data file created by combining stopwatch and MTM data.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a	b		c				d	e								f

a = Record type

F = Foreign element
S = Start of element
E or C = End of element
T = Comment line

b = Subject that executes the element
c = Elements ID or description
d = Unused
e = Observation time (MU's)
f = End of line code

FIGURE 6. Standard data base format. Seventeen bytes per element observation.

Memory Map

The microprocessor's ability to address 65KB of memory means that these addresses will have to be divided in such a way as to accommodate both the programs and data. The address locations are shown in hexadecimal notations. Thus 00 00_h is location 0000 0000 0000 0000_2 or 0 in decimal. $00FF_h$ is 0000 0000 1111 1111_2 and there are 256 bytes between 00 00_h to 00 FF_h. This amount of memory is known as a "page". There are potentially 256 pages of 256 bytes each, or 65, 536 bytes (65KB) of addresses that are available. In fact, some of these addresses are used to identify input and output ports while others are left unused for future expansion. The AIM 65 system with this work measurement package has a memory map as shown in Figure 8.

System Flow (Figure 9)

When an AIM 65 is activated, it automatically goes to the monitor (loc $E000_h$ to $FFFF_h$) which directs the display and printer to output the message "ROCKWELL AIM 65".

When the key "N" is depressed, the Work Measurement Menu is activated. This changes the program counter of the computer to location D0 00_h where the menu program is stored. The menu gives the option:

```
*ENTER*
M=MTM FILE
G=SIMOGRAM
U=SIMULATE
ESC=TEXT EDITOR USE
=?
```

Operator	Element	Time/ Piece	Left Hand	Time	Time	Right Hand
1	Adjusting Machine	35	R30A	17.5	16.7	R28A
1	Operating Machine	48	G1C1	7.3	5.6	G2
1	Unloading Machine	24	M14C	16.9	16.7	M14C

Time = 1/100 Minute

Time = TMU

Integrated Data File Format		
E1	ADMA	00005833
E1	OPMA	00008000
E1	UNMA	00004000
E1	R30A	00000175
E2	R28A	00000167
E1	G1C1	00000073
E2	G2	00000056
E1	M14C	00000169
E2	M14C	00000169

FIGURE 7. Mixed Data File

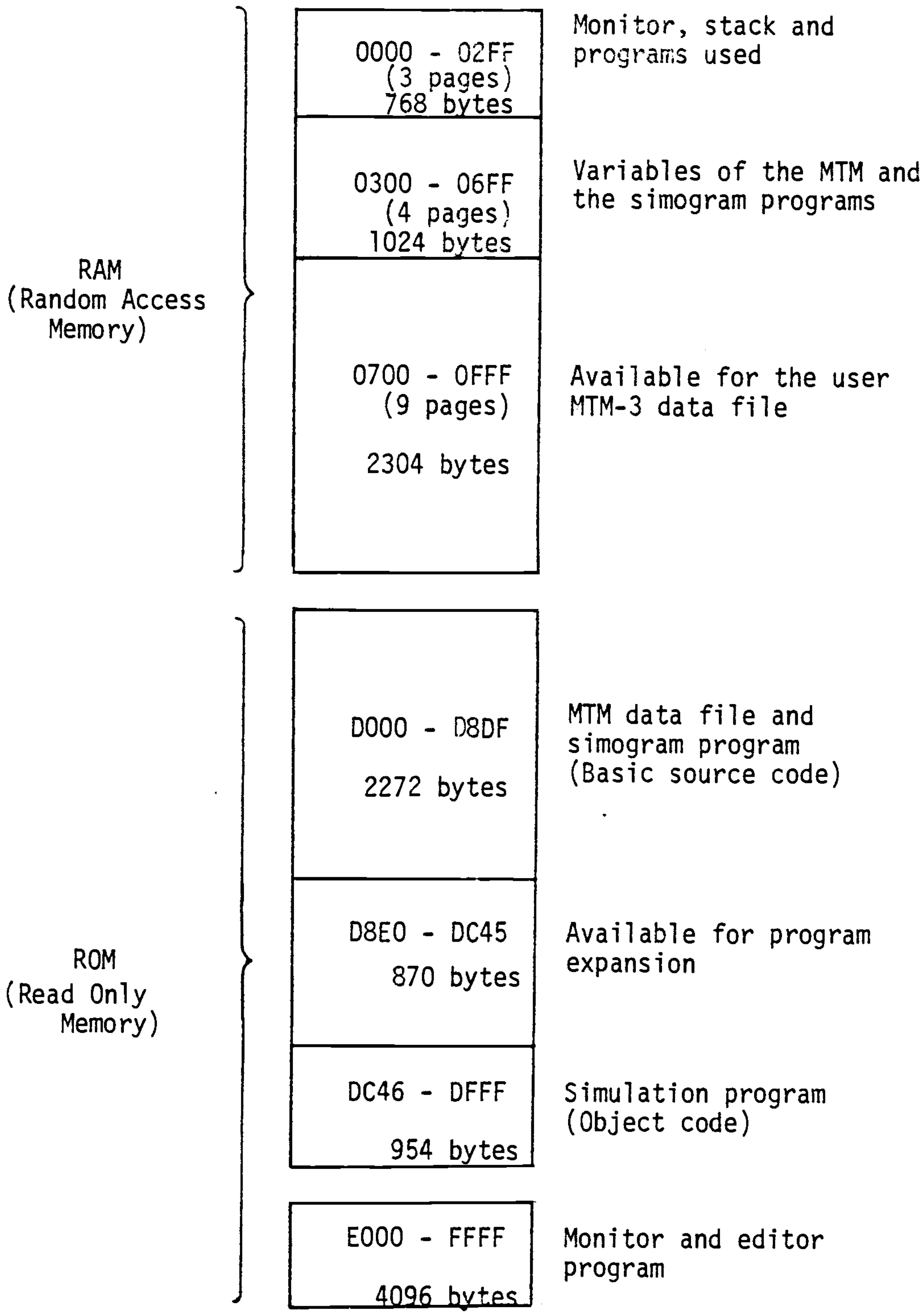

RAM
(Random Access Memory)
0000 - 02FF
(3 pages)
768 bytes
Monitor, stack and programs used
0300 - 06FF
(4 pages)
1024 bytes
Variables of the MTM and the simogram programs
0700 - 0FFF
(9 pages)
2304 bytes
Available for the user MTM-3 data file
ROM
(Read Only Memory)
D000 - D8DF
2272 bytes
MTM data file and simogram program (Basic source code)
D8E0 - DC45
870 bytes
Available for program expansion
DC46 - DFFF
954 bytes
Simulation program (Object code)
E000 - FFFF
4096 bytes
Monitor and editor program

FIGURE 8: Memory Map

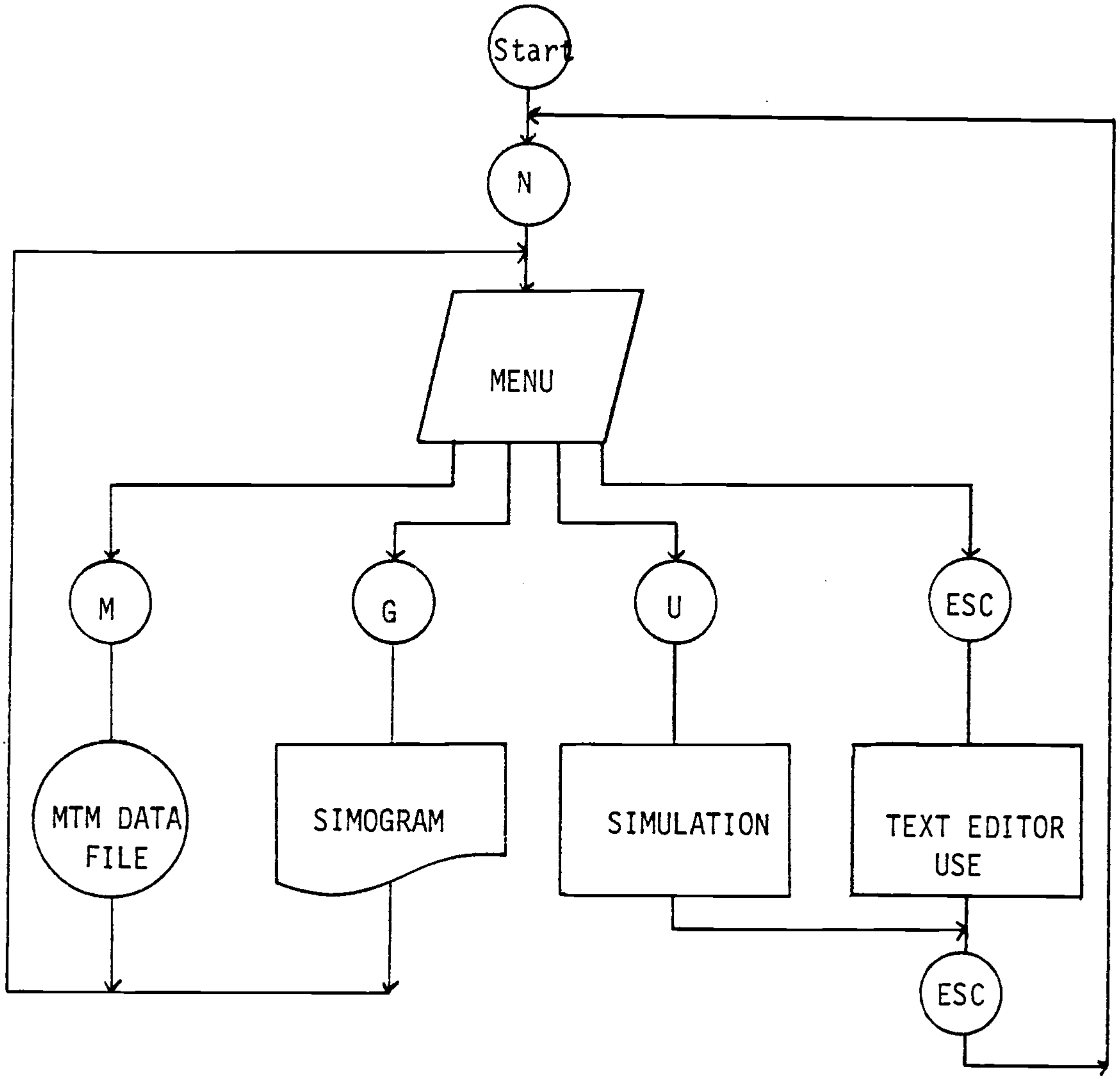


FIGURE 9. System Flowchart

Depressing M or G will put the starting address of the MTM data file creation or of the simogram program in the Program Counter so that the microprocessor will start following these programs in ROM. The actual data created during these programs will reside in locations 0700_h to $0FFF_h$ (2,304 bytes of space) which are in the RAM section, since ROM locations are "read-only" and cannot be written on.

Depressing U sends the microprocessor to follow the simulation program stored in locations DC 46_h to DF FF_h (954 bytes). Depressing the escape ("ESC") key or the RESET button will send the computer back to the AIM 65 Monitor.

From the monitor we have the following options:

N to return to the work-measurement program.

T to go to the text editor where the data file is now stored.

E to erase the content of the text editor and the current file in it. The text editor pointers will be automatically reset if "N" and then "M" are depressed.

5 Depressing 5 lets the user enter the BASIC interpreter.

In order to create a BASIC program to analyze data stored in the Editor area, we enter 1791 as the answer to "MEMORY SIZE?" This will enable the user to write the BASIC program in locations 02 13 to 06 FF_h (1261 bytes). Caution must be exercised not to depress N when the BASIC program is to be executed. N will change the BASIC pointer to the BASIC program stored in location D000 to D8DF. The simulation program in locations DC46 to DFFF is written in machine codes and is not affected by the BASIC program.

6 Key 6 is used to reenter the BASIC program without losing its content. If key N is used, key 6 will direct you to the current BASIC program.

Process Flow

A typical process flow followed by an MTM analyst is as shown in Figure 10.

Step 1: Initialization Depress N to enter the menu.

Step 2: MTM file creation Depress M and CR to enter the MTM file creation program.

* To enter a title for this data file, answer Y to "TITLE(Y-N)?"
* If Y is depressed, the computer will say "ENTER MAX 15 CHRS" as a reminder that the title cannot be longer than 15 character counting spaces. Use the DEL key to backspace over errors.
* To "SUBJECT #?" answer by typing a number between 1 and 9.
* To "MAX 4 CHR NAME?" type in either the MTM-3 codes, such as H32A, T6B, or a 1 to 4 character description such as "IDLE", "WORK", etc. If the name is one of the MTM-3 symbols below, the computer will automatically find the time value. Otherwise the program will ask "#### TIME(TMU)=?" where #### is the name entered.

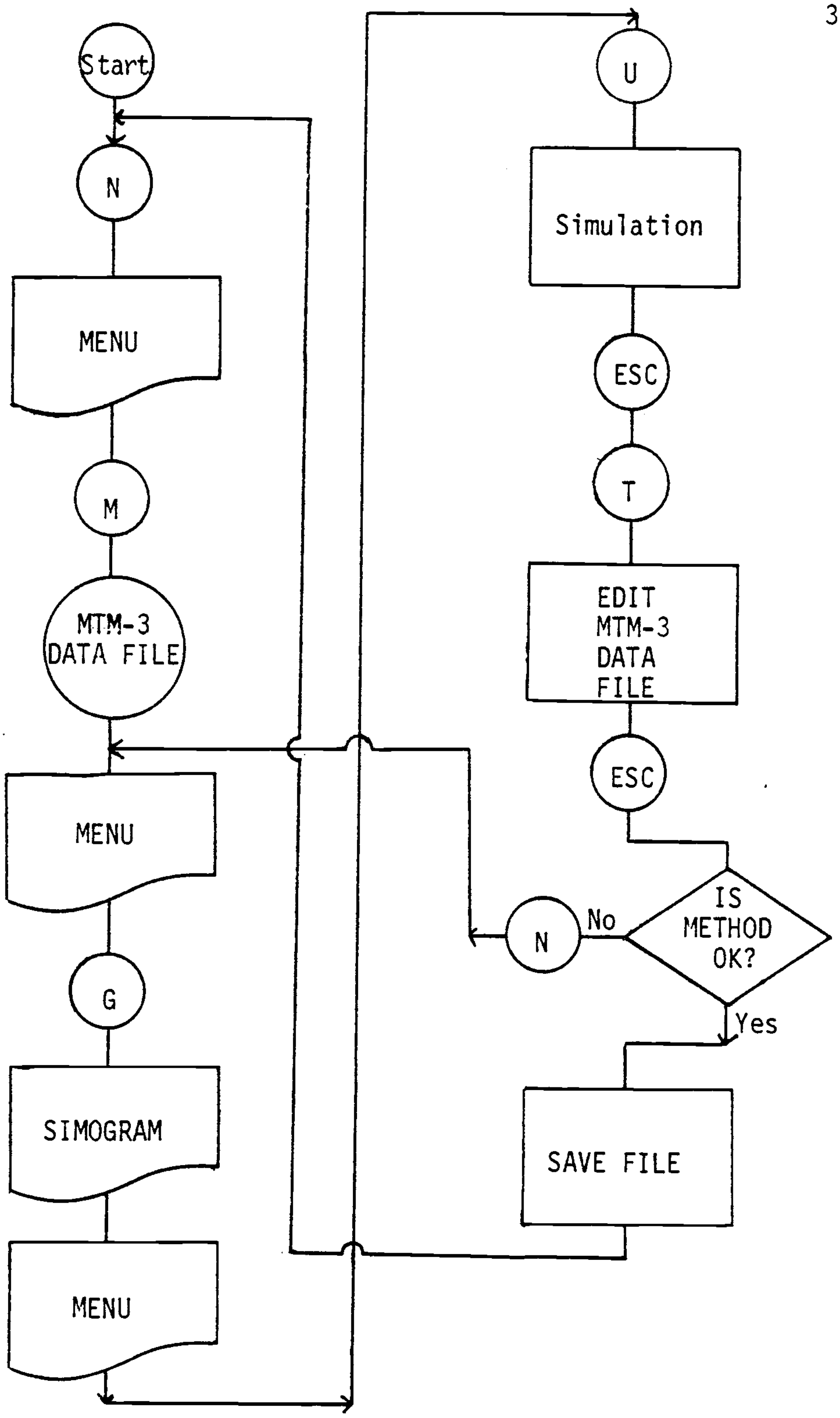


FIGURE 10. A Typical Process Flow

The MTM-3 names are entered with the numerical data sandwiched in between the alphabet codes. Valid codes are:

H6A H32A H6B H32B T6A T32A T6B T32B SF STEP B BEND

- * To end the data entry, type END when the question asks "SUBJECT #?"
- * It is not necessary to have grouped together all the observations for one subject. Within each subject matter data must be ordered chronologically. For example, we may either enter:

Order	Subject	Name	Time (TMU)	
1	1	H32A	--	(Time of 34TMU is automatically generated)
2	3	IDLE	30	
3	1	H32B	--	(Time of 48TMU is automatically generated)
4	3	BEND	--	(Time of 61 TMU is automatically generated)
	END			

or;

Order	Subject	Name	Time (TMU)
1	3	IDLE	30
2	3	BEND	--
3	1	H32A	--
4	1	H32B	--
	END		

- * After the END is entered, the computer will print out the number of entries and remind the user to "EDIT, LIST, SAVE FILE" before shutting it off.

The computer message for the above short example will be:

```
 4 ENTRIES
EDIT,LIST, SAVE FILE
*ENTER*
M=MTM FILE
G=SIMOGRAM
.......
```

Step 3: Simogram Depressing G from the menu will allow us to sort the data by subjects and compute the cumulative time. For the above example we shall obtain the following.

```
    SUBJ. # 1
 ELEM.  TIME-TMU,D&C
 H32A   34  34
 H32B   48  82
TOT. TIME=  82


    SUBJ. # 2
 ELEM. TIME-TMU,D&C
TOT. TIME= 0

    SUBJ. # 3
 ELEM. TIME-TMU,D&C
 IDLE  30  30
 BEND  61  91
TOT. TIME= 91
```

* The analyst may wish to separate the printout according to subject and place them side by side to facilitate the construction of a simo-chart, a man-machine chart, or a gang chart.

* At the end of the program the computer will automatically return to the menu.

Step 4: Simulation

- Depressing U in the menu will lead us to the simulation program written in machine language.
- The program will ask "SPEED CHANGE(Y-N)?" Enter Y if a change in the speed (rate) of the simulation is going to be performed. The first time the data file is created the times recorded are the normal times obtained from the MTM-3 tables. If it is desired to slow down or speed up the simulation, this can be done by entering different speed ratings for each subject as shown below. If "N" is entered skip the next step.
- When Y is entered "ENTER SPEED COEFFICIENT FOR EACH SUBJ. (%)" is printed. After this, the display will show "SUBJ. # 1 % ?" four times corresponding to the four subjects capacity of the display. Enter the speed for each subject. For example, if 70 is entered for subject 1, the speed of the simulation will be 70% of the normal 100% speed. After the four speeds for each subject have been entered the program will modify the times of the data file to the correspondings values.
- Since this program does actually change the time data on file, it may be used to rate the observed time, or create a standard time data from the observed stopwatch time by entering the factor. (rate/1-allowance).

- If the standard time data are stored on tape, this program can be used to create the learner's standards by using the learning rate.
- To obtain the original data, a rated time must be multiplied by its reciprocal (e.g., an 80% file may be multiplied by 125% to regain the original data).
- The program will ask "PUSH O=ONCE C=CONT". Depressing the letter O Key will run the program through once and return it to the same message.
- Answering the program question with a C Key will continue the simulation until the silver RESET button is depressed.
- A typical cycle will be displayed as:

	DISPLAY	
Time (TMU)	CHANNEL 1	CHANNEL 3
0	H32A	IDLE
30	H32A	BEND
34	H32B	BEND
82		BEND
91	H32A	IDLE (repeating from time 0)
121	H32A	BEND

....

Step 5: Editing

- Interrupting the simulation by depressing the RESET button will return the computer to the AIM 65 monitor. Depressing T will then access the text editor.

- If the title TEST1 was entered in response to "TITLE(Y-N)?" and "ENTER MAX 15 CHRS", then the computer will display "TTEST1". The first T is to indicate that the record is to be a comment line.
- To turn the printer on, depress CTRL and the PRINT key simultaneously until the word "ON" appears at the end of the display.
- Depress the L key and then the period key (.) to the prompt "/". This indicates to the computer that (you want) the entire file is to be printed.
- Carriage return (depress the RETURN key) when asked "OUT=". This will start the printing if the printer is turned on, otherwise, the data will be flashed on the display.
- The data file created by our example will appear as follows:

```
/.
OUT=
TTEST1
E1 H32A ØØØØ34Ø
E3 IDLE ØØØØ3ØØ
E1 H32B ØØØØ48Ø
E3 BEND ØØØØ61Ø
```

- To change the data, enter C, then the string to change from, followed by the new string. For example, if we wish to change from H32B to H6A, we shall follow the steps:

  - /T enter T to move the cursor to the top of the editor file.

- TTEST1 will indicate that the cursor is at the top.
- Depress C, then type in "32B", depress RETURN.
- E1 H32B ØØØØ48Ø will be displayed to show that the computer has found the correct line.
- Depressing RETURN will display "TO=". Answer "6A ". then RETURN.
- E1 H6A ØØØØ48Ø will be displayed to show the correction.
- Note that the spacing is critical. Entering "6A" instead of "6A "will result in a misaligned data line: E1 H6A ØØØØ48Ø which is incorrect.

* Note that changing the element description in the editor will not automatically change the TMU value. This will have to be changed manually. FROM=48Ø TO=Ø7Ø

* When editing is completed, depress ESC, then depress N to return to the menu.

* Note that the MTM names may be changed to more descriptive terms after the data have been correctly identified. To run the simulation, select the subjects 1, 2, 3, or 4, when asked by the MTM data file creation program.

Step 6: Saving the File

* Steps 1 through 5 may be repeated as many times as needed by the analyst to perfect the method and timing.

- * To save the data file onto a casette tape, go to the editor by depressing T, then depress L, and then period under the slash. T in response to "OUT=", data file name (5 characters) in response to "F=", and 1 or 2 in response to "T=" to identify where the tape drive is connected. For example, OUT=T F=DATA1 T=1 will save the current content of the editor as a "DATA1" on drive 1.
- * To read into the editor the data stored on a casette, depress R and answer T to "IN=," the file name to "F=" and 1 or 2 to"T=". E.g., OUT=T F=DATA1 T=1 will read the file DATA1 from tape drive 1.

Example: Valve Assembly

To illustrate the entire process more succinctly, let us consider an analyst who is asked to study a valve assembly process. The operator uses his right and left hands, his body motion, and the press.

Method Description

Figure 11 shows the method description developed by the analyst. He can go to the AIM 65, enter the MTM program by depressing the N key, then the M key and then the RETURN key.

OPERATION: Valve Assembly

LEFT HAND (1)	MTM3 or TMU	RIGHT HAND (2)	MTM3 or TMU	BODY (3)	MTM3 or TMU	PRESS (4)	MTM3 or TMU
Move hand and pick up body	H32A	Move hand and pick up stem	H32A	Idle	34	Idle	34
Insert body in lubricant container Idle	H32A 14	Insert stem into the body	H32B	Idle	48	Idle	48
Transport to a washer	T6B	Transport to a washer	T6B	Idle	21	Idle	21
Hold body	34	Move the hose and wash body	H6B	Idle	34	Idle	34
Idle	Idle	Move body to assembly position	T32A	Idle	16	Idle	16
Pick up nut and position it on body	H32B	Pick up bonnet and position it on body	H32B	Bend	B	Idle	61
Idle	120	Idle	120	Step away from the machine	SF	Assembly valve	120
Idle	34	Move body to final box	H32A	Step back to machine Idle	SE 105	Idle	34

FIGURE 11. Valve Assembly Method Description

As he enters data, he can mix MTM-3 notations, such as T6B and H32A, with other MTM or non-MTM notations, such as R32A or HOLD. For those entries, for which there are no standard MTM-3 data, the analyst is asked to enter his estimates. Figure 12 shows the data file that resulted from entering data found in Figure 11.

Simogram

The analyst then asks for the simogram ("G") listing. The listing, as shown in Figure 13, differentiates each subject (1 for left-hand, 2 for right-hand, 3 for the body motion, and 4 for the machine).

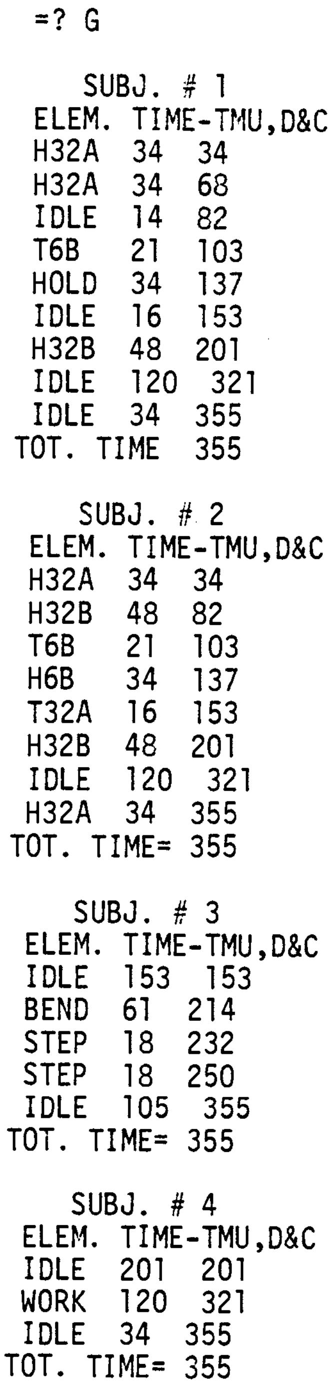
Simulation

The analyst then depresses U to simulate the data. He may, for example, take a videotape of the actual operation with the AIM 65 display also showing and analyze the discrepancies between the theoretical and observed times. Another application may be to train a worker in a new procedure or synchronize machineries to permit operator reaction time at critical moments.

Whenever it is necessary to add, delete, or change elements, the analyst calls out the Text Editor to assist in his revision work. He can also save the content of the current file onto a cassette tape for further analyses. Different simulations of the same data file at different speeds can also be performed.

```
OUT=
TVALVE A. 8881
E1 H32A 00000340
E2 H32A 00000340
E3 IDLE 00001530
E4 IDLE 00002010
E1 H32A 00000340
E1 IDLE 00000140
E2 H32B 00000480
E1 T6B  00000210
E2 T6B  00000210
E1 HOLD 00000340
E2 H6B  00000340
E1 IDLE 00000160
E2 T32A 00000160
E1 H32B 00000480
E2 H32B 00000480
E3 BEND 00000610
E1 IDLE 00001200
E2 IDLE 00001200
E3 STEP 00000180
E4 WORK 00001200
E1 IDLE 00000340
E2 H32A 00000340
E3 STEP 00000180
E3 IDLE 00001050
E4 IDLE 00000340
END
```

FIGURE 12. Editor listing of the valve assembly example data file.



```
 =? G

    SUBJ. # 1
 ELEM. TIME-TMU,D&C
 H32A  34  34
 H32A  34  68
 IDLE  14  82
 T6B   21  103
 HOLD  34  137
 IDLE  16  153
 H32B  48  201
 IDLE  120  321
 IDLE  34  355
TOT. TIME  355

    SUBJ. # 2
 ELEM. TIME-TMU,D&C
 H32A  34  34
 H32B  48  82
 T6B   21  103
 H6B   34  137
 T32A  16  153
 H32B  48  201
 IDLE  120  321
 H32A  34  355
TOT. TIME= 355

    SUBJ. # 3
 ELEM. TIME-TMU,D&C
 IDLE  153  153
 BEND  61  214
 STEP  18  232
 STEP  18  250
 IDLE  105  355
TOT. TIME= 355

    SUBJ. # 4
 ELEM. TIME-TMU,D&C
 IDLE  201  201
 WORK  120  321
 IDLE  34  355
TOT. TIME= 355
```

FIGURE 13. Simogram obtained from the valve assembly data file.

The AIM 65 proved to be an easy machine to work with. The development work for creating our own ROM was done also on the AIM 65 and the PROM programmer that can be attached to it. This low-cost PROM programmer also comes with an extended Code Editor (COED) which facilitates relocation of program addresses and editing of machine codes without having to return to an assembler.

The integrated approach was found to be flexible and applicable to a broad range of work measurement and design. Its application can be extended to cover queueing analysis models and direct creation of data file using the microcomputer as a multi-activity stopwatch.

The availability of expansion space for the program, locations D8E0 to DC45, means that we have 870 bytes of ROM storage available for additional features. Inclusion of MTM-1 or MTM-2 tables, learning curve computations, and regression curve analysis are among some of the features being considered.

The ability to save the data file is also an important attribute of this program. This frees the analyst from having to recreate similar data files from scratch. A new procedure that is similar to an existing procedure can be created rapidly by editing the old file and storing it under the new name. The data file can also be utilized later for learning analysis, productivity indices, operator training, new equipment justification, and method comparison.

The "showmanship" of the simulation program proved to have irresistible charm. It helps to "sell" the new method by showing in real-life how the actual timing will perform. Synchronizing machine cycles and determining the assembly line speed and line balancing are also made easier.

Having four channels to display during the simulation allows the simulation in real-time of two different MTM methods simultaneously. Since the Text Editor has the ability to read two or more data files and combine them into one buffer file, comparing two different methods becomes an easy task at making use of this feature. Also the simulation of any particular method can be executed at different speeds, under or above the normal pace.

The simulation also helps the worker become involved in the work design and analysis, and facilitates his/her making real contributions to methods design. Instead of being "observed," the worker can now be the observer of a new method and an astute critic of the new method being developed. The ease with which data can be changed, added, or deleted, makes the analyst's job more creative and productive.

Work Measurement and Microprocessors are cornerstones of productivity that can be joined together to build a solid foundation for a more prosperous and higher quality "tomorrow." This small project pointed out many advantages that can result from the marriage of the two.... and it is only a beginning.

III. DESIGN OF A UNIVERSAL TIME DATA STRUCTURE FOR WORK MEASUREMENT AND DESIGN

Rodolfo Blanco, Department of Industrial Engineering, Instituto Tecnologico de Costa Rica, Apartado 159, Cartago, Costa Rica.

Michael S. Inoue, Department of Industrial and General Engineering, Oregon State University, Corvallis, Oregon 97331.

Abstract

Recent emphasis on productivity has renewed interest in work measurement tools that provide time data essential in evaluating labor and equipment productivity. To enhance the productivity of Industrial Engineering time study activities, electronic stopwatches, terminal data input devices, and computer programs for synthetic times and statistical data processing are becoming increasingly more popular. Unfortunately, these tools are developed independently of each other and hamper productivity because of the lack of a common data base. The purpose of the research described in this paper was to design a Universal Time Data (UTD) structure based upon integration of desired IE functions. The design criteria included four hypotheses ranging from the use of a data structure on a low-cost single-board microcomputer with programs on limited size ROMs, to having all files accessible from the system's Text Editor. The prototype design on a 4K ROM to interface with a 6502 microprocessor based "Advanced Interactive Microcomputer (AIM 65)" is described. Proposed file structures, multi-activity data collection, text editing, and real-time simulation are discussed.

Industrial Engineers have always been concerned with productivity. Work Measurement is the basic tool used to evaluate labor and capital productivity. The traditional approach to Work Measurement is for the analyst to use his pencil and paper, stopwatch and/or standard data tables. With these he, observes the operator, equipment, film or video-tape and obtains time data about the operation. If the analyst wishes to make direct observations, he must endure many cycles of stopwatch readings and statistical computations to attain the confidence level and interval level required from the analysis. If the analyst wishes to use synthetic time data, he/she must look up elements in the table and hope that the synthesized operation nearly resembles the actual operation. If the analyst chooses to use work sampling, he/she is liable to become a slave to the clock and numerous rounds of data collection tours.

Ever since electronics has come of age, Industrial Engineers have been trying to use it to improve their productivity. The Electronic Time Recorder was introduced by Buffa in 1958 (7), and electronic stopwatches have largely replaced mechanical stopwatches among today's Industrial Engineers.

Many computer programs have been developed to perform table-lookup of snythetic data. ADAM (43), MOST (47), 4M (6), WOCOM (44), and Univation (6) are examples of such systems.

The drudgery of statistical analyses is largely eliminated through the use of hand-held calculators (16), and special programs on computers (12), (38). Finally, computer-aided work sampling (33) is coming to the aid of Industrial Engineers who are frustrated with the tedious tasks associated with work sampling studies.

The present state of the art lacks integration. First, it is observed that the degree of computerization varies greatly from one company to another. Second, there is no common data base structure that permits the analyst to "mix-and-match" data from different sources without having to hand rekey the data from one electronic device to another, or sometimes from one computer program to another computer program on the same computer. And, perhaps even more detrimental, is the fact that many computer programs run on COBOL or FORTRAN and require a fair-size computer that is costly and largely inaccessible to the analyst on a day to day basis.

Objectives of the Study

The main objective of the project described in this paper was to design a system for Work Measurement based upon a Universal Time Data structure using a low-cost microcomputer system.

More specifically, the study was to investigate four hypotheses:

(1) A Unified Data Base structure could be used to accommodate a variety of industrial needs.

(2) Data files created according to the unified time data structure could be accessed and edited by the analyst, much like the way office clerks use word processing machines.

(3) The programs needed could be written, compiled, and stored onto a limited size Read Only Memory (e.g., 4-K bytes ROM), so that the system is "turn-key" ready.

(4) The timer functions of microprocessors could be used, not only to measure and record time data, but also to real-time simulate activities recorded in the Random Access Memory (e.g., 4-K bytes ROM).

Finally, we wanted the entire system to be cost-effective so that most Industrial Engineering Department could justify its acquisition.

Industrial Surveys

The first hypothesis to be tested was that a Unified Data Base structure could be designed to accommodate a large variety of industrial needs. We began our study by identifying what such needs were.

A survey presented in Industrial Engineering in 1977 (32) showed that 95% of manufacturing companies surveyed used work measurement techniques. Out of these, 46% used stopwatches to establish time standards, 23% used a standard data method, and 12% used a predetermined time system. The survey also demonstrated how nearly half of the 1,500 respondents believed that there were not enough Industrial Engineers to expand applications of Work Measurement and wage incentives to any great extent.

In early 1981, we conducted a small survey of six companies in the Pacific Northwest. The study included FMC Corporation, OMARK Industries, Boeing Commercial Airplane Division, Tektronix, Freightliner, and Weyerhaeuser. The divisions surveyed ranged in employee size from 1,200 to 22,000 with an average of 7,950, and in number of Industrial Engineers from 2 to 300, with an average of 65.

TABLE 1. Pacific NW Survey 1981

		Manual		
DL=Direct Labor			Computerized	
IL=Industrial Labor	Not		Manual &	Computer
Subject	Used	Manual Only	Computer	Only
Stopwatch (DL)	1	3	1	1
" (Equipment)	1	3	1	1
Work Sample (DL)	0	4	1	1
" (IL)	3	2	0	1
Predetermined (DL)	3	2	0	1
" (IL)	4	1	0	1
Std. Data System	3	1	0	1
Estimating & Costing	1	1	3	1
Production Scheduling	1	2	2	1
Equipment Utilization	1	3	1	1
Justify NC Machines	2	2	1	1
" Robots	4	0	1	1

Table I illustrates that stopwatch and work sampling still remain popular methods of time measurement. Even those who did not currently use computerized methods, indicated that these two methods could be made more useful through computerization.

In the areas of application, it is interesting to note that companies that perform robotization studies do not attempt to do so without the help of a computer. Other comments received during the study indicated that for each hour of time study, the analyst spent an average of one hour of desk work to analyze the results. The need was felt for a multielement memory stopwatch that would eliminate much of the desk work and permit the combined recording of piece counts and production time data. Other data from the survey revealed that the most popular manual predetermined time system appeared to be MTM-based.

Electronic data gathering systems included DATAMYTE, Uptime-meter, RADAC, and minicomputers. Presently, other devices available include Downtimer and a printing electronic stopwatch "Time Gun" used by Dr. P. Ramalingam in the I.E. Laboratory of the California State Polytechnic University in Pomona, California.

Though the survey results seemed to indicate a wide variety of work measurement applications to exist, the essential data appear to be in most cases limited to:

(1) Identification of the observed subject (e.g., right-hand, machine #1, etc.).

(2) Description of the time element (e.g., R12A, MOVE, IDLE, WAIT, etc.).

(3) Time data, either elemental or cumulative, and no finer than MU=1/10 of TMU or one millionth of an hour.

It was decided that the very variety of electronic devices and computer programs already in use in many companies, and their non-standard data formats, did justify our continuing the research toward a standard data base format compatible to most work measurement applications.

Time Data Format

To evaluate the second hypothesis that the data files created using the universal time data base could be accessed and edited easily by the analyst and also text edited, we started by studying the common data structure encountered in both the stopwatch and MTM methods as shown in Figure 14. This common data structure, as identified in our first hypothesis, is composed of three basic parts: 1) Subject that executes the element (who?), 2) Element description (what?), and 3) Element time (how long?).

1	2	3
Who?	What?	How Long?

1. The subject that executes the element
2. The element description
3. The amount of time it takes to execute the element.

FIGURE 14. Common data structure in both the MTM and stopwatch methods.

1 - The subject that executes the element is basically the operator and/or machine in the stopwatch method or a part of the body (left hand, right hand, legs) in the motion time measurement (MTM) method.

2 - The element description is the alphanumeric character string used to describe the element or micromovement executed in any operation. The MTM method uses a symbol code to describe a micromovement. For instance, R32A means the operator is to reach a distance of 32 inches using a case A type of reach movement. Case A reach is one of four reach micromovement cases. It means that the operator is to reach an object in a fixed location or an object in the opposite hand. For an explicit description of MTM, see (26) or the Motion and Time Study references.

In stopwatch studies, the element description is basically brief, usually not more than six words, and explains what the operator or machine does. Often, the description can be abbreviated by using four character words or initials (e.g., WORK, IDLE, WAIT, ASM5, SUB3, etc.).

Using the programs described in this paper, a description can be entered as one or more title lines by preceding it with the character "T" or by entering a maximum of four characters in the element description positions as described above and elaborated on later.

3 - The time unit used with the MTM method is the TMU (Time Measurement Unit). This unit is equivalent to one hundred thousandth

of an hour (1/100,000 hr.). This time unit is used to define the MTM micromovement time whose duration varies from 1.6 to 76.7 TMUs. Some other predetermined time systems use the MU which is equivalent to one millionth of an hour (1/1,000,000 hr. = 1/10TMU) (3). Once the normal time is calculated it is often translated to minutes or hundredths of a minute before implementation. The time units used with stopwatch studies are basically the minute or the hundredth of a minute. The reason for this is that the elements evaluated using the stopwatch method are longer than those used with the MTM method.

For both methods the time unit that seems most appropriate to use as a standard time unit is the MU. There are two reasons why this is so.

a) The best precision is obtained. Table 2 shows the maximum errors obtained from three different time units. The maximum error is obtained when the key is depressed one cycle before the clock finishes counting the current time unit. For example, when counting MU time units the microcomputer's time is loaded with 3598 cycles which corresponds to a MU time unit. The maximum error will occur when 3597 cycles, which is equivalent to 0.000000999 hours, are counted. The timers will be explained more specifically later on in this paper.

Table 2. Maximum Errors

	MU	TMU	1/16 SEC.
Maximum Error (hours)	$9.991*10^{-7}$	$9.999*10^{-6}$	$1.736*10^{-5}$

b) MTM contains tenths of TMU as a normal time. For example, R28A has a normal time of 16.7 TMUs or 167 MUs. Using this time unit makes it easier to take stopwatch observations of MTM movements.

To satisfy the second hypothesis, we need a data structure that is compatible with the data editing capability that is commonly available on computers. The ID and the time data could be represented either numerically in a binary or packed-decimal form, or alphanumerically as we store alphabets and symbols by assigning binary codes to each letter. The former methods, especially the binary method, are more efficient in conserving the memory space required to store high precision numbers in small number of bits. The packed decimal, a method pioneered by IBM during the 1960s, packs two numbers in one byte.

Highest number that can be represented by one byte (8 bits):

binary	$2^8 = 256_{10}$	stored as $1111\ 1111_2$
packed decimal	99_{10}	stored as $1001\ 1001_2$
alphanumeric (ASCII)	9_{10}	stored as $0011\ 1001_2$

The description of the observation requires that the data are in alphanumeric form to allow MTM nomenclature and English wording. Similarly, it will be advantageous to have the title and remarks stored alphanumerically rather than numerically only. Having numeric and alphanumeric formats mixed in a data file means that we need two different editing schemes, one for numbers and another for descriptions. This would make the system less flexible and harder to learn to operate. Also, there is a universally adopted convention to represent alphanumeric data, the American Standard Codes for Information Interchange (ASCII). There is no uniformly adopted method for binary storage of numerical data on all computers.

For these reasons, it was decided to represent all data in the Universal Time Data structure to be in ASCII. Figure 16 illustrates a UTD structure in 17 bytes (17x8=136 bits) constituting one time data record.

Microcomputer Structure

Before discussing the data base structure further it is necessary to describe the microcomputer's features (35) (37). The computer we have chosen for our implementation is an Advanced Interactive Microcomputer, AIM 65, manufactured by Rockwell International. The AIM 65 microcomputer uses a 8 bit 6502 microprocessor as a CPU operating a 1 MHz which is equivalent to a million cycles per second. This provides a minimum execution time of two microseconds or two cycles. In other words, 200,000 to 500,000 machine language instructions may be executed in one second. Thirteen addressing modes and 56 operation codes provide a great deal of flexibility in programming. In addition, a symbolic assembler, a fully relocatable assembler, and a BASIC interpreter can be used to write programs.

The microcomputer board also has a Versatile Interface Adapter (VIA) which contains two eight bit timers, two eight bit bi-directional ports, four control lines, and a shift register. This Versatile Interface Adapter can be used to interface external circuitry and equipment.

The internal monitor, stored in 8K ROM (Read Only Memory), is used to control the operation of the AIM 65. This monitor can be used to 1) alter memory positions and registers, 2) enter instructions in assembler format and translate them to object code, 3) obtain the assembler codes of the instructions stored as object codes, 4) execute the user programs, and 5) load, dump and save on cassette tape the object code and the assembler source code.

This internal monitor also has a built-in Text Editor which can be used to delete, insert, and change data or list and save data files. This Text Editor allows only ASCII codes to be used. The ASCII code defines the binary values for all the alphanumeric characters of the keyboard. For example, the letter A in ASCII is equivalent to 01000001 in binary, 41 in hexadecimal, and 65 in decimal. In this system seven bits of each byte are used for the representation of the character, leaving the last bit to be used for verification purposes when needed.

The 20 column thermal paper onboard printer provides a useful means of obtaining a printout of the results immediately after the execution of the programs. This provides a permanent record of the input data, commands used, and data obtained. The speed of this printer is 120 lines per minute and it prints the complete ASCII's 64 characters.

The onboard 20 character display provides visual feedback during keyboard operations. The keyboard module contains a 54 key ASCII code full size keyboard. The keyboard has 70 functions (26 alphabetic, 10 numeric, 22 special, 9 control, and 3 user defined) used by the AIM 65.

The microcomputer also operates by using a memory map addressing mode which provides a flexible capability for memory expansion. Figure 15 shows the AIM 65 microcomputer.

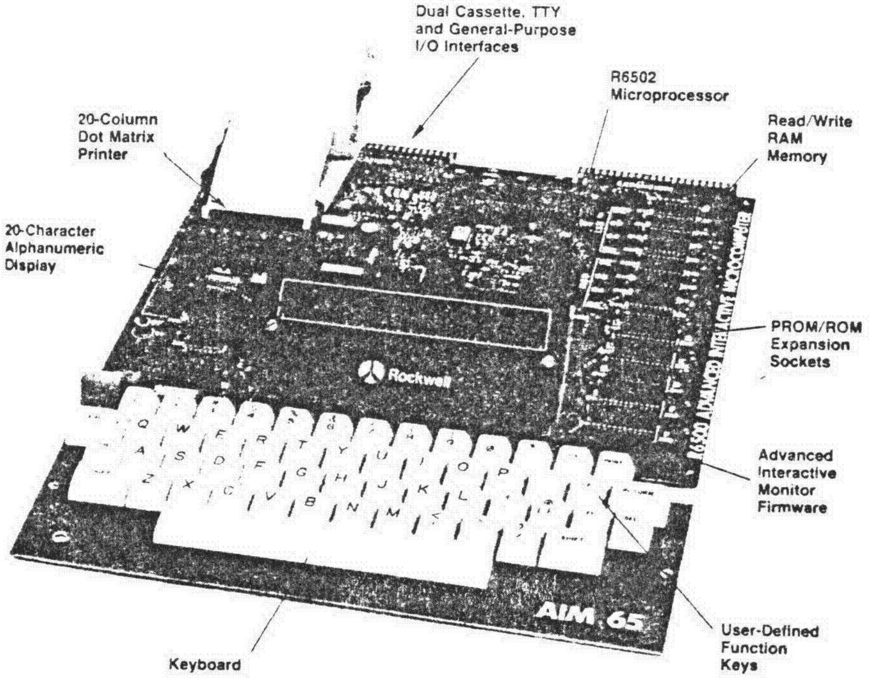

Dual Cassette, TTY and General-Purpose I/O Interfaces
R6502 Microprocessor
20-Column Dot Matrix Printer
Read/Write RAM Memory
20-Character Alphanumeric Display
PROM/ROM Expansion Sockets
Rockwell
Advanced Interactive Monitor Firmware
AIM 65
User-Defined Function Keys
Keyboard

FIGURE 15. AIM 65 Microcomputer

Universal Time Data Structure on the AIM 65

The 4K RAM (Random Access Memory) provides the on board AIM 65 microcomputer with 3584 bytes of memory available to store data and/or programs.

To define the number of bytes assigned to each part of the data structure, the memory capacity, the display capacity, the text editor structure, the duration of elements, and the element description were taken into consideration. Figure 16 shows the universal Time Data structure.

The first byte is used for the initial code. The initial code is used when data files are created from the field using the stopwatch method. This initial byte designates when an element starts, ends, and whether or not it is a foreign element.

The second and third ASCII bytes numerically designate the subject that executes the element. This gives the analyst the capacity to define up to 99 different subjects. When implementing the MTM method these two bytes can be used to define the left hand, right hand, body, or machine involved in the analysis. When using the stopwatch method they can be used to designate which operator or machine is doing the element.

The fourth, fifth, sixth, and seventh bytes are used to store the description of the element. Four bytes are assigned to this task so that the simulator program can display four channels simultaneously. The AIM 65 display has a capacity of 20 characters.

One Time Record

Byte Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Data Elements	a) I D	b) Who?		c) What?				d)	e) How Long?								f) E O R

S = Start of Element

E = End of Element

C = End/start of Element

F = End/start of Foreign Element

T = Comment line (Ignored by the programs)

ID a) Initial Code (byte 1)

WHO-b) Subject that executes the operation in 2 bytes (bytes 2+3)

WHAT-c) Element description and/or element number in 4 bytes (bytes 4-7)

d) Expansion of element description of any other special character (byte 8)

HOW LONG-e) Element time (MUs) (bytes 9 through 16)

EOR-f) End of record code (used by the text editor) (byte 16) Carriage return (OD_{rec})

Designed standard data base structure for both MTM and stopwatch methods

FIGURE 16. Universal Time Data (UTD) format.

This provides enough space to display four characters in each channel with a space in between. The four characters provide enough space to display the MTM symbols for the micromovements and also a four character condensation of the element description when using the stopwatch method. (See Figure 26).

The eighth byte is not used, although, it could be used to expand the element description when a larger display is available.

The ninth through the sixteenth bytes are used to store the element time. Eight bytes are assigned to this task since they provide a reasonable maximum element duration of 99.99 hours which is adequate for most industrial situations. See Table 3.

Table 3. Maximum element duration from 1 byte to 10 bytes.

Bytes	Maximum	Element	Duration
	MU's	Minutes	Hours
1	9	0.00054	0.000009
2	99	0.00594	0.000099
3	999	0.05994	0.000999
4	9999	0.59994	0.00999
5	99999	5.99994	0.0999
6	999999	59.9999	0.999
7	9999999	599.9999	9.999
8	99999999	5999.999	99.999
9	999999999	59999.999	999.999
10	9999999999	599999.999	9999.999

The internal clock is kept in ASCII code. One reason for using the ASCII code is that the internal Text Editor used in the programs works only with ASCII codes and not with binary or decimal numbers. This means that eight bits are used to store a number between one and nine. As stated earlier, this feature does not make efficient use of the memory capacity. It can be seen in Table 4 that ASCII codes make very poor use of the memory capacity as far as numerical representation is concerned.

Table 4. Capacity used by three different systems with eight bytes available

	Binary	Packed Decimal	ASCII
8 bytes	$0\text{-}1.8446744*10^{10}$	0-9999999999999999	0-99999999
% of Usage	100%	0.054210%	$5.4210*10^{-10}\%$

The last byte of each observation is used for the end of the line code. This byte is primarily used by the Text Editor. When the user is editing the data file, the Text Editor displays the ASCII characters until it finds the end of the line code. This gives the user the ability to edit the entire data file, observation by observation. The user can make changes on each observation, delete or insert new lines, list the data file or save it on a cassette tape.

Table 5 shows the commands used by the AIM 65 Text Editor and Figure 17 shows an example of editing a data file using the Text

Editor. This demonstrates that the analyst can access the data files created and then edit them as stated in the second hypothesis.

Table 5. AIM 65 Test Editor Commands

CATEGORY	COMMAND	FUNCTION
Editor Entry and Exit	E	Enter and Initialize Editor
	T	Re-Enter Editor
	Q	Quit Editor and Re-Enter Monitor
	RESET	Enter and Initialize Monitor
	ESC	Re-Enter Monitor
Text Input/ Output and Update	R	Read Into Text Buffer
	L	List From Text Buffer
	I	Insert One Line
	K	Delete One Line
Line Pointer Positioning and Display	T	Move the Line Pointer to the Top
	B	Move the Line Pointer to the Bottom
	U	Move the Line Pointer Up One
	D	Move the Line Pointer Down One
	SPACE	Display Current Line
Character String	F	Find Character String
	C	Change Character String

```
(T)                          Reenter Text Editor
TFIRST DATA FILE
=(L)                         List
/.                           entire file.
OUT=CR
TFIRST DATA FILE
E1 H32A ØØØØ034Ø
E2 H32A ØØØØ034Ø
E3 IDLE ØØØØ09ØØ
E4 IDLE ØØØØ078Ø
E1 T32B ØØØØ029Ø
E2 T32B ØØØØ029Ø

(T)                          Go to the top of the file
TFIRST DATA FILE             Change:  FIRST
=(C)
FIRST
TFIRST DATA FILE             to MTM3.
TO=MTM3
T MTM3 DATA FILE             Find IDLE.
=(F)
IDLE
E3 IDLE ØØØØ09ØØ
= (C)                        Change IDLE.
IDLE
E3 IDLE ØØØØ09ØØ             to WORK.
TO=WORK
E3 WORK ØØØØ09ØØ
=(D)                         Move down one line.
E4 IDLE ØØØØ078Ø             Delete that line.
=(D)
E4 IDLE ØØØØ078Ø
E1 T32B ØØØØ029Ø

=(T)                         Go to the top.
T MTM3 DATA FILE
=(L)
/.                           List the
OUT=CR                       entire file once again.
T MTM3 DATA FILE
E1 H32A ØØØØ034Ø
E2 H32A ØØØØ034Ø
E3 WORK ØØØØ09ØØ
E1 T32B ØØØØ029Ø
E2 T32B ØØØØ029Ø             ESCAPE.  Send control to the monitor.
END ESC
```

FIGURE 17. Example of the Text Editor Usage.
(User entries are underlined)

Data Flow

The third hypothesis states that the programs created can be written, compiled, and stored in a 4K byte ROM (Read Only Memory) so that the computer is ready whenever it is turned on. In order to validate this, we first had to design the data flow logic and the specification of each program. Figure 18 shows the data logic for the work measurement studies used by the system proposed in this paper.

A. Data File Creation

The use of the approach presented in this paper begins by using the stopwatch program to create an initial data file from the field. The Industrial Engineer simply needs to enter the observations as they occur, using the microcomputer keyboard. For each observation that the Industrial Engineer enters, a new line is added to the data file. This new line is composed of the initial code, the subject that does the operation, the element number, and the element's continuous time.

The continuous time is automatically recorded by the stopwatch program clock which runs continuously instead of resetting to zero each time an element finishes. Using this continuous method (29), each time an element finishes, the current time of the stopwatch is recorded as the continuous time for that observation. When all the observations are taken, the discrete time for each observation is calculated by subtracting the element start time from the time

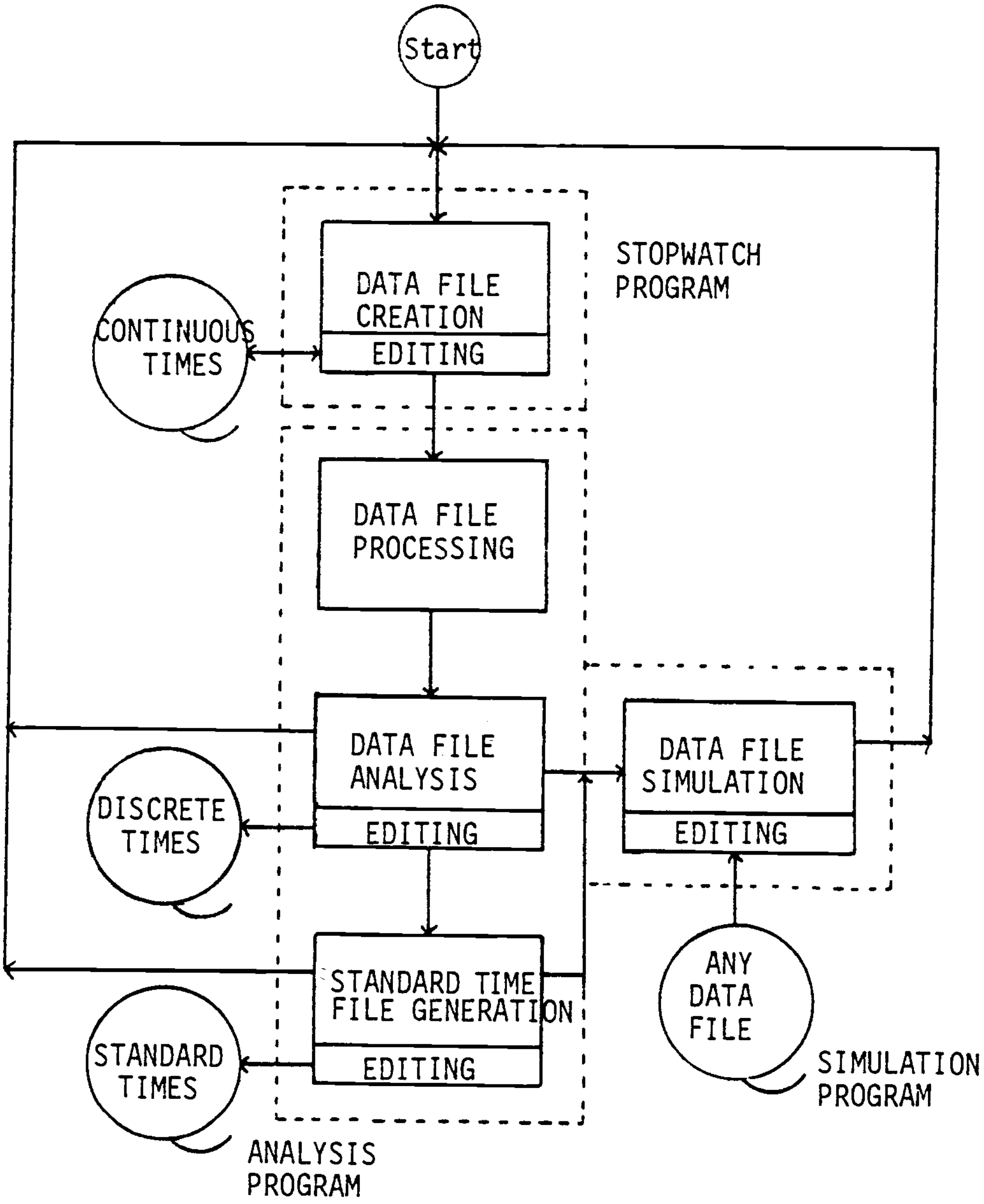


FIGURE 18. Flow chart showing the relationship between each program and their data file functions.

recorded when the element is finished. Upon recording the last observation at the end of the study, the overall time spent in the field is obtained automatically.

The RAM memory available on the microcomputer was designed to have the capacity of 90 lines of 17 bytes each which can be stored as the initial data file. Figure 19 illustrates an initial data file obtained from the field using the stopwatch program.

When the user decides not to make any further observations or when the memory is full, the program automatically prepares the Text Editor. Then, the user can edit, add a title line and comments, and then list or save on a casette tape the initial data file. This initial data file is not usually edited because within the stopwatch program deleting and correcting observations is easily performed.

B. Data Processing and Analysis

Data file processing is achieved through an analysis program. This program basically does the necessary calculations to convert the continuous times to discrete times for each observation recorded in the initial data file. The program calculates the discrete time, outputs it to the printer, and changes it in the data file. For most analyses, it is more convenient to have the discrete time for each observation instead of the continuous time.

DATA FILE	DESCRIPTION	
OUT=		
S1 1 00005418	Start of element 1.	
C1 1 00008140	End of element 1 and start of next element	
C1 2 00009493	End of element 2 and start of next element	
C1 3 00013665	End of element 3 and start of next element	
C1 1 00016541	End of element 1 and start of next element	
C1 2 00017954	End of element 2 and start of next element	
C1 3 00022028	End of element 3 and start of next element	
C1 1 00024809	End of element 1 and start of next element	
C1 2 00026179	End of element 2 and start of next element	
C1 3 00030393	End of element 3 and start of next element	
C1 1 00033267	End of element 1 and start of next element	
C1 2 00034667	End of element 2 and start of next element	
C1 3 00038784	End of element 3 and start of next element	
C1 1 00041529	End of element 1 and start of next element	
C1 2 00042904	End of element 2 and start of next element	
F1 3 00044283	Start of foreign element	Inside element 3
F1 00045660	End of foreign element	
C1 3 00048484	End of element 3	
END		

FIGURE 19. Example of an Initial Data File

After the data file has been changed to discrete time, calculations of the average time, the standard deviation, and the number of observations required for precision desired are obtained. A printed output of all these results is then made available.

The following formula is used to obtain the average time (21).

$$A_i = \frac{\sum_{j=1}^{N_i} X_{ij}}{N_i}$$

i = Number of elements.
j = Observation number.
A_i = Average time of element
X_{ij} = Observation time j for element i.
N_i = Total number of observations for element i.

The following formula is used to obtain an unbiased standard deviation (21).

$$S_i = \sqrt{\frac{\sum_{j=1}^{N_i} X_{ij}^2 - (\sum_{j=1}^{N_i} X_{ij})^2 / N_i}{N_i - 1}}$$

S_i = Standard deviation of element i.

$\sum_{j=1}^{N_i} X_{ij}^2$ = Summation of the squares of the observed times for element i.

$(\sum_{j=1}^{N_i} X_{ij})^2$ = Square of the summation of the observed times for element i.

N_i = Number of observations for element i.

To calculate the number of observations required for each element, 95% confidence level is kept constant while the precision is entered into the program as a variable.

The formula used is (21):

$$NT_i = \left(\frac{\frac{2}{P/100} N_i \sqrt{N_i S_i}}{\sum_{j=i}^{N_i} X_{ij}} \right)^2$$

Where:

NTi = Number of observations required to take for element i.
P = Precision desired (%)
Ni = Number of observations already taken for element i.
Si = Standard deviation for element i as defined above.
X_{ij} = As defined above.

All the observations previously taken are used to calculate the number of observations required to meet the desired precision. In order to obtain an average time more representative of the real situation happening in the field, all the element observations must be included (28).

Figure 20 shows an example of the statistical analysis obtained from the data file.

Following the scheme shown in Figure 5, the Industrial Engineer has several ways in which to continue once the initial data file has been analyzed. a) A new initial data file can be created, b) the discrete times data file obtained after the analysis can be edited, c) a standard times data file can be generated, or d) any data file can be simulated.

If the number of observations required to meet the precision is less than the number of observations already taken for the elements under study, the user should continue taking observations. In this situation the Engineer would have to return to the stopwatch program in order to create a new initial data file. This new initial data file would contain new observations which would be added later to the existing results already in the microprocessor memory. Once the new data file is completed, the processing and analyzing steps are repeated until the number of observations required are satisfied.

C. Data File Editing

The Engineer may be interested in editing the new discrete time data file in order to add a title line at the top of the file, to list the file, or to save in a cassette tape the data file for later analysis or simulation. This can be done by using the Text Editor commands as shown in Table 5.

```
(N)
# ELEMENTS? 3
PRECISION(%)? 5

TIME UNIT=1/100 MIN.


    *DATA FILE*
ELE#  TIME   OB.#
 1  16.332  ( 1 )
 2  8.118   ( 1 )
 3  25.032  ( 1 )
 1  17.256  ( 2 )
 2  8.478   ( 2 )
 3  24.444  ( 2 )
 1  16.686  ( 3 )
 2  8.22    ( 3 )
 3  25.284  ( 3 )
 1  17.244  ( 4 )
 2  8.4     ( 4 )
 3  24.702  ( 4 )
 1  16.47   ( 5 )
 2  8.25    ( 5 )
F  8.262    ( 1 )
 3  25.218  ( 5 )

     SUMMARY

OBS. TIME= 290.904

ELE#  AVG. T & STD
 1  16.7976  .43
 2  8.2932  .14
 3  24.936  .35
F.  8.262

ELE# OB.NOW OB.REQ
 1     5     2
 2     5     1
 3     5     1
F.     1
```

FIGURE 20. Example of the analysis program output using the initial data file of Figure 18.

D. Standard Time File Generation

When the Engineer is satisfied with the number of observations taken of all the elements, a new data file can be created using the analysis program. This new data file contains the standard times for all the elements studied and is created interactively with the microcomputer. A printout of the rating and allowances entered for each element is also obtained. Figure 21 shows an example of the printout and data file obtained using this analysis program.

For each element, the rating and allowances are asked to be entered into the microcomputer. Once these two variables are entered, the program uses the following formula to calculate the standard time (16):

$$STi = \frac{OT_i * (Ri/100)}{1 - Ti/100}$$

Where:
ST_i = Standard time of element i. (1/100 min.)
OT_i = Observed time of element i.
Ri = Rating of element i (%).
Ti = Allowance for element i (%).

The standard time for the entire operation is calculated by summing the standard times of all the elements. Because decimal minute (1/100 minute) is the most commonly used time unit, the observed times, average times, and standard times are printed using it instead of MU's.

```
EL. RTG ALW STD. TIME
 1  100  9
 18.4589011
 2  120  10
 11.0576
 3  90   5         Output from the
 23.623579         analysis program
TOT TIME=
 53.1400801


 FILE CREATED
```

```
(T)                Reenter the Text Editor
E1  1   00003076
=(I)               Insertion of the
T STANDARD TIMES   title line
E1  1   00003076
=(T)               Top of the line
T STANDARD TIMES
=(L)               List
/.                 entire file
OUT=
T STANDARD TIMES
E1  1   00003076
E1  2   00001842   New standard time
E1  3   00003937   data file
END
```

FIGURE 21. Example of the standard time generation output and the insertion of a title line on the standard time data file generated using the analysis program.

E. Real Time Simulation

The main objective most common to the Industrial Engineer is to obtain the standard time of the operation. In the most common cases the stopwatch and analysis programs are enough to meet this goal quickly, reliably, and at low cost. But when using work design, evaluating new methods, training employees, implementing and verifying new improved methods, and evaluating employee speed, a four channel real time simulator program would be useful to efficiently achieve these tasks.

The four channel simulation program displays a four character element description during the real discrete time defined in the data file. This is done for a maximum of four subjects. This means that the element description of the four subjects can be displayed simultaneously on the 20 character AIM 65 display.

The number stored in the two-byte position designed for the subject that executes the element is used to define which of the four channels will be employed. This indicates that if the number one appears in that position, channel one will be used to display all the elements executed by subject one. This allows a simulation of up to four machines and/or operators. For instance, one operator can be simulated with three other machines, or two operators can be simulated with two machines, or four operators can be simulated simultaneously, etc. After the simulation has been completed, any data file can be created, edited, or loaded from the cassette tape in order to follow the entire data flow process again.

Programs Logic

A description of each program is included in this discussion.

A. Stopwatch Program Logic

The purpose of this program is to create an initial data file containing the observations taken in the field. The program algorithm consists of a main program and an interrupt subroutine. The algorithm of the main program begins by preparing the Text Editor to accept data observations created from the keyboard. It then initializes the timer number one of the VIA that is used for the clock. Immediately following this, a message is displayed to inform the user that the program is now ready to accept data. The analyst simply enters the element number in the display and depresses the key corresponding to the beginning or the end of the element. The program then continuously executes the instructions corresponding to the key depressed.

The interrupt subroutine algorithm only increases the clock memory positions by one each time it is executed. Figure 22 presents a general flowchart of the main program and interrupt subroutine.

A more explicit description of the main program is as follows. After the element number has been entered in the display, the S, E, C, or F keys can be depressed in order to create a new observation in the data file. To create this new observation the letter of the key depressed is stored as the initial code. The

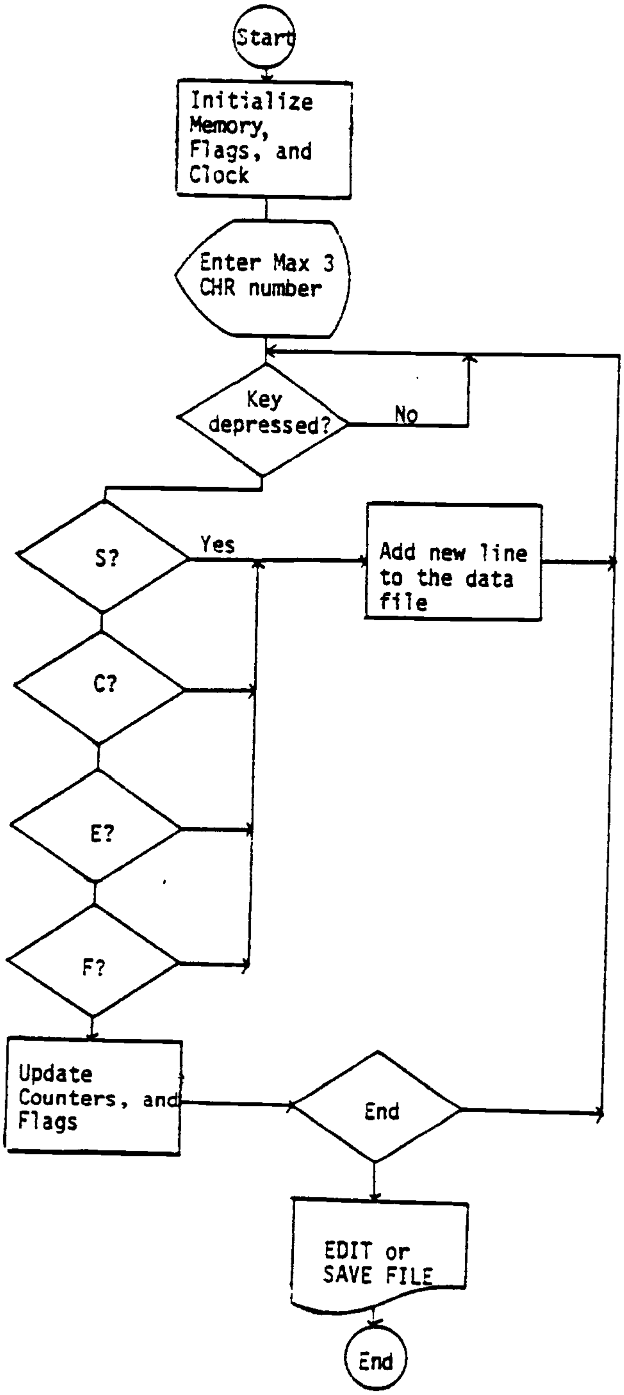

Start
Initialize Memory, Flags, and Clock
Enter Max 3 CHR number
Key depressed?
No
S?
Yes
Add new line to the data file
C?
E?
F?
Update Counters, and Flags
End
EDIT or SAVE FILE
End

FIGURE 22. Stopwatch program flowchart.

number 1 is stored as the subject that executes the operation. The element number entered is stored as the element number and the current time from the clock is stored in the memory position designated for the element time. The program then stores the end of the line code and prepares the memory pointers for the next observation. When K is depressed the program sets the current memory pointer back one line. This is done in order to begin the following line in the same memory position as the previous one. This K key is used to erase the last observation taken.

In order to erase an element number that has been entered, the analyst depresses the DEL key. When this happens the program sets the memory pointers back to the beginning of the data line. This is done in order to begin entering the correct element numbers in the initial positions once again. It also erases from the display the element number entered beforehand.

When the RETURN key is depressed or when the memory available for the data file storage is full, the program calculates and prepares all the memory positions used by the Text Editor. This is done so that the user will be able to edit the data file. The program sets the necessary memory positions used by the BASIC analysis program so that the user has the option of adding the results of previous data files to the analysis of the current data file.

The RAM memory assigned to the initial data file created by this program, allows a capacity of 90 lines or observations to be entered. Figure 23 shows the input/process/output functions of this program.

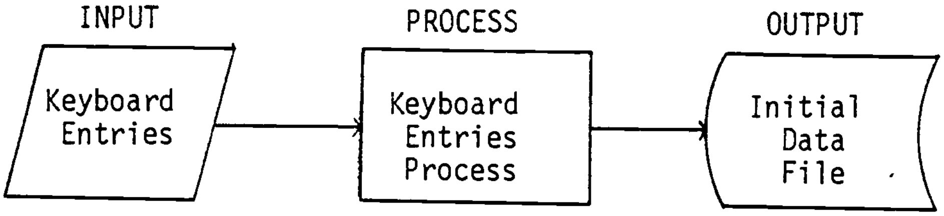


FIGURE 23. Input/Process/Output Functions of the Stopwatch Program.

B. Analysis Program Logic

The purpose of this program is to analyze the data files created by the stopwatch program and to create a new data file containing the standard times for the operation under study.

The program algorithm consists of three sections. The first section calculates and prints out the discrete times for all the observations of the data file. It also changes the continuous times in the initial data file to discrete times. The second section of the program calculates and prints the average time, the standard deviation, the number of observations taken, and the number of observations required for all the elements. The third section creates a new data file containing the standard time of the elements. Figure 24 shows the flowchart of the program.

More specifically, the program begins by asking the Industrial Engineer to enter the number of elements to be analyzed and the

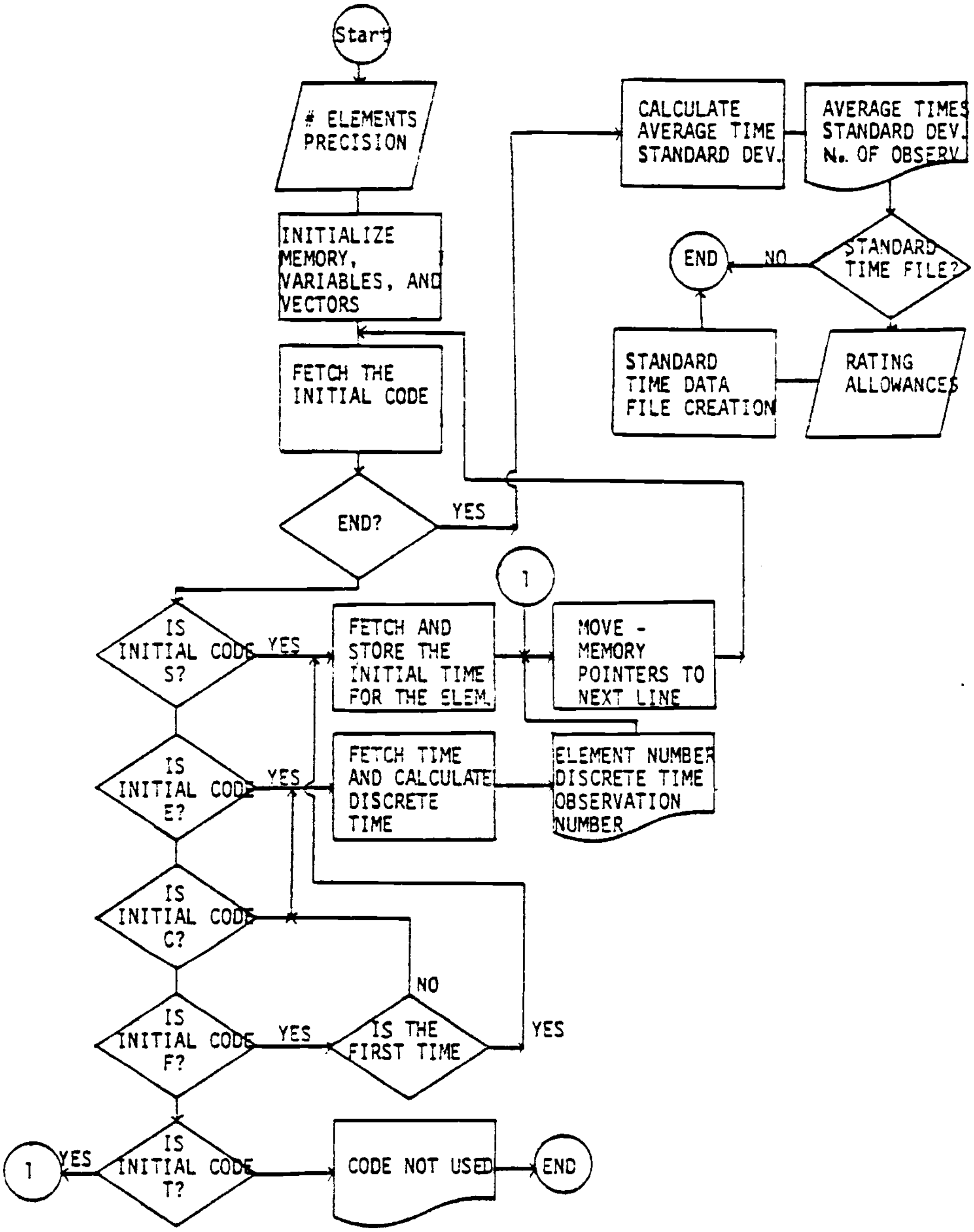

Start
# ELEMENTS PRECISION
INITIALIZE MEMORY, VARIABLES, AND VECTORS
FETCH THE INITIAL CODE
END?
YES
CALCULATE AVERAGE TIME STANDARD DEV.
AVERAGE TIMES STANDARD DEV. N. OF OBSERV.
STANDARD TIME FILE?
NO
END
STANDARD TIME DATA FILE CREATION
RATING ALLOWANCES
IS INITIAL CODE S?
YES
FETCH AND STORE THE INITIAL TIME FOR THE ELEM.
1
MOVE - MEMORY POINTERS TO NEXT LINE
IS INITIAL CODE E?
YES
FETCH TIME AND CALCULATE DISCRETE TIME
ELEMENT NUMBER DISCRETE TIME OBSERVATION NUMBER
IS INITIAL CODE C?
NO
IS INITIAL CODE F?
YES
IS THE FIRST TIME
YES
1
YES
IS INITIAL CODE T?
CODE NOT USED
END

FIGURE 24. Analysis program flowchart.

percentage of precision desired for the study. After this step is completed the program prints "TIME UNIT = 1/100 MIN". This time unit is used to display or print the observed times, average times and standard times throughout the entire program. Next, the program calculates and prints the discrete time and the number for each observation of the data file. This is done according to the initial code. For example, when the letter S appears as the initial code in any line of the data file, the time recorded in the observation is taken as the initial time for the element. But when the letter E appears as the initial code, the discrete time for that element is calculated and the corresponding change is made in the data file. The Industrial Engineer may turn off the printer before the analysis begins if a printout of the analysis is not desired. The program then displays and prints the total time spent in the field taking observations.

Once the program finishes this first analysis, it calculates the average time, standard deviation, the number of observations taken, and the number of observations required for each element. It then asks the Industrial Engineer if a standard time data file is going to be created. If the response is negative the program terminates. If the response is positive the program clears the memory positions needed for this new data file. It then asks the Engineer to enter the ratings and the allowances for each element. These ratings and allowances are used to calculate the standard time and a printout of them is obtained. The program then calculates and prints the total time for the operation and displays

"FILE CREATED" as a means of informing the Engineer that the standard data file is completed.

The RAM memory assigned to the variables of this program allows a maximum of 39 elements to be analyzed. Figure 25 shows the input/process/output of the program.

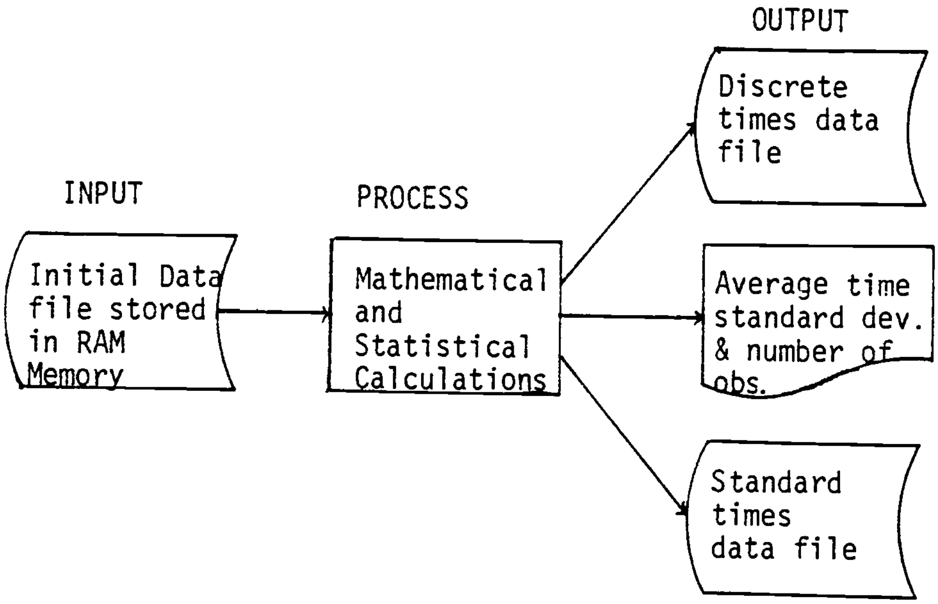

OUTPUT
INPUT
PROCESS
Initial Data file stored in RAM Memory
Mathematical and Statistical Calculations
Discrete times data file
Average time standard dev. & number of obs.
Standard times data file

FIGURE 25. Input/Process/Output Functions of the Analysis Program.

C. Simulation Program Logic

The purpose of the simulation program is to simulate in real time the elements described in any data file created or modified by the analysis program or by the Text Editor.

The program algorithm simulates up to four channels. These four channels are assigned to the four subjects, number one through number four, (i.e., operator, machine, left hand, right hand, etc.). For instance, subject number one is displayed in channel number one and so forth. See Figure 26 for an explanation of the display positions. Note: There are four characters for each channel and each is separated by a space.

The program algorithm consists of two sections. The first section begins by preparing the memory pointers, memory locations, and flags needed to execute the program. Next, the timer that runs the four clocks and the timer that sends the signals to the speaker are initialized. Each channel has a separate clock. The program then asks the Industrial Engineer to state his preference for continuous or single shot simulation by depressing the corresponding key. Once the key has been depressed the program begins the simulation by displaying the corresponding element description for each subject. The program uses the number stored in the memory positions designated for the subject that executes the element (see Figure 16), to define in which of the four channels the element description is to be displayed.

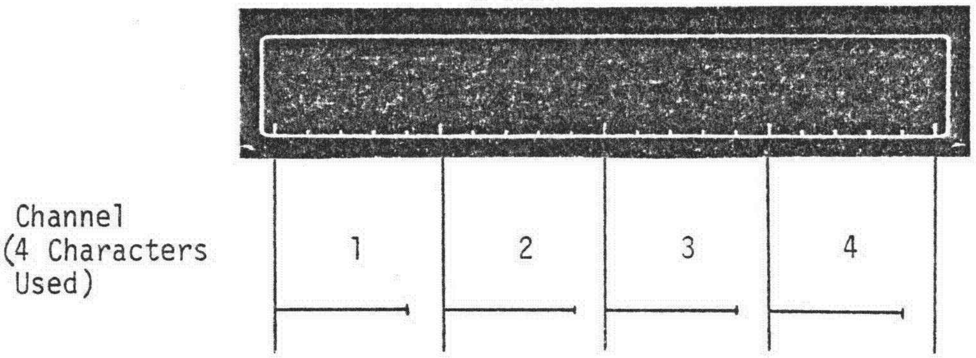


FIGURE 26. Display positions of the simulation program four channels.

The program displays the element description for all four channels during the exact real time indicated by the time stored in the data file. The program continues displaying, one at a time, the sequence of element descriptions stored in the data file for each channel. When there are no further elements to be displayed, the program clears the display and ends the simulation period. If continuous simulation is being used the program automatically starts the simulation period once again, whereas, if the single shot simulation is being used it waits for a keyboard entry before beginning the simulation again.

The second section of the program algorithm is the interrupt subroutine. This interrupt subroutine is executed every MU time unit. When it is executed,all the clocks being used are increased by one time unit. Once the employed clocks are updated, the program compares the clock time against the time defined for each element description being displayed. If any of the clock time equals the time that the element description is supposed to be displayed, a "time completed" flag is set for that particular channel. When all the channels are checked, the interrupt flag is cleared so that the interrupt subroutine can be executed the next time it is called. Figure 27 shows a flowchart of this program.

Memory Map (Addresses)

Figure 28 shows the RAM and ROM memory locations assigned to the variables, programs, and data file. As is seen in this figure, the user has available fourteen pages of RAM. This means that 3584 bytes are free to store data. The stopwatch and simulator programs written in assembler are stored in ROM along with the analysis program written in BASIC. Table 6 illustrates the first and last bytes for each program.

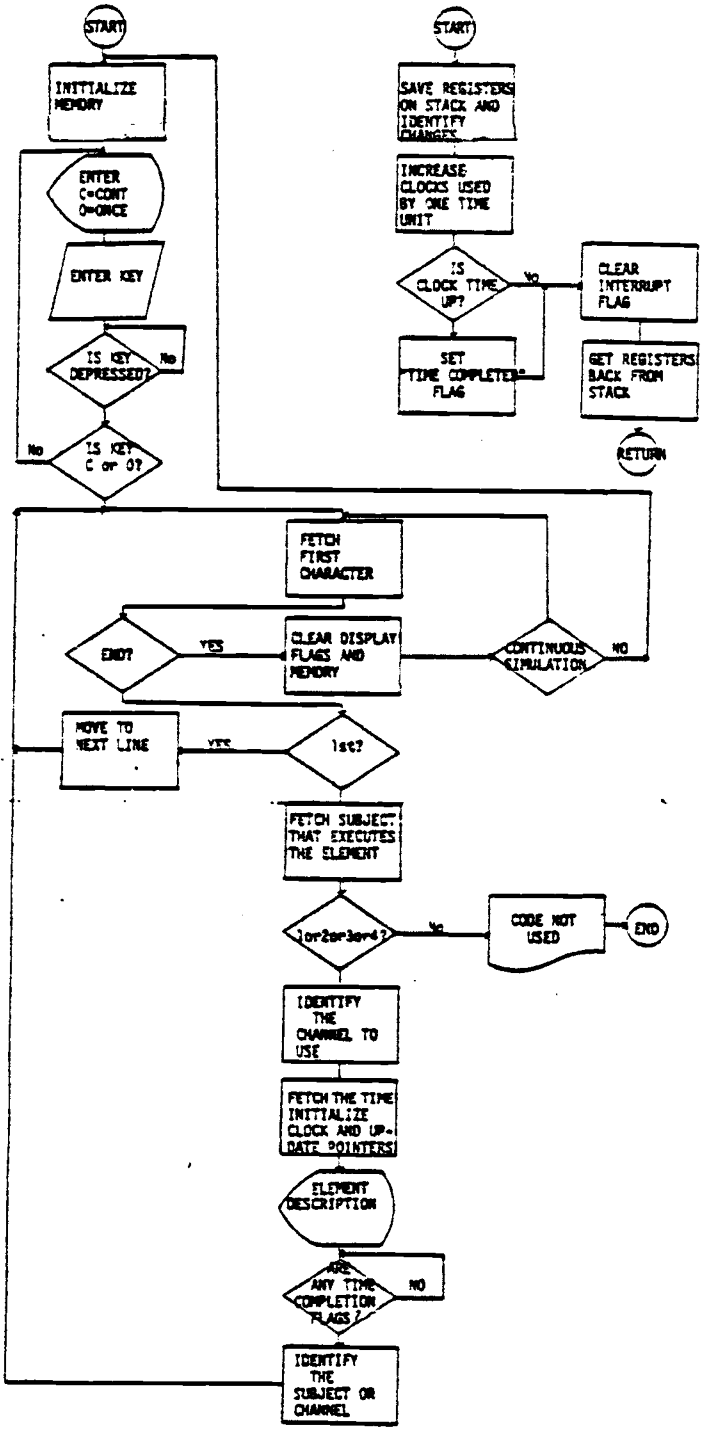

START
INITIALIZE MEMORY
ENTER C=CONT O=ONCE
ENTER KEY
IS KEY DEPRESSED?
No
IS KEY C or O?
No
START
SAVE REGISTERS ON STACK AND IDENTIFY CHANGES
INCREASE CLOCKS USED BY ONE TIME UNIT
IS CLOCK TIME UP?
No
CLEAR INTERRUPT FLAG
SET "TIME COMPLETED" FLAG
GET REGISTERS BACK FROM STACK
RETURN
FETCH FIRST CHARACTER
END?
YES
CLEAR DISPLAY FLAGS AND MEMORY
CONTINUOUS SIMULATION
NO
MOVE TO NEXT LINE
YES
1st?
FETCH SUBJECT THAT EXECUTES THE ELEMENT
1or2or3or4?
No
CODE NOT USED
END
IDENTIFY THE CHANNEL TO USE
FETCH THE TIME INITIALIZE CLOCK AND UP-DATE POINTERS
ELEMENT DESCRIPTION
ARE ANY TIME COMPLETION FLAGS?
NO
IDENTIFY THE SUBJECT OR CHANNEL

FIGURE 27. Simulation program flowchart.

Table 6. 4K ROM Memory Locations of the Stopwatch, Analysis, and Simulation Programs

	Analysis	Stopwatch	Simulator
First Byte Address	D000	DA41	DC46
Last Byte Address	DA3A	DC44	DFF9
Total Number of Bytes	2612	516	954

Storing these three programs in ROM is advantageous as a result of more space being created in the RAM memory for data storage. It is also advantageous in that the microprocessor performs similar to a calculator in which all the functions are permanently programmed. This signifies that the user does not have to reload any program from a cassette tape or by any other means. The user simply turns on the microprocessor and runs the different programs by depressing the corresponding keys on the keyboard. The programs are not lost when the microprocessor's power is turned off.

One RAM page (256 bytes) is used for the program's flags, counters, pointers, and clocks, four RAM pages (1024 bytes) are used for the program variables, and nine RAM pages (2304 bytes) are used for the data files.

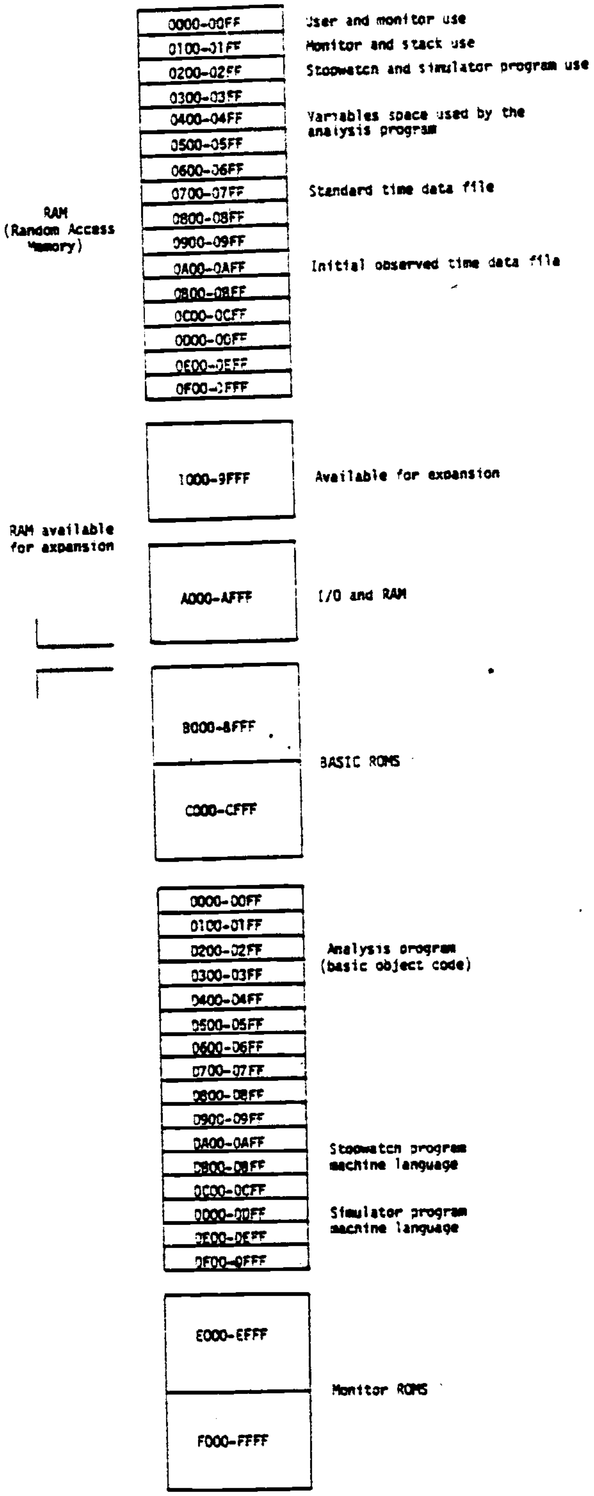


FIGURE 28. RAM and ROM memory map used by the system described in this paper.

Table 7. RAM memory assigned for the program variables and for data files.

	RAM Memory				
	Programs Use	Analysis Prog. Var.	Std. Time Data File	Initial Data File	Total For Data Files
First Byte	0000	0300	0700	0A00	0700
Last Byte	02FF	06FF	09FF	0FFF	0FFF
Total # of Bytes	768	1024	768	1536	2304
Maximum # of Elements	--	39	45	--	--
Maximum # Observations	--	--	45	90	135

This structure designed for the RAM and ROM proves the third hypothesis in which it is stated that the programs created could be written and compiled into 4K bytes ROM, so that the computer is ready whenever it is turned on.

Time, Timers, and Input/Output Ports

The following segment of this paper attempts to prove the fourth hypothesis. It states that the microcomputer's inherent timer and input/output port functions can be used, not only to measure time, but to simultaneously simulate real time operations, create musical tones, and output signals to a light panel. The discussion of the fourth hypothesis will begin with a description of the Versatile Interface Adapter of the microcomputer.

A. Versatile Interface Adapter (VIA) (9)

The Versatile Interface Adapter contains two powerful, flexible interval timers, a serial-to-parallel, parallel-to-serial shift register, and input latching on the peripheral ports. Handshaking capability is expanded to allow control of bi-directional data transfers between VIA's in multiple processor systems and peripherals.

Control of peripherals is primarily through two eight-bit bi-directional ports. Each of these ports can be programmed to act as an input or an output. Peripheral I/O lines can be selectively controlled by the interval timers to generate programmable-frequency square waves and/or to count externally generated pulses. Positive control of VIA functions is gained through its internal register organization: Interrupt Flag Register, Interrupt Enable Register, and two Function Control Registers.

The internal elements of the VIA include (40):

- Data Bus Buffers
- Interrupt Control Logic
- Function Control Logic
- Two Interval Timers
- Two eight-bit Bi-directional ports
- One eight-bit Shift Register

Figure 29 illustrates a block diagram of the VIA.

Some of the VIA's most important uses are to (13):

- generate output signals
- produce interrupts to the microprocessor
- transfer parallel data
- design precision timing programs
- implement frequency counters
- compose music

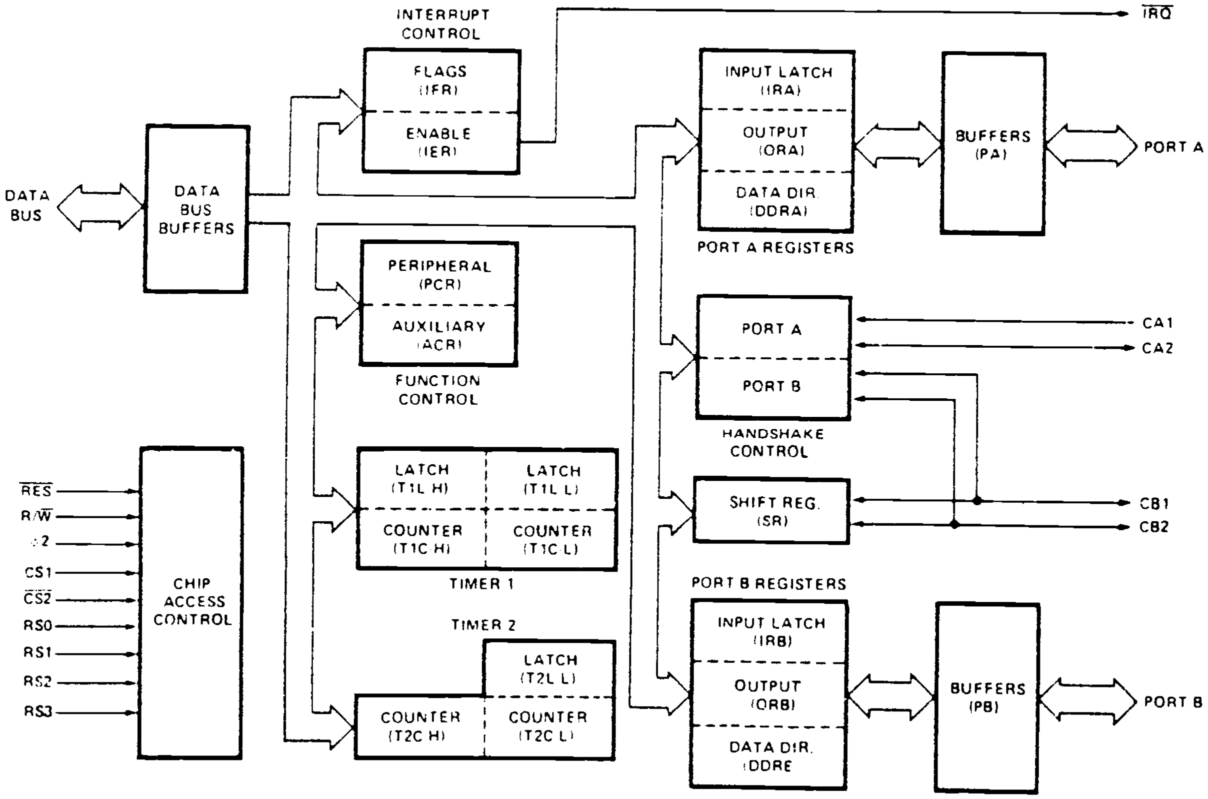


FIGURE 29. VIA Block Diagram

B. Timers

The two timers found on the AIM 65 are positions of memory controlled by the Versatile Interface Adapter (VIA). Each timer consists of a latch and a counter. A latch is a memory position that sustains the number of cycles counted by the timer counter. A counter is a memory position (two bytes) that is loaded along with the number of cycles stored in the latch. The number of cycles stored in the latch is equivalent to the time unit used. When the counter is loaded with the number of cycles desired, the VIA begins decreasing the number of cycles at the system clock rate until the counter reaches zero. When this happens a flag is set in the Interrupt Flag Register to inform the microprocessor that the time is completed. If the corresponding bit in the Interrupt Enable Register is set at logic 1, the microprocessor serves the interrupt subroutine as defined by the user.

The various features of both timers are controlled by the eight-bit status of the Interrupt Enable Register (IER), Figure 30, the Interrupt Flag Register (IFR), Figure 31, and the Auxiliar Control Register (ACR), Figure 32.

The system clock frequency equals one megaherz. This indicates that a million cycles are executed every second. In order to use MU time units the number of cycles to be counted by the timer are:

$$\frac{1000000 \text{ cycles}}{\text{seconds}} * \frac{3600 \text{ sec.}}{1 \text{ hr.}} * \frac{0.00001 \text{ hrs.}}{1\text{TMU}} * \frac{1\text{TMU}}{1 \text{ MU}} = \frac{3600 \text{ cycles}}{\text{MU}}$$

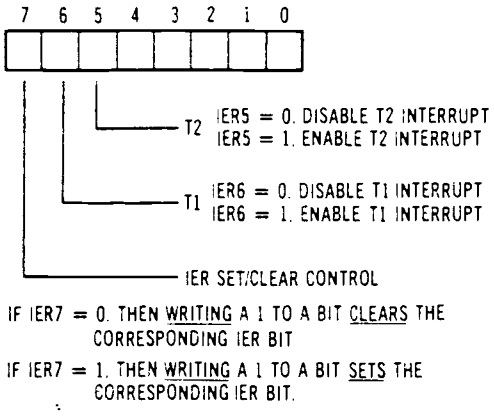

7 6 5 4 3 2 1 0
T2
IER5 = 0. DISABLE T2 INTERRUPT
IER5 = 1. ENABLE T2 INTERRUPT
T1
IER6 = 0. DISABLE T1 INTERRUPT
IER6 = 1. ENABLE T1 INTERRUPT
IER SET/CLEAR CONTROL
IF IER7 = 0. THEN WRITING A 1 TO A BIT CLEARS THE CORRESPONDING IER BIT
IF IER7 = 1. THEN WRITING A 1 TO A BIT SETS THE CORRESPONDING IER BIT.

FIGURE 30. Operation of the Interrupt Enable Register (IER).

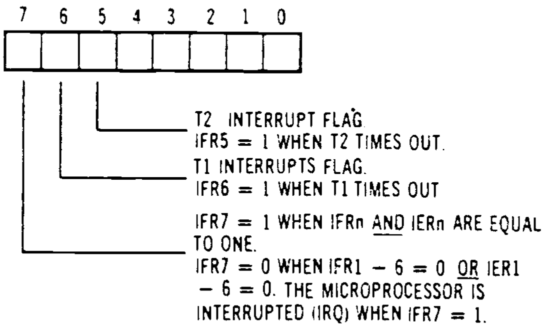

7 6 5 4 3 2 1 0
T2 INTERRUPT FLAG
IFR5 = 1 WHEN T2 TIMES OUT.
T1 INTERRUPTS FLAG.
IFR6 = 1 WHEN T1 TIMES OUT
IFR7 = 1 WHEN IFRn AND IERn ARE EQUAL TO ONE.
IFR7 = 0 WHEN IFR1 – 6 = 0 OR IER1 – 6 = 0. THE MICROPROCESSOR IS INTERRUPTED (IRQ) WHEN IFR7 = 1.

FIGURE 31. Operation of the Interrupt Flag Register (IFR).

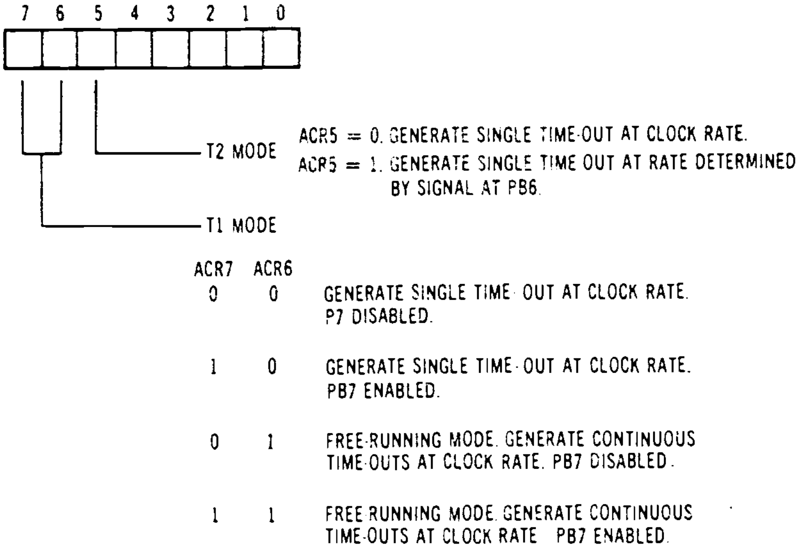
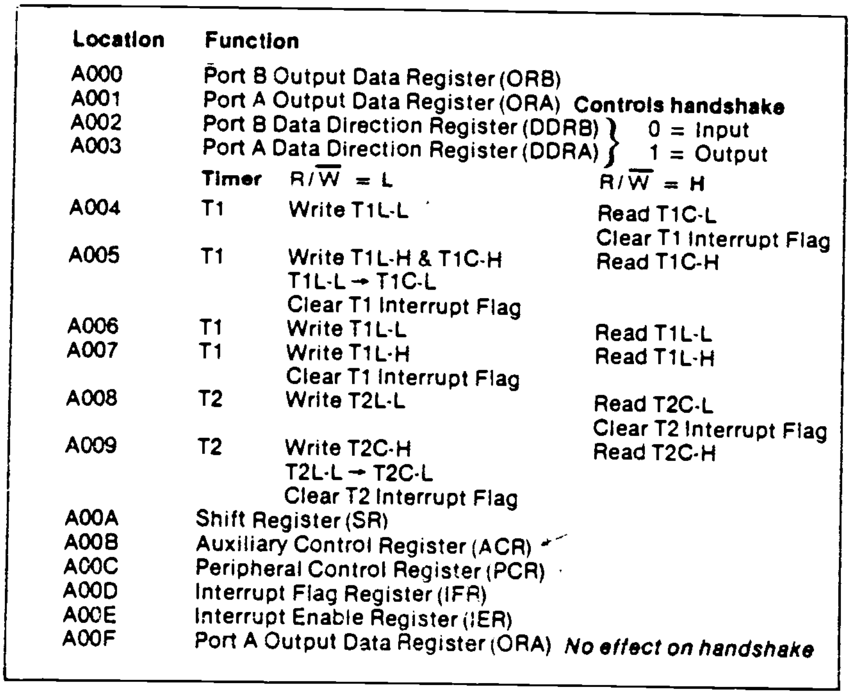

7 6 5 4 3 2 1 0
T2 MODE
ACR5 = 0. GENERATE SINGLE TIME-OUT AT CLOCK RATE.
ACR5 = 1. GENERATE SINGLE TIME OUT AT RATE DETERMINED BY SIGNAL AT PB6.
T1 MODE
ACR7 ACR6
0 0 GENERATE SINGLE TIME OUT AT CLOCK RATE. P7 DISABLED.
1 0 GENERATE SINGLE TIME-OUT AT CLOCK RATE. PB7 ENABLED.
0 1 FREE-RUNNING MODE. GENERATE CONTINUOUS TIME-OUTS AT CLOCK RATE. PB7 DISABLED.
1 1 FREE RUNNING MODE. GENERATE CONTINUOUS TIME-OUTS AT CLOCK RATE PB7 ENABLED.

FIGURE 32. Operation of the Auxiliary Control Register (ACR).

Table 8. Memory locations used by the VIA.



Location	**Function**		
A000	Port B Output Data Register (ORB)		
A001	Port A Output Data Register (ORA)	**Controls handshake**	
A002	Port B Data Direction Register (DDRB)	0 = Input	
A003	Port A Data Direction Register (DDRA)	1 = Output	
	Timer	$R/\overline{W}$ = L	$R/\overline{W}$ = H
A004	T1	Write T1L-L	Read T1C-L Clear T1 Interrupt Flag
A005	T1	Write T1L-H & T1C-H T1L-L → T1C-L Clear T1 Interrupt Flag	Read T1C-H
A006	T1	Write T1L-L	Read T1L-L
A007	T1	Write T1L-H Clear T1 Interrupt Flag	Read T1L-H
A008	T2	Write T2L-L	Read T2C-L Clear T2 Interrupt Flag
A009	T2	Write T2C-H T2L-L → T2C-L Clear T2 Interrupt Flag	Read T2C-H
A00A	Shift Register (SR)		
A00B	Auxiliary Control Register (ACR)		
A00C	Peripheral Control Register (PCR)		
A00D	Interrupt Flag Register (IFR)		
A00E	Interrupt Enable Register (IER)		
A00F	Port A Output Data Register (ORA)	*No effect on handshake*	

The number to be loaded into the timer latch is 3600-2 as a result of two additional cycles being used to reload the counter after each interrupt (13). Table 8 shows the memory location used by the timers in the AIM 65 microcomputer (37).

The number of cycles loaded into the timer latch (a temporary data register) should be less than the number of cycles used in the interrupt subroutine, otherwise errors will occur in the user interrupt subroutine. An interrupt subroutine taking 74 cycles to be executed is used in the stopwatch program. Another interrupt subroutine taking 280 cycles to be executed is used in the simulation program. This number of cycles is obtained by totaling the cycles required for each machine instruction in the interrupt subroutine.

C. Input/Output Ports

The VIA contains two sixteen-bit bi-directional ports (A and B) for input and output interfacing. A port consists of a one byte (eight bits) memory location that is used to input or output signals depending on the status of the Port Data Direction Register (see Table 8). When the status of a bit is at logic 0, the bit is used as an input port and when the status of a bit is at logic 1, the bit is used as an output port. For example, when bit 5 of port B Data Direction Register Byte is at logic 1 and bit 5 of port B is at logic 1, a signal of 5 volts is sent out through port B number 5.

In the simulation program described above, the two timers and four bits of port B are put to work together with the microprocessor. Timer number one is used to run four clocks in ASCII code independently of each other. The timer is loaded with OEOE in hexadecimal (3598 in decimal) in order to produce an interrupt to the microprocessor every MU time unit. When an interrupt occurs, the interrupt subroutine is executed and the clocks that are being used by the program are incremented by one unit using decimal addition.

Timer number two is used to generate an output signal to the speaker through the shift register. The frequency of the output signal is 0.064252 seconds (250 decimal loaded on the timer two latch). The duration of the output signal is determined by a loop of 42 cycle instructions. This means that when the program is checking the "time completed" flags for the four channels, it simultaneously increases a two byte counter stored in the memory. When the high order byte of this counter reaches 256 in decimal, the timer is loaded with zero in order to stop the output signal to the speaker. This loop takes approximately 21500 cycles (42 cycles * 256*2) which is equivalent to 0.0215 seconds (21500 cycles/1000000 cycles-second).

Bits zero through three of port B are used for the lights corresponding to each channel. All the bits of port B are set as output ports by loading 255 decimal into the Data Direction Register of port B. When the element description is displayed, the corresponding bit of port B is set to high (light on) if it was previously low, or to low (light off) if it was previously high. This signifies that there is one light for each channel that shifts from on to off and vice versa each time a new element is displayed. The speaker beeps each time a change occurs in the display. This feature assists the analyst because it does not require him/her to continuously look at the display.

This demonstrates that the two timers and four bits of port B can be put to work together in order to simultaneously measure time, to output signals to the speaker, and to control a light panel of four lights while running the simulation of the data files.

An Illustrated Use of the Stopwatch, Analysis and Simulation Programs

The following is the number of elements in which the operation being measured is divided.

Operation: Sacking Candy

Elements	4 Character Symbols
1. Get and move bag to hopper mouth	GEMO
2. Fill bag with candy	FILL
3. Weigh and fold top of bag	WG&F
4. Staple bag shut	STPL
5. Place bag in finished product box	PLCE

A - Procedure to Initialize the F1 and F2 Keys (The user entries and responses are underlined).

1. Turn the microcomputer on. (Note: The EPROM chip must be inserted in the Z24 position of the microcomputer before executing this step.)

ROCKWELL AIM 65

2. Depress the N key followed by the ESCAPE key in order to initialize the F1 and F2 jump instructions.

```
(N)
# ELEMENTS?  ESCAPE
```

NOTE: CR = RETURN key (carriage return)
This example follows the data flow logic shown in Figure 5.

B - Procedure to Change the Gap Size (the space between blocks of data) Used.

3. If the gap size is to be greater than the default size (08 hexadecimal), change its corresponding value in memory position A409 by using the memory insertion command. In the following example a gap size of 20 is used.

```
(M)=A409 08 02 CA 02
(/) A409 20
```

C - Procedure to Use the Stopwatch Program

4. Depress the F1 key to start running the stopwatch program. The program will display "ENTER MAX 3 CHR #".

5. Enter the observations as they occur in the field. Begin by entering the element number followed by depressing; a) the S key when the element starts, b) the E key when the element ends, c) the C key when the element ends and the next element starts,

d) the F key when a foreign element starts or ends or 3) the DEL key when the element number entered in the display is to be erased. Depress the K key to erase the last observation taken.

6. Depress the RETURN key when no further observations are desired. The stopwatch program will print:

```
EDIT OR SAVE FILE
```

D - A Procedure to List and Save the Initial Data File

7. Depress the ESCAPE key.

8. Reenter the Text Editor by depressing the T key. The top line of the data file will be printed.

```
(T)
S1 1   00004211
```

9. Depress the L key followed by the period key (.) to indicate that the entire program is to be listed.

```
=(L)
/.
OUT= CR
S1 1   ØØØØ4211
E1 1   ØØØØ69Ø9
C1 2   ØØØ11424
C1 3   ØØØ175Ø8
C1 4   ØØØ19693
C1 5   ØØØ2Ø859
C1 1   ØØØ23737
C1 2   ØØØ2775Ø
C1 3   ØØØ34133
C1 4   ØØØ36221
C1 5   ØØØ3733Ø
C1 1   ØØØ4Ø395
C1 2   ØØØ44626
C1 3   ØØØ5Ø68Ø
C1 4   ØØØ52869
C1 5   ØØØ54181
```

```
S1 2   ØØØ5686Ø
E1 2   ØØØ61386
C1 3   ØØØ67614
S1 5   ØØØ69643
E1 5   ØØØ71ØØ6
C1 1   ØØØ7367Ø
C1 2   ØØØ78118
C1 3   ØØØ84129
C1 4   ØØØ86327
C1 5   ØØØ87775
```

10. Depress the T key in order to reset the Text Editor pointer to the top of the data file.

```
=(T)
S1 1   ØØØØ4211
```

11. Once again depress the L key followed by the period key. Now, depress the T key. Enter the five character name for the data file along with the tape recorder number (default is 1). Once this is done place a cassette tape in the tape recorder in the position desired to save the data file. Take note of the tape recorder counter. Turn the volume to maximum and the tone to high.
12. Depress the RECORD and PLAY keys in order to start the tape recorder recording. Depress the RETURN key of the microcomputer to begin saving the data file. The data block number being saved is displayed in the last two positions of the display.
13. Once the data file is saved the display will show "END". Stop the tape recorder. Take note of the tape recorder counter for future reference.

14. Depress the ESCAPE key to send control to the monitor.

```
=(L)
/.
OUT=T F=DAFI1 T=1 CR

END ESCAPE
```

E - Procedure to Use the Analysis Program

15. Depress the N key to start running the program.

16. Enter the number of elements and the precision desired.

```
(N)
# ELEMENTS? 5
PRECISION(%)? 5

TIME UNIT=1/1ØØ MIN.
```

17. The program will print the discrete times, in 1/100 of a minute, calculated for each observation recorded in the data file. The number for each observation of each element is also displayed.

```
    *DATA FILE*
ELE#  TIME    OB.#
1   16.188   ( 1 )
2   27.09    ( 1 )
3   36.504   ( 1 )
4   13.11    ( 1 )
5   6.996    ( 1 )
1   17.268   ( 2 )
2   24.078   ( 2 )
3   38.298   ( 2 )
4   12.528   ( 2 )
5   6.654    ( 2 )
1   18.39    ( 3 )
2   25.386   ( 3 )
3   36.324   ( 3 )
4   13.124   ( 3 )
5   7.872    ( 3 )
1   27.156   ( 4 )
2   37.368   ( 4 )
3   8.175    ( 4 )
5   15.984   ( 4 )
```

```
2  26.688   ( 5 )
3  36.066   ( 5 )
4  13.188   ( 4 )
5  8.688    ( 5 )
```

18. A summary will also be obtained for all the elements. The summary begins by displaying the total time spent in the observation field. It then displays the average time (AVG. T), the standard deviation (STD), the number of observations taken (OB. NOW), and the number of observations required for all the elements including the foreign elements (OB. REQ).

```
      SUMMARY
OBS. TIME= 526.65

ELE# AVG. T & STD
 1  16.9575  1. 1
 2  26.0976  1. 32
 3  36.912  .91
 4  12.99  .3
 5  7.6776  .83
F.  0

ELE# OB. NOW OB.REQ
 1      4      7
 2      5      5
 3      5      1
 4      4      1
 5      5      20
F.      0
```

As the preceding example demonstrates, more observations are needed for elements one and five. In this situation it is not desirable to create a data file of standard times. This means that N should be entered when "DATA FILE (Y-N?)" is displayed.

```
DATA FILE (Y-N)? N CR
```

19. Depress the ESCAPE key to send control to the monitor. Upon completing this analysis the initial data file is modified. The modifications consist basically in the transformation of continuous time to discrete time. When the program calculates the discrete time for each observation it also stores it in the data file. (See step 17). This modification is done only in the lines of the data file beginning with the letter E and C. This is done because these two letters indicate when an element ends. The lines of the data file beginning with the letter S are not modified because the time stored in these lines only indicates when the elements start.

In the occurrence of foreign elements, the time in the second line is modified. The second line is the line indicating the end of the foreign element. Also, the F of the first line, indicating the starting time of the foreign element, is changed to S.

Comparison of both data files before and after the analysis.

```
Initial Data File              Modified Data File
(Continuous Time)               (Discrete Time)

=(T)                           (T)
S1 1     00004211              ST 1     00004211
=(L)                           =(L)
/.                             /.
OUT=CR                         OUT=CR
S1 1     00003111              S1 1     00004211
E1 1     00006909              E1 1     00002698
C1 2     00011424              C1 2     00004515
C1 3     00017508              C1 3     00006804
C1 4     00019693              C1 4     00002185
C1 5     00020859              C1 5     00001166
C1 1     00023737              C1 1     00002878
C1 2     00027750              C1 2     00004013
C1 3     00034133              C1 3     00006383
C1 4     00036221              C1 4     00002088
C1 5     00037330              C1 5     00001109
C1 1     00040395              C1 1     00003065
C1 2     00044626              C1 2     00004231
C1 3     00050780              C1 3     00006054
C1 4     00052869              C1 4     00002189
C1 5     00054181              C1 5     00001312
S1 2     00056860              S1 2     00056860
E1 2     00061386              E1 2     00004526
C1 3     00067614              C1 3     00006228
S1 5     00069643              S1 5     00069643
E1 5     00071006              E1 5     00001363
C1 1     00073670              C1 1     00002664
C1 2     00078118              C1 2     00004448
C1 3     00084129              C1 3     00006011
C1 4     00086327              C1 4     00002198
C1 5     00087775              C1 5     00001448
```

20. In order to save the new discrete time data file on cassette tape, the same procedure used to save the initial data file is followed. See steps 7 through 14.

```
=(T)
S1 1     00004211
=(L)
/.
OUT=T F=DAF12 T=1 CR
```

21. Continue running the stopwatch program in order to take more observations of those elements requiring that more observations are taken. Follow the same procedure used to run the stopwatch program as described above. The listing of the new data file created is:

```
(T)
S1 1     00005476
=(L)
/.
OUT=CR
S1 1     00005476
E1 1     00008154
S1 5     00020543
E1 5     00021804
C1 1     00024562
S1 5     00037289
E1 5     00038458
C1 1     00041242
S1 5     00054087
E1 5     00055333
S1 1     00071783
E1 1     00074462
S1 5     00087222
E1 5     00088434
C1 1     00091156
S1 5     00103939
E1 5     00105152
C1 1     00107812
S1 5     00120610
E1 5     00121731
C1 1     00124516
S1 5     00137345
E1 5     00138553
C1 1     00141197
S1 5     00154029
E1 5     00155224
C1 1     00157869
S1 5     00170666
E1 5     00171847
S1 5     00187249
E1 5     00188498
```

22. The second data file just created is analyzed by using the analysis program. In order to save the results of the previous data file analysis, reenter the analysis program using the 6 key. Type the word "CONT" and depress the RETURN key to start the analysis program running. The program will output the discrete times of the elements contained in the new data file. A continuation of the observation number for each element is also printed.

```
(6)
CONT CR

    *DATA FILE*
ELE# TIME      OB.#
 1  16.068    ( 5 )
 5  7.566     ( 6 )
 1  16.548    ( 6 )
 5  7.014     ( 7 )
 1  16.704    ( 7 )
 5  7.476     ( 8 )
 1  16.074    ( 8 )
 5  7.272     ( 9 )
 1  16.332    ( 9 )
 5  7.278     ( 10 )
 1  15.96     ( 10 )
 5  6.726     ( 11 )
 1  16.71     ( 11 )
 5  7.248     ( 12 )
 1  15.864    ( 12 )
 5  7.17      ( 13 )
 1  15.87     ( 13 )
 5  7.086     ( 14 )
 5  7.494     ( 15 )
```

23. A new summary is obtained in which the average time, standard deviation, number of observations taken, and number of observations required are calculated using the results of the current and previous data files.

```
      SUMMARY
TOT. TIME= 1657.638

ELE# AVG. T & STD
 1  16.4584615
 .71
 2  26.6796  1.32
 3  36.912   .91
 4  11.99  .3
 5  7.3812  .53
F.  0

ELE# OB.NOW OB.REQ
 1     13     3
 2     5     5
 3     5     1
 4     4     1
 5     15     9
F.     0
```

24. As demonstrated in the above example, no further observations of any elements are required. If this second discrete time data file, obtained by running the analysis programs, is to be listed, edited or saved, enter N when "DATA FILE (Y-N)" is displayed. Depress the ESCAPE key and then save or list the file using the procedure described above to save a data file. To edit it use the Text Editor commands. If this second file is not to be listed, edited, or saved, continue on to step 25.

```
                DATA FILE(Y-N)? N
                DATA FILE(Y-N)? N ESCAPE

Second Data File              Modified Second Data File
(Continuous Time)                  (Discrete Times)

(T)                           (T)
ST 1      00005476            ST 1      00005476
=(L)                          =(L)
/.                            /.
OUT=CR                        OUT=CR
S1 1      00005476            S1 1      00005476
E1 1      00008154            E1 1      00002678
S1 5      00020543            S1 5      00020543
E1 5      00021804            E1 5      00001261
C1 1      00024562            C1 1      00002758
S1 5      00037289            S1 5      00037289
E1 5      00038458            E1 5      00001169
C1 1      00041242            C1 1      00002784
S1 5      00054087            S1 5      00054087
E1 5      00055333            E1 5      00001246
S1 1      00071783            S1 1      00071783
E1 1      00074462            E1 1      00002679
S1 5      00087222            S1 5      00087222
E1 5      00088434            E1 5      00001212
E1 1      00091156            C1 1      00002722
S1 5      00103939            S1 5      00103939
E1 5      00105152            E1 5      00001213
C1 1      00107812            C1 1      00002660
S1 5      00120610            S1 5      00120610
E1 5      00121731            E1 5      00001121
C1 1      00124516            C1 1      00002785
S1 5      00137345            S1 5      00137345
E1 5      00138553            E1 5      00001208
C1 1      00141197            C1 1      00002644
S1 5      00154029            S1 5      00154029
E1 5      00155224            E1 5      00001195
C1 1      00157869            C1 1      00002645
S1 5      00170666            S1 5      00170666
E1 5      00171847            E1 5      00001181
S1 5      00187249            S1 5      00187249
E1 5      00188498            E1 5      00001249
```

25. In order to create a new data file containing only the standard times for the five elements, the analysis program can once again be utilized. In order to do this

reenter the analysis program by depressing the 6 key and typing the word "CONT". Depress the RETURN key and the program will start running.

26. Enter the rating and percentage of allowances for each element as requested by the program. A hard copy of these variables are printed.

```
EL. RTG ALW STD. TIME
 1  105  10
 19.2015285
 2  100  10
 28.9773334
 3  95   10
 38.9626667
 4  115  15
 17.5747059
 5  100  5
 7.76968421
TOT TIME=
 112.485929
```

27. When "FILE CREATED" is displayed, a new data file containing the standard time for all the elements is stored in the memory. The Text Editor memory pointers are also set in order to list, edit, or save this new data file.

```
FILE CREATED
```

28. Depress the ESCAPE key to send control to the monitor.

29. To list or save this new standard time data file,use the same procedure described earlier under how to list or save a data file. This new data file can also be edited using the Text Editor commands.

```
(T)
E1  1    00003200
=(L)
/.
OUT=CR
E1  1    00003200        Listing the Standard
E1  2    00004829        Time Data File
E1  3    00006493
E1  4    00002929
E1  5    00001294


=(T)
E1  1    00003200        Saving The Standard
=(L)                     Time Data File
/.
OUT=T F=STDTM T=1

END ESCAPE
```

30. To add a title line to the new data file begin by depressing T to reenter the Text Editor. Next, insert the title line making sure that the first character is a T and that it is not greater than sixteen characters. Use T to set the Text Editor's pointers back to the top of the data file and L to list it.

```
(T)
E1  1    00003200

=(I)
TST SACKING 8581 CR
E1  1    00003200

=(T)
TST SACKING 8581
=(L)
/.
OUT=CR
TST SACKING 8581
E1  1    00003200
E1  2    00004829
E1  3    00006493
E1  4    00002929
E1  5    00001294
```

F - Procedure to Simulate the Standard Time Data File

31. In order to simulate the new standard time data file, an element description is more appropriate than an element number. The Text Editor commands are used to edit and change the respective element number for each element to a four character description as shown at the beginning of the following example.

```
(T)
TST SACKING 8581
=(D)
El  1     00003200
=(C)
 1
El  1     00003200
TO=GEMO
El GEMO  00003200
=(D)
El  2     00004829
=(C)
 2
El  2     00004829
TO=FILL
El FILL  00004829
=(D)
El  3     00006493
=(C)
 3
El  3     00006493
TO=WG&F
El WG&F  00006493
=(D)
El  4     00002929
=(C)
 4
El  4     00002929
TO=STPL
El STPL  00002929
=(D)
El  5     00001294
=(C)
 5
El  5     00001294
TO=PLCE
El PLCE  00001294
```

```
=(T)
TST SACKING 8581
=(L)
/.
OUT=CR
TST SACKING 8581
E1 GEMO  00003200
E1 FILL  00004829
E1 WG&F  00006493
E1 STPL  00002929
E1 PLCE  00001294
END ESCAPE
```

32. Turn off the printer. Depress the F2 key to start the simulation program running. Depress the C or O key depending on if the simulation is to be done continuously or only once.

33. Depress the RESET key or the ESCAPE key to end the simulation.

Applications

All the programs that have been described in this paper are stored in ROM in a single chip. This chip is located in the Z24 position found on the AIM 65 microcomputer board. The Industrial Engineer does not have to load any of the programs into the microcomputer because all of them are programmed permanently in the chip. Using this single chip the Engineer does not have to spend time manually processing the initial data obtained in the field. The microcomputer may also be turned off without losing the programs. If desired, a tape recorder may be attached to the microcomputer in order to save the data files created from the field.

Yet, if a printout of the data files is all that is needed, it may be obtained from the onboard printer of the microcomputer.

The AIM 65 microcomputer has the advantage of being small (only 14.5' x 12' including the keyboard) and light weight making it easy to carry around. If desired a portable case may be built in order to protect it and facilitate its transportation to the field. Power for the printer unit (24 volts) and for the microcomputer (5 volts) may be obtained by attaching a power supply unit.

The major application of the system presented in this paper is the single and multiple Machine and Operator Motion and Time Study. Time study is conducted primarily by the stopwatch and the analysis program while motion study is performed by the real time simulator program. Time studies can be done on single elements (even the MTM elements), single operations, single subject, multiple operations, and multiple subjects. Due to the structure of the stopwatch program, different elements of multiple operators can be measured simultaneously.

These various programs can measure, analyze, and simulate short-cycled repetitive operations as well as single long-cycled machine and operator operations. The real time data files obtained from the operations can be used for learning curves analysis, production control, incentive plans, and many other Industrial Engineering functions.

The simulation program can also be used for employee training and for implementation of improved methods. The microcomputer is helpful in this area for various reasons. It can be placed in front of an operator and display to the operator exactly what the operation is, the speed at which the operation should be executed, and a description of what the operator is supposed to do at all times. Likewise, the program can simulate a fictitious machine operation.

A verification process can be done to check if there are any missed or extra elements in the operation that were not described in the final method description.

When using the MTM analysis in the plant, a data file of the micromovements can be created using the Text Editor. This data file can be simulated in the display to show the sequence and effectiveness of the method described. The four channels of the simulator program also allows the simulation of two different MTM methods to occur, simultaneously. This helps the Industrial Engineer do a more objective and scientific evaluation of different methods proposed for any operation. All of these applications can be done faster, at a lower cost, and with improved accuracy using the four channel real time simulator program.

This feature is important, not only to justify the purchase of new NC machines or robots, but also to plan their integration into existing production systems. Standard time data can be memorized by the computer and recalled automatically when the activity is identified. The authors have already developed a variation of the ROM described in this paper that contains MTM-3 codes. The analyst need only type in the element description, say M32B, and have the AIM 65 automatically fill in the values.

The simulation program is very useful when using work sampling. It informs the analyst of when a new observation must be taken. The Universal Time Data structure shown in Figure 3 may also be used to define the operator or machine from which the observation was taken. For example, the element description is used to display the observation number while the time is used to store the time of day at which the observation must be taken. Once the simulation program is started at the beginning of the observation period, it displays the observation number for each operator and/or machine. The program also beeps when the observation is to be taken. This feature will keep the analyst from having to be dependent on his/her watch.

The stopwatch program can also be used to record the observations from work sampling. For example, the different keys used to denote the start or end of the elements can also be used to indicate the variables being measured by work sampling. (For example, E = productive, S = non-productive, C = delays, etc.) The element number can also be used to store remarks on the data file corresponding to each observation.

The data base structure and the programming capabilities of the microprocessor allow the Industrial Engineer to develop other analysis programs that will meet specific needs in the entire Work Measurement field.

Conclusions

Several hypotheses were set forth before the project began. Some, rather obvious, others quite ambitious. The overall conclusion is that we were able to meet the primary objective of the project: to design a system for Work Measurement based upon a Universal Time Data structure using a low-cost microcomputer system.

Hypothesis #1: A unified data base structure can be used to accommodate a variety of industrial needs. Surveys showed the stopwatch method and the MTM-based predetermined time method to be the more popular time study tools. Those, as well as work sampling are predominantly carried out by manual methods in the large segment of industry. The UTD format appears to fit these tools effectively.

Hypothesis #2: Data files created according to the Universal Time Data structure are accessed and edited by the analyst using a conventional Text Editor, similar to the editor used in work processing. The UTD format consists of seventeen bytes of ASCII characters. The format combines the stopwatch and MTM methods into a single data record consisting of an ID, the subject, the element, and time data. The Text Editor also facilitated the creation and retrieval of tape files.

Hypothesis #3: The program was written, compiled, and stored onto a 4K-Byte ROM, so that the system is ready as soon as power is applied. User function key 1 is used to jump to the stopwatch program, user function key 2 to the simulation program and N key to the analysis program. This freed the 4-K Bytes of RAM to be used for the program clocks, pointers, and counters (256 bytes), the program variables (1024 Bytes) and the data file (2304 Bytes).

Hypothesis #4: The timer functions of microprocessors were used, not only to measure and record time data, but also to real-time simulate activities recorded in RAMs as well as to create audible tones to prompt the user via a speaker. Four bits of port B of the VIA are used to control a panel of four lights corresponding to each of the first four channels (subjects) during the simulation. The lights alternate on and off as one element is terminated and the next element is started by the same subject. The timer number one was used for the program clocks and the second timer was used to generate the beeps on the speaker.

The AIM 65 microcomputer used in this project turned out to be an excellent machine. It had adequate input/output ports and flexibility, and the user dedicated VIA was just the needed microprocessor to program timing tasks. Together with its on board printer and display, this microcomputer convinced us that help is on its way to alleviate the Industrial Engineers from the drugeries associated with manual time studies, and to help us practice what we preach -- increase our own productivity!

IV. CONCLUSION AND AREAS FOR FUTURE RESEARCH

Much of the conclusions for this project have already been included in Chapter II and Chapter III. All four hypotheses stated in Chapter I were affirmatively supported by the outcomes. Microprocessors do appear to have unsurpassed and unrealized potentials to make the Work Design and Measurement tasks easier for Industrial Engineers.

As to the areas of future research, there are two categories of subjects. First, there are improvements that are possible but have not been implemented because of lack of time and/or resources. Second, there are new combinations of technologies that can make the use of microprocesses even more exciting.

Short-Range Research Areas

The MTM version of ROM still has 870 Bytes of unused space. Several alternative uses may be contemplated:

(1) Expand the MTM data set to include MTM-1, MTM-2, Work Factor, or Robotic Data.

(2) Implement a work sampling program to automatically create a random observation schedule and alert the analyst when time comes for the observation.

(3) Try to compress a shortened version of the stopwatch program, so that one ROM may be used for both stopwatch and MTM applications.

The use of the input and output interface capabilities of the microcomputer has not been fully utilized. A set of flexible electronic sensors may be attached to automatically create time data. Similarly, a set of relays may be used to control machines directly off the simulation program.

There are also new, more powerful microcomputers available. For example, Rockwell International now markets a 40-column version of the AIM 65. This would allow eight channels to be simulated simultaneously. Also the 68000 series of microprocessors provide 16-bit capability of data processing instead of just eight at a time.

Perhaps even more important is the fact that our proposed systems must be tried out by Industrial Engineers in their day-to-day observations and analyses. Though the systems have been demonstrated to a limited number of Industrial Engineers, no extensive application experience is yet available. Many future improvements are likely to come from the feedback obtained after the system has operated in the field for some length of time

Another evaluation is what the microprocessor manufacturer may provide after reviewing the ROM functions. Since both the firmware and the software designs do depend upon available hardware, manufacturer's evaluations could lead to an improved composite system. For example, using a faster (4-Megahertz) clock may improve the precision of the measurement by shortening the time required to process interrupts and monitor VIAs.

Long-Range Research Areas

Microprocessor technology is a recent technological development but by no means the only development available to Industrial Engineers. Interfacing it to other technologies, such as videotape recording, robotics, graphic computer terminals, and optical fiber data transmission, just to name a few is possible.

Computer graphics could provide visual pictures of the operation to be performed, and transducers could be used to sense if the operation has actually taken place or not. Such a system could keep track of the statistical data and suggest updating or modifying the UTD files when necessary.

The types of activities performed by Industrial Engineers will also be changing with the advances in technology. More work measurement activities may be relating to robotics and flexible production methods. The UTD's ability to quickly update itself is already an asset. Perhaps a new predetermined time data system will emerge that enhances the advantages the current MTM system enjoys. Robotic data and ergonomic data may be used interchangeably to design a system that is the most appropriate for a given task and project.

Eventually, the UTD system may feed into an MRP or MIS system, so that the production schedule is produced according not just to demands and inventories, but also according to the production resources and level of expertise among the available work force. Each work team can then decide how it chooses to handle a specific production assignment.

Potential applications and advantages to be gained are literally unlimited. This thesis represents but the first step toward an "intelligent" use of machine intelligence by Industrial Engineers working in Work Design and Measurement.

BIBLIOGRAPHY

1. Adams, S. K., McGrath, J. J., Set 1979, A Procedure for an Economic Comparison of Work Measurement Techniques, Part I: The Model, AIIE Transactions, 11 3, pp. 229-236.

2. ______________, Set 1979, A Procedure for an Economic Comparison of Work Measurement Techniques, Part II: An Application, AIIE Transactions 11 3, pp. 237-244.

3. Antis, W., Honeycutt, J. M. Jr., Koch, E. N., 1969, The Basic Motions of MTM, The Maynard Foundation, Pittsburg, 16-15 p.

4. Barden, W. Jr., 1977. How to Program Microcomputers, Howard and Sams & Co., Inc. 256 p.

5. Barnes, R. M., 1963. Motion and Time Study, design and measurement of work, John Wiley & Sons, Inc. 739 p.

6. Brisley, C. L., Dossett, R. J., 1980. Computer Use and Non-direct Labor Measurement will Transform Profession in the Next Decade, Industrial Engineering, 12 8, pp. 34-43.

7. Buffa, E. S., 1958. The Electronic Time Recorder, a new instrument for work measurement research, Journal of Industrial Engineering, 9 2, pp. 89-92.

8. Butterfield, J., Inside BASIC, Interactive, Issue No. 2, pp. 6-8.

9. Camp, R. C., Smay, T. A., Triska, C. J., 1979. Microprocessor Systems Engineering, Matrix Publishers, Inc. 641 p.

10. Cannon, Don L., Luecke, G., 1979. Understanding Microprocessors, Texas Instruments, Inc., 8-26 p.

11. Clark, D. O., 1972. Meet the MTM Family, The Journal of Methods-Time Measurement, 17 3, pp. 3-13.

12. Coe, D. D., Naliwajek, Rosemary, 1981. The Use of Electronic Desk Top Computers and Time Study Equipment in Industrial Engineering, Annual Conference Proceedings, AIIE, pp. 522-528.

13. DeJong, Marvin L., 1980. Programming and Interfacing the 6502, with experiments, Howard W. Sams and Co., Inc. 414 p.

14. Eady, Karl, 1975. What is the MTM Family Really like? The Journal of Methods-Time Measurement, 1 2, pp. 42-55.

15. Foster, C. C., 1978. Programming a Microcomputer, 6502, Addison-Wesley Publishing Co., Inc. 231 p.

16. Garver, F. M., Block J. III, 1979. Generate Standards Information Faster, Industrial Engineering, 11 7, pp. 32-33.

17. Inman, D., Inman, K., 1981. Apple Machine Language, Reston Publishing Co., Inc. 206 p.

18. Inoue, M. S., 1981. What Should Q.C. Engineers Know About Microprocessors? Transactions, ASQC Quality Congress, San Francisco, CA pp. 603-610.

19. _______________, 1980. Introduction to Computer Science and Engineering, unpublished book draft. Six chapters.

20. Klein, K., 1970. A new way to take time studies, Industrial Engineering, 2 4, pp. 17-19.

21. Konz, S., 1979. Work Design, GRID Inc. 591 p.

22. Kopp, K. K., 1967. A Computer Program for Time Study Analysis, Journal of Industrial Engineering, 18 2, pp. 147-152.

23. Leventhal, L. A., 1979. 6502 Assembly Language Programming, Osborne/ McGraw-Hill. 16-29 p.

24. Magnusson, K., 1972. The Development of MTM-2, MTM-V and MTM-3, The Journal of Methods-Time Measurement, 17 1, pp. 11-23.

25. Marcy, W. M., 1980. Digital Electronics for Microprocessor Applications in Control of Manufacturing Processes, AIIE Transactions, 12 1, pp. 15-22.

26. Maynard, H. B., 1971. Industrial Engineering Handbook, Third Edition, McGraw-Hill, Inc. 13-118 p.

27. Mundel, M. E., 1978. Motion and Time Study, Improving Productivity, Fifth Edition, Prentice Hall, Inc. 750 p.

28. Nadler, G., 1963. Work Design, Richard D. Irwin, Inc. 837 p.

29. Niebel, B. W., 1976. Motion and Time Study. Richard D. Irwin, Inc. 719 p.

30. R6500 Microcomputer System, 1981. Versatile Interface Adapter Data Sheet, Revision 3.

31. R6500 Microcomputer System, 1981. Preparing an AIM 65 BASIC Program for PROM/ROM Operation, Application Note, Revision 2.

32. Rice, R. S., 1977. Survey of Work Measurement and Wage Incentives, Industrial Engineering, 9 7, pp. 18-31.

33. Robertensen, J. A., 1980. Multi-dimensional work sampling at the executive level, Industrial Engineering, 12 8, pp. 70-73.

34. Rockwell International, 1979. AIM 65 Microcomputer, BASIC Language Reference Manual, Revision 1.

35. ______________, 1978. AIM 65 Microcomputer Monitor Program Listing.

36. ____________, 1980. AIM 65 PROM Programmer & CO-EDF User's Manual.

37. ____________, 1979. AIM 65 Microcomputer User's Guide, Revision 3.

38. Ross, R., 1970. Computer Analyzes Data-Man Analyzes the Job. Industrial Engineering, 2 5, pp. 22-25.

39. Samanta, D., 1969. How a Tape Recorder Can Improve Work Measurement, Industrial Engineering, 1 10, pp. 24-27.

40. Scanlon, L. J., 1980. 6502 Software Design, Howard W. Sams & Co., Inc. 270 p.

41. Schmid, R. D., 1972. MTM Data Systems Application Survey, The Journal of Method-Time Measurement, 17 3, pp. 3-13.

42. Smalley, H. E., 1967. Another Look at Work Measurement, Journal of Industrial Engineering, 18 3, pp. 202-218.

43. Towne, D. M., 1979. Today's Computers and Work Measurement, The Journal of Methods-Time Measurement, 7 1, pp. 7-11.

44. Weaver, R. F., Kollmar, J. J., Boepple, E. A. Jr., 1978. Developing Standards by Computer. Industrial Engineering, 10 1, pp. 26-31.

45. Zaks, R., 1979. 6502 Applications Book, SYBEX Inc., 278 p.

46. ____________, 1980. 6502 Games, SYBEX Inc. 291 p.

47. Zandin, K. B., Weiss, R. M., 1977. Most Systems for Work Measurement, Industrial Engineering, 9 6, pp. 43-46.

APPENDICES

APPENDIX A

USER MANUAL

This manual describes the use of the stopwatch, MTM analysis, and simulator programs stored in the 4K ROM chip located in the Z24 position of the AIM 65 microcomputer. For each of these programs a step by step procedure is explained in order to aid in their use. The use of the Text Editor to create data files is also described. This will assist the analyst in creating different data files (i.e., MTM micromovements) directly from the keyboard using Text Editor commands. A procedure to save and load data files onto a cassette tape is described along with a procedure to program a ROM using the PROM/ROM programmer and CO-ED Module.

1. The Stopwatch Program

Using the stopwatch method the analyst has available the AIM 65 microcomputer keyboard to enter the real time observations from the field. The keys used to enter the observations are as follows:

Depress Key	To
0,1,2,3,4,5,6,7,8,9	enter the element number.
S	record when an element starts.
E	record when an element ends. The starting time used to calculate the discrete time is when the S or E key was depressed for the element.

C record when an element, whose starting time is the last time any key was depressed, ends.

F register when a foreign element starts or ends. (Push once when it starts, push again when it ends.)

K delete an observation. Each time this key is depressed it deletes the last observation registered. When it is pushed twice the last two observations are deleted and so forth.

DEL delete the element number in the display.

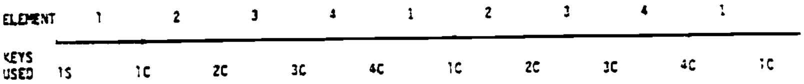
RETURN end the observation period.

Continuous and discrete elements can also be measured. A continuous element is an element which is part of a sequence of elements being measured. For example, if the elements one, two, and three are executed in sequence by the operator, each one is a continuous element. This situation is typical of short-cycled repetitive operations. A discrete element is an element which is being measured alone. For example, if the process time of a certain machine is all that is desired to be measured, the element (process time) will be discrete. Examples of the keys used to record the time of different elements are illustrated below.

NOTE: When taking the time of multiple operators simultaneously it is advisable to use only discrete elements. The reason for this is that the C key cannot be used to take a series of continuous elements for each operator. When an element starts immediately after another element finishes, a small amount of time will be missed. This small amount of time is the time spent in

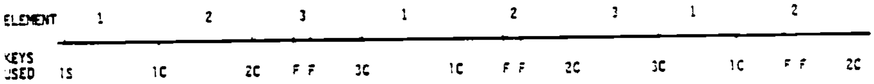
CASE A

A single operator
continuous elements (Usa S and C keys only)



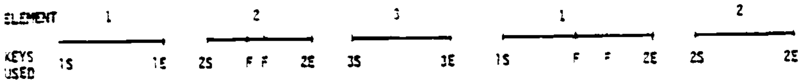
CASE B

A single operator
continuous and foreign elements (Use S, C, and F keys only)



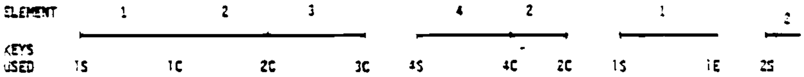
CASE C

A single operator
Discreze and foreign elements (use S, E, and F keys only)



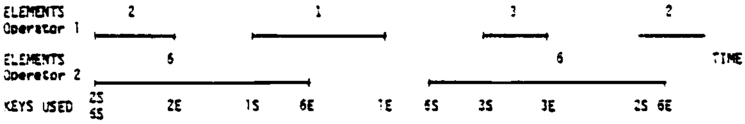
CASE D

Single operator
continuous and discrete elements (Use S, E, C, and F keys)



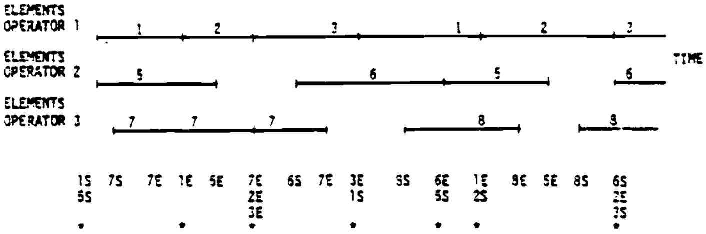
CASE E

Multiple operator
discrete elements (Use S, E, and F keys only)



CASE F

Multiple operator and/or machine
continuous and discrete times (Use S, E, and K keys only)



* ERROR INCLUDED

depressing the keys corresponding to the starting of the new element. When the element finishes, this amount of time missed will be included in the time recorded in the data file. This error is approximately 0.01213 minutes with a standard deviation of 0.00658 minutes. When only discrete elements are measured this error is minimized because no element is taken immediately after another finishes.

Procedure:

1. Depress the F1 key. When the display shows "ENTER MAX 3 CHR.", depress the corresponding keys depending on the element number to be measured. For instance, if the element to be measured is the number ten, enter "10".
2. Make sure the display shows the correct number. If an error exists depress the DEL key and then enter the element number again.
3. If the element starts, depress the S key. If the element is going to end, and it is a discrete element, depress the E key. If the element is going to end, and it is part of a sequence of elements being measured, be ready to depress the C key. When the C key is depressed, it serves as the end of the element being displayed plus the beginning of the next element. This means that when an element is finished using the C key, its starting time is taken as the last time any key was depressed.

If a foreign element appears before the end of the element being measured, depress the F key when the foreign element begins and wait until it ends to once again push the F key. Then enter in again the number of the element being measured.

NOTE: This last step must be done because the foreign elements are supposed to be a part of an element being studied. This means that the time of the foreign element is subtracted from the time of the regular element. The foreign element can also be treated as a regular independent element. If so, a different number is given to the foreign elements and it can be measured as regular. The drawback of this approach is that the foreign elements appear without previous knowledge so the precision under which they are measured decreases considerably.

4. When it is desired to delete the last observation taken, depress the K key. Each time the K key is depressed the last line of the data file is deleted.
5. Continue taking observations until the available memory is full (90 observations) or until no further observations are desired. Depress the RETURN key to end the data collection period. The microcomputer will print "EDIT OR SAVE FILE".
6. The data file can be edited or saved for future analysis. To edit or save the data file depress the T key which

is used to reenter the Text Editor. Use the Text Editor commands to edit the data file.

2. Analysis Program

The data file can now be analyzed to change the continuous time observations to discrete time observations, and to calculate the average time, standard deviation, and number of observations required for the precision desired. In order to do this the following procedure is followed.

1. Turn the printer on if a hard copy of the discrete time observations from the data file is desired.
2. Depress the N key to start running the program.
3. When requested to do so by the program, enter the number of elements in which the operation is divided. This should not be confused with the number of observations in the data file.
4. When "TIME UNIT = 1/100 MIN" is displayed,all the times printed or displayed from this point on are in 1/100 of a minute.
5. When requested to do so by the microcomputer, enter the precision desired. The confidence level is maintained at 95%. The program then calculates, displays, prints (if the printer is on) and changes the discrete times for each observation taken and recorded in the data file. The program also gives the total time spent taking the observations.

6. After the program has analyzed the data file, a summary of results are displayed. The summary of results consists of the average time, in hundreths of a minute, and the standard deviation for each element, including foreign elements. It also gives the number of observations required to meet the precision desired.

After these results are obtained, the analyst may continue taking more observations of the same elements being studied. This is done by following the procedure used to run the stopwatch program. Once a new data file has been created by the stopwatch program, the analysis program may be used again to add the new results onto the results of the previous data files. The only change to be made is in step two of the analysis program procedure described above. This change consists of depressing the 6 key instead of the N key. After depressing the 6 key, type the word "CONT" and depress the RETURN key. This begins the program running once again. By typing in the word "CONT", the results from the current data file analysis are added to the results of the previous data file analysis. This feature overcomes the drawback of the limited memory available for data file storage. The Industrial Engineer can continue taking observations and creating different data files without losing the results of previous data files.

3. Four Channel Real Time Multipurpose Simulator Program

This program is written in assembler object code. It has the capability of simulating four operations simultaneously. The program displays four characters for each channel on the AIM 65 display. The characters are displayed during the time indicated in the data file for each observation on each of four subjects. A channel is equivalent to one operation executed by a subject (hand, the body, a machine, or an operator). This means that any four operations can be simulated simultaneously. This is very important in order to compare methods and to verify the correct sequence of elements. It is also very applicable to train personnel and implement new improved methods.

Procedure

1. Push the F2 key to initialize the program.
2. When "ENTER C=CONT O=ONCE" is displayed, depress the C key to simulate the data file continuously. Otherwise, depress O to simulate the data file once again.
3. Use the RESET key to escape from the simulation program.

4. The MTM Program

This program allows the creation of a data file containing discrete times for MTM micromovements. The program itself has the data necessary to handle MTM-3 data files. If other MTM systems are used, the corresponding times for all of the micromovements must be entered. The program also allows the generation of simograms. Simograms are created by listing all the micromovements for each subject (i.e., left hand, right hand, legs, etc.). The inclusion of machine elements or other system elements is also permitted.

Procedure:

Depress the N to begin running the program. The program will bring a menu of keys used a) to create a MTM file (M key), b) to generate a simogram (6 key), c) to simulate the MTM file (U key), or d) to use the Text Editor (ESCAPE key).

a) Follow these steps to create a data file.

1. Enter M to begin running the program.
2. When "TITLE (Y-N)?" is displayed, enter N and continue with step 4 only if a title line for the data file is <u>not</u> desired. If a title line is desired enter Y.
3. When the program prints "ENTER MAX 15 CHARS", enter a maximum of fifteen characters corresponding to the title desired for the data file.
4. When "SUBJECT #" is displayed, enter the subject that executes the element. The numbers allowed are between 1 and 99. For example, if the MTM analysis uses the left hand, right hand, body, and machine the following might be a sample of the element subjects.

   1 - left hand
   2 - right hand
   3 - body
   4 - machine

5. Once the ele. subject is entered the program displays "MAX 4 CHR NAME". Now enter a maximum of four characters to represent the name or number of the element. For example, enter R32C if that is the name of the micromovement.
6. When using the MTM-3 method, the program will automatically calculate the time for the element entered. If this is the case continue on to step 8.
7. When the program prints "4 CHRS (from step 5) TIME(TMU) = ?", enter the number of TMU's for the element time.
8. The program will one again request the subject, name, and time of the next micromovement. The memory capacity assigned to the data file allows a maximum of 134 observations to be entered. When "SUBJECT #" is displayed, enter "END" if no further micromovements are to be entered.
9. The program will print the number of micromovements entered along with "EDIT, LIST, SAVE FILE". It will also print the menu once again.

b) Follow these steps to generate a simogram.

1. Enter G to begin running the program.
2. A list of the micromovements and the total TMU time is obtained for each subject entered. For example, subject #1 will be the left hand as defined in step 4.

3. After all the micromovements have been listed the program will return to the menu.

c) Follow these steps to simulate a MTM data file.

1. Enter U and then enter the speed coefficient for each subject as requested by the program. This is done only if a change in the simulation speed is required.
2. Enter O for a single shot simulation or C for continuous simulation once "PUSH O=ONCE C=CONT" is displayed.
3. Depress the ESCAPE or RESET key to exit the simulation program and then depress the N key to reenter the menu.

5. Direct Data File Creation Using the Text Editor

The Text Editor can be used to create a data file directly from the keyboard. Once the analyst has the MTM simochart, the normal time, or the standard time of the elements prepared, the Text Editor can be initialized to start putting the information into the data file.

Procedure

1. Depress the RESET key to send control to the monitor.
2. Depress the E key to initialize the editor.
3. Enter 0700 when "FROM" is displayed.
4. Enter OFFF when "TO" is displayed. This defines the memory size for the Text Buffer in which the data file is going to be stored.

This means that we are assigning the space for memory from 0700 to OFFF (hexadecimal) for the data file. This space consists of 2504 bytes which allows a maximum of 135 observations in RAM (17 bytes per observation).

5. When the display prints "IN=",depress the RETURN key.
6. At this stage the editor is ready to accept data. Start entering the data. There are eight bytes for the element description and another eight bytes for the element time (Time Unit = MU's) for each observation. Mistakes can be deleted using the DEL key. Depress the RETURN key at the end of each element observation.
7. Depress the RETURN key when all the data has been entered.
8. Use the Text Editor commands to verify the data file.
9. Use the ESCAPE key to exit the editor.
10. Once the data file is created the T key can be used to reenter the editor and reedit the data file.

6. Procedure to Save a Data File on Cassette Tape (The computer cassette tape from Radio Shack Cat.No. 26-301 has given satisfactory results.)

1. Depress the T key to reenter the Text Editor.
2. Depress L to indicate that the data file is going to be saved or listed.
3. Depress the period key "." to indicate that the entire data file is going to be saved. The microcomputer will respond by printing "OUT=".

4. If the data file is going to be listed on the printer, depress the RETURN key. Make sure the printer is on by keeping the CONTROL key depressed while at the same time depressing the PRINT key. The printer switches from ON to OFF or vice versa each time the PRINT key is depressed.

   If the data file is going to be saved on a cassette tape, depress T. When the display shows "F=," enter the five character file name. When the program displays "T=" enter the tape recorder number (1 for tape recorder #1 or 2 for tape recorder #2).

   NOTE: If the file is to be recorded on audio cassette tape for substantial input to the assembler or as added text to a partially loaded Text Buffer, the GAP parameter in address $A409 must be changed to $80 in order to provide a large gap between recorded data blocks. (Source: AIM 65 user's guide.)
5. Depress the RECORD and PLAY key of the tape recorder. Depress the RETURN key of the microcomputer. The data block number being saved is displayed in the last two positions of the microcomputer display.
6. Once the display of the AIM 65 displays "END", stop the tape recorder and take note of the counter.
7. Exit the Text Editor by depressing the ESCAPE key.

7. Procedure to Load a Data File From a Cassette Tape to a Text Editor Buffer

a) Loading a data file created by the stopwatch program

(Continuous time data file or discrete time data file)

1. Depress the ESCAPE key to send control to the monitor.
2. Depress the E key to initialize the Text Editor.
3. Enter "OAOO" (hexadecimal) when "FROM=" is displayed, This indicates the initial address of the Text Buffer.
4. Enter "OFFF" (hexadecimal) when "TO=" is displayed. This indicates the final address of the Text Buffer. Its capacity allows a maximum of 90 lines to be entered.
5. Enter T when "IN=" is displayed. This indicates that the tape recorder is the data being input.
6. When the program prints "F=" enter the five character file name.
7. Enter the tape recorder number when "T=" is displayed. (The default value is one.)
8. Locate the beginning of the file on the cassette tape (five inches behind).
9. Depress the RETURN key to let the microcomputer search the file and then depress the PLAY key of the tape recorder.
10. When the microcomputer reaches the data file it will begin loading the data onto the Text Editor Buffer. It will also display the block data number as it loads the file.

11. When finished, the microcomputer will display "END". The control is now in the Text Editor. Depress the ESCAPE key to get out of the Text Buffer. If desired, the Text Editor commands can be used to edit the data file.

b) Loading a data file created by the analysis program. (Standard time data file)

The procedure for loading a data file created by the analysis program is the same as the procedure for loading a data file created by the stopwatch program as described above. The only difference is that in step 3 the "0700" (hexadecimal) is used instead of the "0A00".

8. Procedure to Program a 4K Bytes PROM on the PROM/ROM Programmer (PROM = TMS 2532).

NOTE: In the following procedure it is assumed that the object code has been previously prepared for PROM programming and is saved on a cassette tape.

1. Initialize the firmware functions by entering the following underlined commands.

```
(*) = 9000   RETURN
(G) / .
```

The AIM 65 will print:

```
PROM PROGRAMMER
AND CO-ED USER 1.0
```

2. Depress the F1 key and enter 2532 as the PROM type used. Specify the PROM Base Address as $D000.

3. Fill the memory with $00 using the M command.

```
ENTER COMMAND =M=
MEM FILL = 00
FROM = D000
TO = DFFF
DONE
```

4. Using the L command load the memory with the object code from the cassette tape.

```
ENTER COMMAND =L=
LOAD OBJECT
IN =T F= EXOBJ T=1

READING AT #### (#### = initial address)
DONE
```

5. Install the 2532 PROM in the PROM/ROM Programmer module by pulling the socket level up, placing the PROM into the PROM socket, and then pushing the socket level down.

6. Program the PROM data over the desired PROM address range with the P command.

```
ENTER COMMAND =P=
PROGRAM
ERROR LIST = P

FROM = D000
TO = DFFF
********************
********************
********************
******DONE
VERIFYING
DONE
```

7. Depress the ESCAPE key, turn off the AIM 65 and remove the PROM from the PROM/ROM programmer module.

9. Procedure to Insert the PROM Chip Onto the AIM 65 Microcomputer.

1. Turn off the microcomputer.
2. Before touching the PROM or the microcomputer, discharge any static electrical charge in your body by touching a ground connection.
3. Make sure that the EPROM's 24 pins are straight and firm. Insert the chip in the Z24 position by firmly applying pressure to all angles of its surface. The half circle shaped hole of the chip should be facing the printer.
4. Turn the microcomputer on and depress the N key to start running the ROM program.

APPENDIX B

COMPUTER PROGRAM LISTING

1 - Stopwatch Program. This program is written in assembler. It must initially be entered in the Text Editor and then assembled in order to obtain the object code.

```
OUT=
*=$0A41
LDA #>INT
STA $A401
LDA #<INT
STA $A400
LDA #00
STA $A411
STA $81
STA $84
STA $214
STA $E1
STA $85
JSR $EA14
LDA #01
STA $04
LDA #10
STA $E4
STA $05
STA $8C
LDA #$FF
STA $E5
LDA #$0F
STA $E6
LDA #$30
LDX #01
STX $21F
ADA STA $217,X
INX
CPX #9
BNE ADA
LDA #$C0
STA $A00E
LDA #$40
STA $A00B
CLI
LDA #$3E
STA $A004
STA $A005
DCB LDA $214
```

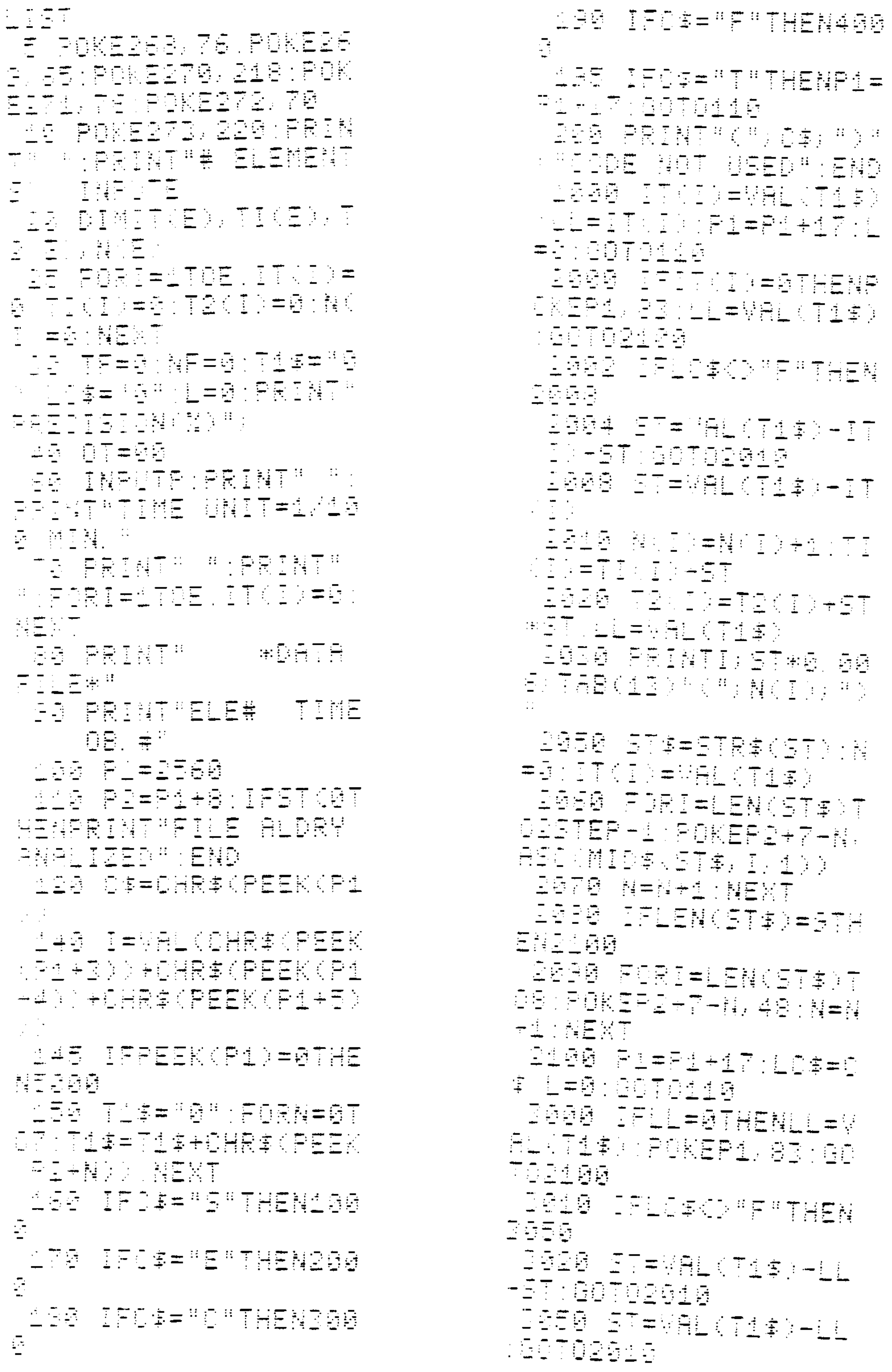
```
CMP #90
BNE FEA
JMP TURN
FEA LDY #00
LDX #00
UNO LDA MES,X
JSR $E97A
INX
CPX #13
BNE UNO
LDY #00
LDA #$31
STA ($04,X)
CLC
LDA $04
ADC #01
STA $04
LDA $05
ADC #00
STA $05
LDA #$20
STA ($04,X)
CLC
LDA $04
ADC #01
STA $04
LDA $05
ADC #00
STA $05
INP JSR $E973
LDX #00
CMP #$7F
BNE KTE
JMP LLL
KTE CMP #$'K
BNE SSS
JMP KKK
SSS CMP #$'S
BNE CTES
```

```
JMP ESE
CTES CMP #$7C
BNE FIN
JMP CEE
FIN CMP #$7E
BNE RETU
JMP EEE
RETU CMP #$0D
BNE ETE
JMP TURN
ETE CMP #$7F
BNE NONE
STA ($85,X)
JMP YES
NONE STA ($04,X)
LDA $04
CLC
ADC #$01
STA $04
LDA $05
ADC #$00
STA $05
INY
JMP INP
ESE LDA #$75
STA ($85,X)
JMP YES
TRES LDA #$20
STA ($04,X)
LDA $04
CLC
ADC #$01
STA $04
LDA $05
ADC #$00
STA $05
INY
YES CPY #5
BNE TRES
LDY #00
CINC LDA $0218,Y
STA ($04,X)
LDA $04
CLC
ADC #$01
STA $04
LDA $05
ADC #$00
STA $05
INY
CPY #08
BNE CINC
```

```
LDA #$0D
STA ($04,X)
LDA $04
CLC
ADC #$01
STA $04
STA $85
LDA $05
ADC #$00
STA $86
STA $05
INC $214
RIC LDA $04
CLC
ADC #$01
STA $04
LDA $05
ADC #$00
STA $05
ALA JSR $EB44
JMP DOS
KKK LDA $214
CMP #00
BEQ ALA
LDA $85
SEC
SBC #$11
STA $04
STA $85
LDA $86
SBC #00
STA $05
STA $86
DEC $214
JMP RIC
LLL LDA $85
STA $04
LDA $86
STA $05
JMP RIC
CEE LDA #$7C
STA ($85,X)
JMP YES
EEE LDA #$7E
STA ($85,X)
JMP YES
TURN LDA #$00
STA ($85,X)
LDA $85
STA $E1
LDA $86
STA $E2
```

```
JSR $EB44
JSR $EA24
LDY #00
LDA #123
STA $A411
CAMA LDA SILA,Y
JSR $E97A
INY
CPY #18
BNE CAMA
JSR $EA24
LDA #70
STA $81
STA $83
LDA #$83
STA $85
LDA #$D1
STA $86
SEI
LDA #0
STA $A400
STA $A401
JMP $EA13
MES .BYT 'ENTER MAX
I CHR # '
SILA .BYT ' EDIT OR
SAVE FILE'
INT PHA
TXA
PHA
CLC
SED
LDX #06
OSA LDA $217,X
ADC $215
CMP #$60
STA $217,X
BNE LEON
LDA #$30
STA $217,X
LEON LDA #00
STA $215
DEX
BNE OSA
INC $215
LDA $A004
CLD
PLA
TAX
PLA
RTI
.END
```

2 - Analysis Program (BASIC Language).

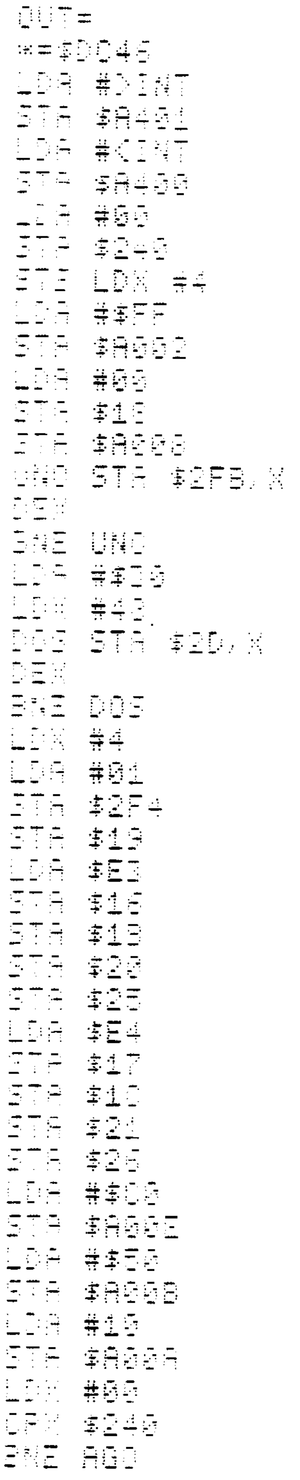


```
LIST
 5 POKE268,76:POKE26
9,55:POKE270,218:POK
E271,78:POKE272,70
 10 POKE273,220:PRIN
T" ":PRINT"# ELEMENT
S":INPUTE
 12 DIMIT(E),TI(E),T
2(E),N(E)
 15 FORI=1TOE:IT(I)=
0:TI(I)=0:T2(I)=0:N(
I)=0:NEXT
 30 TF=0:NF=0:T1$="0
":L1$="0":L=0:PRINT"
PRECISION(%)";
 40 OT=00
 50 INPUTP:PRINT" ":
PRINT"TIME UNIT=1/10
0 MIN."
 70 PRINT" ":PRINT"
":FORI=1TOE:IT(I)=0:
NEXT
 80 PRINT"     *DATA
FILE*"
 90 PRINT"ELE#  TIME
     OB.#"
 100 P1=2560
 110 P2=P1+8:IFST<0T
HENPRINT"FILE ALDRY
ANALIZED":END
 120 C$=CHR$(PEEK(P1
))
 140 I=VAL(CHR$(PEEK
(P1+3))+CHR$(PEEK(P1
+4))+CHR$(PEEK(P1+5)
))
 145 IFPEEK(P1)=0THE
N3000
 150 T1$="0":FORN=0T
O7:T1$=T1$+CHR$(PEEK
(P1+N)):NEXT
 160 IFC$="S"THEN100
0
 170 IFC$="E"THEN200
0
 180 IFC$="C"THEN300
0
 190 IFC$="F"THEN400
0
 195 IFC$="T"THENP1=
P1+17:GOTO110
 200 PRINT"(";C$;")"
;"CODE NOT USED":END
 1000 IT(I)=VAL(T1$)
:LL=IT(I):P1=P1+17:L
=0:GOTO110
 2000 IFIT(I)=0THENP
OKEP1,83:LL=VAL(T1$)
:GOTO2100
 2002 IFLC$<>"F"THEN
2008
 2004 ST=VAL(T1$)-IT
(I)-ST:GOTO2010
 2008 ST=VAL(T1$)-IT
(I)
 2010 N(I)=N(I)+1:TI
(I)=TI(I)+ST
 2020 T2(I)=T2(I)+ST
*ST:LL=VAL(T1$)
 2030 PRINTI;ST*0.06
;TAB(13)"(";N(I);")"

 2050 ST$=STR$(ST):N
=0:IT(I)=VAL(T1$)
 2060 FORI=LEN(ST$)T
O2STEP-1:POKEP2+7-N,
ASC(MID$(ST$,I,1))
 2070 N=N+1:NEXT
 2080 IFLEN(ST$)=9TH
EN2100
 2090 FORI=LEN(ST$)T
O8:POKEP2+7-N,48:N=N
+1:NEXT
 2100 P1=P1+17:LC$=C
$:L=0:GOTO110
 3000 IFLL=0THENLL=V
AL(T1$):POKEP1,83:GO
TO2100
 3010 IFLC$<>"F"THEN
3050
 3020 ST=VAL(T1$)-LL
-ST:GOTO2010
 3050 ST=VAL(T1$)-LL
:GOTO2010
```

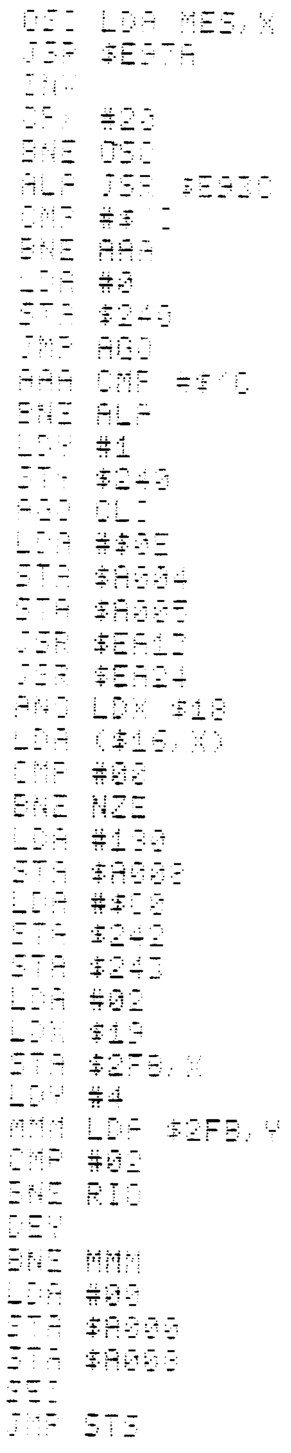
```
 4000 IFL=1THEN4500
 4010 F1=VAL(T1$):L=
1:POKEP1,83:P1=P1+17
:GOTO110
 4500 ST=VAL(T1$)-F1
:NF=NF+1:TF=TF+ST:AF
=TF/NF
 4510 PRINT"F. ";ST*.
006;TAB(13)"(";NF;")
":GOTO2050
 5000 POKE42001,128:
PRINT" ":OT=OT+VAL(T
1$)*.006:PRINT"
SUMMARY"
 5005 PRINT" ":PRINT
"OBS.TIME=";OT:PRINT
" "

 5010 PRINT"ELE#   ";
"AVG.T & STD"
 5020 FORI=1TOE:IFN(
I)=0THEN5050
 5025 IFN(I)=1THENSD
=0:GOTO5040
 5030 SD=INT(SQR((T2
(I)-TI(I)*TI(I)/N(I)
)/(N(I)-1))*.6)
 5040 PRINTI;TAB(4)T
I(I)/N(I)*.006;SD/10
0
 5050 NEXT
 5056 PRINT"F. ";TAB(
4)AF*.006
 5060 PRINT" ":PRINT
"ELE# OB.NOW OB.REQ"
 5070 FORI=1TOE:IFN(
I)=0THEN5092
 5075 IFN(I)=1THENPR
INTI;TAB(6)N(I);"
---":GOTO5092
 5080 PRINTI;TAB(6)N
(I);"    ";
 5085 SD=SQR((T2(I)-
TI(I)*TI(I)/N(I))/(N
(I)-1))
 5090 PRINTTAB(6)INT
((2/P*100*N(I)*SD/TI
(I))^2)+1
 5092 NEXT:PRINT"F. "
;TAB(6)NF
 5100 PRINT" ":PRINT
"DATA FILE(Y-N)";:IN
PUTC$:IFC$="Y"THEN51
80
```

```
 5175 POKE129,236:PO
KE130,19:POKE131,236
:POKE132,19
 5176 POKE133,104:PO
KE134,215:END
 5180 POKE42001,128:
PRINT" ":PRINT"EL. R
T% ALW STD.TIME":POK
E42001,0
 5200 P1=1792:POKE22
7,0:POKE228,7:POKE22
9,255:POKE230,9
 5250 FORI=0TO767:PO
KEP1+I,32:NEXT:Z=0
 5300 FORI=1TOE:IFN(
I)=0THEN5550
 5350 B$=STR$(I):FOR
P=1TOLEN(B$):POKEP1+
2-P,ASC(MID$(B$,P,1)
)
 5370 NEXT:PRINT"#";
I;"RATING(%)=";:INPU
TR:PRINT"#";I;"ALLOW
ANCE(%)=";
 5371 INPUTW:T=TI(I)
/N(I)*(R/100)/((100-
W)/100)
 5375 POKE42001,128:
PRINTI;TAB(3)R;TAB(8
)W;T*.006:POKEP1,69
 5376 POKEP1+1,49:T$
=STR$(INT(T)):C=0:Z=
Z+T:POKE42001,0
 5400 FORP=LEN(T$)TO
1STEP-1:POKEP1+15-C,
ASC(MID$(T$,P,1))
 5450 C=C+1:NEXT:IFL
EN(T$)=9THEN5500
 5475 FORP=LEN(T$)TO
9:POKEP1+15-C,48:C=C
+1:NEXT
 5500 POKEP1+16,13:P
1=P1+17
 5550 NEXT:PRINT" ":
POKE42001,128:PRINT"
TOT TIME=";Z*.006:PO
KEP1,0
 5551 :POKE226,INT(P
1/256):POKE42001,0
 5555 POKE225,(P1/25
6-INT(P1/256))*256:P
RINT" FILE CREATED":
END
```

3 - Simulation Program (Assembler Language).



```
OUT=
*=$DC45
LDA #>INT
STA $A401
LDA #<INT
STA $A400
LDA #00
STA $240
STS LDX #4
LDA #$FF
STA $A002
LDA #00
STA $18
STA $A000
UNO STA $2FB,X
DEX
BNE UNO
LDA #$30
LDX #42
DOS STA $2D,X
DEX
BNE DOS
LDX #4
LDA #01
STA $2F4
STA $19
LDA $E3
STA $16
STA $1B
STA $20
STA $25
LDA $E4
STA $17
STA $1C
STA $21
STA $26
LDA #$C0
STA $A00E
LDA #$50
STA $A00B
LDA #10
STA $A00A
LDX #00
CPX $240
BNE AGO
```



```
OSC LDA MES,X
JSR $E97A
INX
CPX #23
BNE OSC
ALP JSR $E93C
CMP #$'I
BNE AAA
LDA #0
STA $240
JMP AGO
AAA CMP #$'C
BNE ALP
LDY #1
STY $240
AGO CLI
LDA #$0E
STA $A004
STA $A005
JSR $EA13
JSR $EA24
AND LDX $18
LDA ($16,X)
CMP #00
BNE NZE
LDA #130
STA $A00E
LDA #$C0
STA $242
STA $243
LDA #02
LDX #19
STA $2FB,X
LDY #4
MMM LDA $2FB,Y
CMP #02
BNE RIO
DEY
BNE MMM
LDA #00
STA $A000
STA $A008
SEI
JMP STS
```

```
RII LDX $18
LDA #$20
ORA #$80
JSR $EF7B
INC $18
LDX $18
ORA #$80
JSR $EF7B
INC $18
LDX $18
ORA #$80
JSR $EF7B
INC $18
LDX $18
ORA #$80
JSR $EF7B
DEC $18
DEC $18
DEC $18
JMP COM
NZE CMP #$'E
BNE CCC
JMP CHA
CCC CMP #$'C
BNE FFF
JMP CHA
FFF CMP #$'F
BEQ CHA
GAP LDX $18
LDA $16,X
CLC
ADC #17
STA $16,X
LDA $17,X
ADC #00
STA $17,X
JMP ANC
CHA LDX $18
LDA $16,X
CLC
ADC #1
STA $16,X
LDA $17,X
ADC #00
STA $17,X
LDA ($16,X)
TAY
SEC
LDA $16,X
SBC #1
STA $16,X
LDA $17,X
```

```
SBC #00
STA $17,X
TYA
STX $1F
SEC
SBC #$30
TAX
CPX $19
BEQ CDO
JMP GAP
CDO STX $19
CPX #1
BNE NDO
LDA #8
BIT $A000
BEQ A1A
LDA $A000
AND #7
STA $A000
JMP AA1
A1A ORA $A000
STA $A000
AA1 LDA #00
STA $1C
STA $20
STA $1E
LDA #$36
STA $1D
LDA #$2E
STA $21
JMP SRE
NDO CPX #2
BNE NTR
LDA #4
BIT $A000
BEQ A2A
LDA $A000
AND #11
STA $A000
JMP AA2
A2A ORA $A000
STA $A000
AA2 LDA #5
STA $1B
LDA #$46
STA $1D
LDA #00
STA $1E
STA $20
LDA #$3E
STA $21
JMP SRE
```

```
NTR CPX #3
BNE NCU
LDA #2
BIT $A000
BEQ A3A
LDA $A000
AND #13
STA $A000
JMP AA3
A3A ORA $A000
STA $A000
AA3 LDA #10
STA $19
LDA #$56
STA $1D
LDA #00
STA $1E
STA $23
LDA #$4E
STA $22
JMP SRE
NCU CPX #4
BEQ TOR
JSR $EA24
JSR $E97A
LDA #$3F
JMP $E97A
TOR LDA #15
STA $19
LDA #1
BIT $A000
BEQ A4A
LDA $A000
AND #14
STA $A000
JMP AA4
A4A ORA $A000
STA $A000
AA4 LDA #$E8
STA $1D
LDA #$02
STA $1E
STA $23
LDA #$E0
STA $22
SRE LDX $18
LDA $16,X
CLC
ADC #8
STA $16,X
LDA $17,X
ADC #00
```

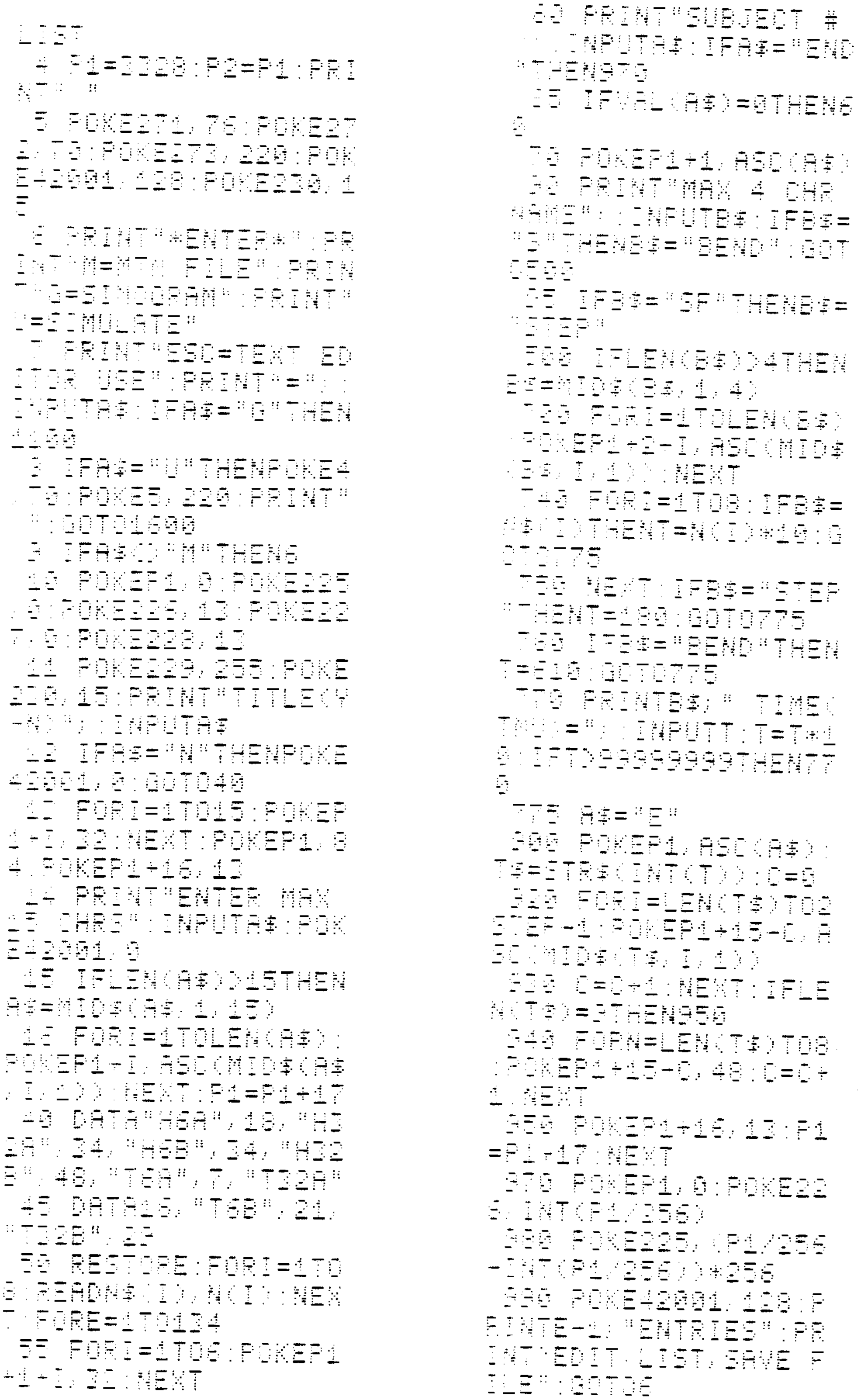
```
STA $17,X
LDY #00
SM1 LDA ($16,X)
STA ($1D),Y
LDA $16,X
CLC
ADC #1
STA $16,X
LDA $17,X
ADC #00
STA $17,X
INY
CPY #8
BNE SM1
LDX $18
LDA $16,X
SEC
SBC #13
STA $16,X
LDA $17,X
SBC #0
STA $17,X
LDA #250
STA $A008
LDY #00
STA $242
STA $243
LDX $19
LDA #$30
CLT STA ($22),Y
INY
CPY #8
BNE CLT
LDX $19
INC $2FB,X
LDY #0
LDX $18
STX $1A
PRT LDA ($16,X)
LDX $1A
ORA #$80
JSR $EF7B
LDX $18
LDA $16,X
CLC
ADC #1
STA $16,X
LDA $17,X
ADC #00
STA $17,X
INC $1A
INY
```

```
CPY #4
BNE PRT
LDA $16,X
CLC
ADC #10
STA $16,X
LDA $17,X
ADC #00
STA $17,X
COM LDX #00
CLC
LDA $243
ADC #1
STA $243
LDA $241
ADC #00
STA $242
LDA $242
CMP #255
BNE CHE
LDA #00
STA $A000
CHE INX
CPX #5
BEQ COM
LDA $2FB,X
CMP #0
BNE CHE
STX $13
CPX #1
BNE UL2
LDA #00
STA $18
JMP ANO
UL2 CPX #2
BNE UL3
LDA #5
STA $18
JMP ANO
UL3 CPX #3
BNE UL4
LDA #10
STA $18
JMP ANO
UL4 LDA #15
STA $18
JMP ANO
MES .BYT 'PUSH  O=ON
CE  C=CONT'
INT PHA
TXA
PHA
```

```
TYA
PHA
SED
LDY #1
ISU LDA $2FB,X
CMP #00
BNE CAR
JMP NLI
CAR CMP #02
BNE SIG
JMP NLI
SIG CPX #1
BNE PC2
LDA #$2D
STA $5
LDA #$35
STA $85
LDA #0
STA $6
STA $86
JMP IP1
PC2 CPX #2
BNE PC3
LDA #$3D
STA $5
LDA #$45
STA $85
LDA #00
STA $6
STA $86
JMP IP1
PC3 CPX #3
BNE PC4
LDA #$4D
STA $5
LDA #$55
STA $85
LDA #00
STA $6
STA $86
JMP IP1
PC4 CPX #4
BEQ NIN
JMP FIN
NIN LDA #$DF
STA $5
LDA #$E7
STA $85
LDA #2
STA $6
STA $86
IP1 LDY #28
```

```
CLC
OSA LDA ($5),Y
ADC $2F4
CMP #$40
STA ($5),Y
BNE LEON
LDA #$30
STA ($5),Y
LEON LDA #00
STA $2F4
DEY
BNE OSA
INC $2F4
LDA #00
STA $2F8
LDY #1
ASA LDA ($5),Y
CMP ($85),Y
BNE QUE
INC $2F8
QUE INY
CPY #9
BNE ASA
LDA $2F8
CMP #8
BNE NLI
LDA #05
STA $2FB,X
NLI INX
FIN CPX #5
BEQ TER
JMP ISU
TER LDA $A004
CLD
PLA
TAY
PLA
TAX
PLA
RTI
END
```

4 - MTM Program (BASIC Language)



```
LIST
 4 P1=3328:P2=P1:PRI
NT" "
 5 POKE271,76:POKE27
2,73:POKE273,220:POK
E42001,128:POKE230,1
5
 6 PRINT"*ENTER*":PR
INT"M=MTM FILE":PRIN
T"G=SIMOGRAM":PRINT"
U=SIMULATE"
 7 PRINT"ESC=TEXT ED
ITOR USE":PRINT"=";:
INPUTA$:IFA$="G"THEN
1100
 8 IFA$="U"THENPOKE4
,73:POKE5,220:PRINT"
":GOTO1600
 9 IFA$<>"M"THEN6
 10 POKEP1,0:POKE225
,0:POKE226,13:POKE22
7,0:POKE228,13
 11 POKE229,255:POKE
230,15:PRINT"TITLE(Y
-N)";:INPUTA$
 12 IFA$="N"THENPOKE
42001,0:GOTO40
 13 FORI=1TO15:POKEP
1+I,32:NEXT:POKEP1,8
4:POKEP1+16,13
 14 PRINT"ENTER MAX
15 CHRS":INPUTA$:POK
E42001,0
 15 IFLEN(A$)>15THEN
A$=MID$(A$,1,15)
 16 FORI=1TOLEN(A$):
POKEP1+I,ASC(MID$(A$
,I,1)):NEXT:P1=P1+17
 40 DATA"H6A",18,"H3
2A",34,"H6B",34,"H32
B",48,"T6A",7,"T32A"
 45 DATA15,"T6B",21,
"T32B",23
 50 RESTORE:FORI=1TO
8:READN$(I),N(I):NEX
T:FORE=1TO134
 55 FORI=1TO6:POKEP1
+1+I,32:NEXT
 60 PRINT"SUBJECT #
";:INPUTA$:IFA$="END
"THEN970
 65 IFVAL(A$)=0THEN6
0
 70 POKEP1+1,ASC(A$)
 80 PRINT"MAX 4 CHR
NAME";:INPUTB$:IFB$=
"S"THENB$="BEND":GOT
O700
 85 IFB$="SF"THENB$=
"STEP"
 700 IFLEN(B$)>4THEN
B$=MID$(B$,1,4)
 720 FORI=1TOLEN(B$)
:POKEP1+2+I,ASC(MID$
(B$,I,1)):NEXT
 740 FORI=1TO8:IFB$=
N$(I)THENT=N(I)*10:G
OTO775
 750 NEXT:IFB$="STEP
"THENT=180:GOTO775
 760 IFB$="BEND"THEN
T=610:GOTO775
 770 PRINTB$;" TIME(
TMU)=";:INPUTT:T=T*1
0:IFT>9999999THEN77
0
 775 A$="E"
 900 POKEP1,ASC(A$):
T$=STR$(INT(T)):C=0
 920 FORI=LEN(T$)TO2
STEP-1:POKEP1+15-C,A
SC(MID$(T$,I,1))
 930 C=C+1:NEXT:IFLE
N(T$)=9THEN950
 940 FORN=LEN(T$)TO8
:POKEP1+15-C,48:C=C+
1:NEXT
 950 POKEP1+16,13:P1
=P1+17:NEXT
 970 POKEP1,0:POKE22
6,INT(P1/256)
 980 POKE225,(P1/256
-INT(P1/256))*256
 990 POKE42001,128:P
RINTE-1;"ENTRIES":PR
INT"EDIT,LIST,SAVE F
ILE":GOTO6
```

```
 1100 PRINT" ":P1=I2
25:P2=P1:S1=0:FORI=1
TO10:T(I)=0:NEXT
 1102 C=PEEK(P1):IFC
=84THEN1108
 1103 IFC=0THEN1109
 1104 S=VAL(CHR$(PEE
K(P1+1))+CHR$(PEEK(P
1+2))):IFS>S1THENS1=
S
 1108 P1=P1+17:GOTO1
102
 1109 P1=P2:FORE=1TO
S1:PRINT"     SUBJ. #
";E:PRINT" ELEM. TIM
E-TMU,D&C"
 1110 C=PEEK(P1):IFC
=0THEN1500
 1150 S=VAL(CHR$(PEE
K(P1+1))+CHR$(PEEK(P
1+2)))
 1180 I$=" ":FORN=3T
O7:I$=I$+CHR$(PEEK(P
1+N)):NEXT
 1200 T1$="0":FORN=0
TO7:T1$=T1$+CHR$(PEE
K(P1+8+N)):NEXT
 1250 IFC=84THEN1400
 1270 IFS<>ETHEN1400
 1290 PRINTI$;TAB(6)
VAL(T1$)/10;:T(E)=T(
E)+VAL(T1$):PRINTT(E
)/10
 1400 P1=P1+17:GOTO1
110
 1500 P1=P2:PRINT"TO
T. TIME=";T(E)/10:PR
INT" ":PRINT" ":NEXT
:GOTO6
 1600 PRINT"SPEED CH
ANGE(Y-N)";:INPUTA$:
IFA$="N"THEN1690
 1620 PRINT"ENTER SP
EED COEFFCNT FOR EAC
H SUBJ.(%)":FORI=1TO
4
 1630 PRINT"SUBJ.#";
I;" %";:INPUTT(I):NE
XT:P1=P2
 1640 C=PEEK(P1):IFC
=84THEN1680
 1645 IFC=0THEN1690
 1650 S=VAL(CHR$(PEE
K(P1+1))+CHR$(PEEK(P
1+2))):IFS>4THEN1680
 1655 T1$="0":FORN=0
TO7:T1$=T1$+CHR$(PEE
K(P1+8+N)):NEXT
 1658 FORI=8TO15:POK
EP1+I,48:NEXT:C=0
 1660 T1$=STR$(INT(V
AL(T1$)/(T(S)/100)+.
5))
 1666 FORI=LEN(T1$)T
O2STEP-1:POKEP1+15-C
,ASC(MID$(T1$,I,1))
 1669 C=C+1:NEXT
 1680 P1=P1+17:GOTO1
640
 1690 POKE42001,0:I=
USR(1)
 1695 GOTO6
```

APPENDIX C

SURVEY

The presentation letter, a copy of the survey and the responses obtained from the companies surveyed are presented.

January 9, 1981

Dear Sir,

The following questions are for the purpose of collecting data about the uses of computerized systems on work measurement. This information is to be used only as part of my thesis titled "Applications of Microprocessor in Work Measurement".

I would appreciate the valuable time you could spend in answering my questions and returning them to me.

Sincerely,

Rodolfo Blanco

Graduate Student in Industrial Engineering

Oregon State University

Name______ ______________ Company____________________ __________ Address__________________ ______

Title__________________ Number of Employees____________ Main Product____________________

N. of Industrial Engineers__________

We never have used or done it before.	We use or do it but manually.	We use or do it using computer programs developed inside the company.	We use or do it using computer programs purchased outside the company.	We would like to see it computerized.		When was it introduced?	Satisf. or Unsatis. with Results	It could be improved with Com. Programs
					Time Study of Direct Labor using Stop-watch.			
					Time Study of Machine Utilization using Stop-watch.			
					Work Sampling of Direct labor.			
					Work Sampling of Indirect labor.			
					M.T.M. in Direct Labor.			
					Work-factor in Direct labor.			
					Other Predetermined Time Systems in Direct labor.(Specify ____________)			
					M.T.M. in Indirect labor.			
					Work-factor in Indirect Labor.			
					Other Predetermined Time Syetems in Indirect labor.(Specify ________)			
					Standard Data Systems.(Specify ____________________)			
					Electronic Time Measurement Devices.(Specify ______________)			
					Estimating and Costing using Work Measurement.			
					Wage Incentives using Work Measurement.			
					Production Scheduling using Work Measurement.			
					Justifying Numerical Control Machines using Work Measurement.			
					Justifying Robotics using Work Measurement.			
					Equipment Utilization using Work Measurement.			
					Work Measurement for other usage.(Specify ______________)			
					Video Tape/TV Monitor.			
					Memomotion or Micromotion using Camera.			

What problems should Industrial Engineers be addressing in Work Measurement?

How Micro-electronics and Computer Devices may be helpful in your Work Measurement and Work Design?

-Please use the other side for the answers- Thank you.

Name PHIL WAMBA Company BOEING COMM. FAB. DIVN Address PO BOX 3707 M/S 15-59 A5400 SEATTLE WA. 98124

Title M[illegible] . MFG MGT SYS. Number of Employees 12 [illegible] Main Product AIRFRAME PAR

N. of Industrial Engineers 300

We never have used or done it before.	We use or do it but manually.	We use or do it using computer programs developed inside the company.	We use or do it using computer programs purchased outside the company.	We would like to see it computerized.		When was it introduced ?	Satisf. or Unsatis. with Results	It could be improved with Com Programs
	X				Time Study of Direct Labor using Stop-watch.	1950's	SATIS.	DATA RETR...
X		[illegible]			Time Study of Machine Utilization using Stop-watch.	~~[illegible]~~		
	X				Work Sampling of Direct Labor.	1950's	SATIS.	
	X				Work Sampling of Indirect Labor. MAINTENANCE WORK (LIMITED USE)	1970's	SATIS	
	X				M.T.M. in Direct Labor.	1980's	SATIS	X
					Work-factor in Direct Labor.			
X					Other Predetermined Time Systems in Direct Labor.(Specify)			
X					M.T.M. in Indirect Labor.			
					Work-factor in Indirect Labor.			
X					Other Predetermined Time Systems in Indirect Labor.(Specify)			
	X				Standard Data Systems.(Specify BOEING DEVELOPED ELEMENTAL DATA)	EARLY 1950's	SATIS.	DATA RETRIEVAL.
X					Electronic Time Measurement Devices.(Specify)			
	X	X			Estimating and Costing using Work Measurement.	1970's	SATIS.	
X					Wage Incentives using Work Measurement.			
	X	X			Production Scheduling using Work Measurement. INDIVIDUAL EQUIP LOADING	1960's	SATIS.	
	X				Justifying Numerical Control Machines using Work Measurement.	1970's	SATIS	
X					Justifying Robotics using Work Measurement.			
	X				Equipment Utilization using Work Measurement.	1960's	UNSATIS. NEED MACHINE HOURS	
	X	X			Work Measurement for other usage.(Specify MANPOWER PLANNING)	1960's	SATIS.	
X					Video Tape/TV Monitor.			
X					Memomotion or Micromotion using Camera.			

What problems should Industrial Engineers be addressing in Work Measurement?

How Micro-electronics and Computer Devices may be helpful in your Work Measurement and Work Design?

-Please use the other side for the answers- Thank you.

Name TOM MOFFEIT Company WEYERHAEUSER Address 785 N. 42ND ST SPRINGFIELD OR

Title IN RIAL ENGINEER Number of Employees 12 0 Main Product WOOD PRODUCTS LUMBER PLYWOOD PARTICLE BOARD

N. of Industrial Engineers 2

We never have used or done it before.	We use or do it but manually.	We use or do it using computer programs developed inside the company.	We use or do it using computer programs purchased outside the company.	We would like to see it computerized.		When was it introduced?	Satisf. or Unsatis. with Results	It could be improved with Com. Programs
	X			X	Time Study of Direct Labor using Stop-watch.			
	X			X	Time Study of Machine Utilization using Stop-watch.			
	X				Work Sampling of Direct Labor.			
X					Work Sampling of Indirect Labor.			
	X				M.T.M. in Direct Labor.			
	X				Work-factor in Direct Labor.			
					Other Predetermined Time Systems in Direct Labor.(Specify)			
X					M.T.M. in Indirect Labor.			
	X				Work-factor in Indirect Labor.			
X					Other Predetermined Time Systems in Indirect Labor.(Specify)			
					Standard Data Systems.(Specify)			
	X	X			Electronic Time Measurement Devices.(Specify UPTIME METERS)			
	X			X	Estimating and Costing using Work Measurement.			
	X				Wage Incentives using Work Measurement.			
	X			X	Production Scheduling using Work Measurement.			
					Justifying Numerical Control Machines using Work Measurement.			
					Justifying Robotics using Work Measurement.			
	X				Equipment Utilization using Work Measurement.			
					Work Measurement for other usage.(Specify)			
	X				Video Tape/TV Monitor.			
	X				Memomotion or Micromotion using Camera.			

What problems should Industrial Engineers be addressing in Work Measurement?

How Micro-electronics and Computer Devices may be helpful in your Work Measurement and Work Design?

-Please use the other side for the answers: Thank you.

Name D.S. Rushing Company FMC Marine & Rail Address 4700 NW Front Ave

Title MG IND ENGRG Number of Employees 150 Main Product Rail Cars, Barges Ships

N. of Industrial Engineers 7

We never have used or done it before.	We use or do it but manually.	We use or do it using computer programs developed inside the company.	We use or do it using computer programs purchased outside the company.	We would like to see it computerized.		When was it introduced ?	Satisf. or Unsatis. with Results	It could be improved with Com. Programs
	X				Time Study of Direct labor using Stop-watch.			
	X				Time Study of Machine Utilization using Stop-watch.			
	X	X			Work Sampling of Direct labor.			
X					Work Sampling of Indirect labor.			
X					M.T.M. in Direct labor.			
X					Work-factor in Direct labor.			
X					Other Predetermined Time Systems in Direct labor.(Specify)			
X					M.T.M. in Indirect labor.			
X					Work-factor in Indirect labor.			
X					Other Predetermined Time Systems in Indirect labor.(Specify)			
X					Standard Data Systems.(Specify)			
X					Electronic Time Measurement Devices.(Specify)			
	X				Estimating and Costing using Work Measurement.			
X					Wage Incentives using Work Measurement.			
	X				Production Scheduling using Work Measurement.			
	X				Justifying Numerical Control Machines using Work Measurement.			
X					Justifying Robotics using Work Measurement.			
	X				Equipment Utilization using Work Measurement.			
					Work Measurement for other usage.(Specify)			
X					Video Tape/TV Monitor.			
X					Memomotion or Micromotion using Camera.			

What problems should Industrial Engineers be addressing in Work Measurement?

How Micro-electronics and Computer Devices may be helpful in your Work Measurement and Work Design?

-Please use the other side for the answers. Thank you.

Name LARRY WHITE Company OMARK INDUSTRIES, INC. Address 4909 INTERNATIONAL WAY MILWAUKIE, OR 97222

Title PR UCTION MANAGER Number of Employees 151 Main Product OREGON® SAW CHAIN

N. of Industrial Engineers 8

We never have used or done it before.	We use or do it but manually.	We use or do it using computer programs developed inside the company.	We use or do it using computer programs purchased outside the company.	We would like to see it computerized.		When was it introduced ?	Satisf. or Unsatis. with Results	It could be improved with Com Programs
✓					Time Study of Direct labor using Stop-watch.	—		
	✓				Time Study of Machine Utilization using Stop-watch.		SATISFIED	
	✓				Work Sampling of Direct Labor.		"	
✓					Work Sampling of Indirect labor.		"	
✓					M.T.M. in Direct labor.		"	
✓					Work-factor in Direct labor.		"	
✓					Other Predetermined Time Systems in Direct Labor.(Specify)		"	
✓					M.T.M. in Indirect labor.		"	
✓					Work-factor in Indirect labor.		"	
✓					Other Predetermined Time Systems in Indirect Iabor.(Specify)		"	
		✓	✓		Standard Data Systems.(Specify IBM, DEC, GE TIME SHARE, WANG)		"	
		✓			Electronic Time Measurement Devices.(Specify MINI - COMPUTERS)		"	
✓					Estimating and Costing using Work Measurement.		"	
✓					Wage Incentives using Work Measurement.		"	
✓					Production Scheduling using Work Measurement.		"	
✓					Justifying Numerical Control Machines using Work Measurement.		"	
✓					Justifying Robotics using Work Measurement.		"	
✓					Equipment Utilization using Work Measurement.		"	
✓					Work Measurement for other usage.(Specify)		"	
✓					Video Tape/TV Monitor.		"	
✓					Memomotion or Micromotion using Camera.		"	

What problems should Industrial Engineers be addressing in Work Measurement?

How Micro-electronics and Computer Devices may be helpful in your Work Measurement and Work Design?

-Please use the other side for the answers. Thank you.

Name JIM LUSHINA Company TEKTRONIX Address BEAVERTON OR.
Title Mgr Ind Stds Mgr. Number of Employees 22 00 Main Product
N. of Industrial Engineers 50 +

We never have used or done it before.	We use or do it but manually.	We use or do it using computer programs developed inside the company.	We use or do it using computer programs purchased outside the company.	We would like to see it computerized.		When was it introduced ?	Satisf. or Unsatis. with Results	It could be improved with Com Programs
		X			Time Study of Direct Labor using Stop-watch.	1974	OK	
		X			Time Study of Machine Utilization using Stop-watch.	"	"	
		X			Work Sampling of Direct Labor.	76	"	
		X			Work Sampling of Indirect Labor.	76	"	
		X	X		M.T.M. in Direct labor.	75	"	
X					Work-factor in Direct Labor.			
		X			Other Predetermined Time Systems in Direct Labor.(Specify Std. Data)	75	"	
		X			M.T.M. in Indirect labor.	75	"	
X					Work-factor in Indirect Labor.			
X					Other Predetermined Time Systems in Indirect Labor.(Specify)			
		X			Standard Data Systems.(Specify Physical Characteristic Technology)	76	"	
		X			Electronic Time Measurement Devices.(Specify RADAC)	77	"	
		X			Estimating and Costing using Work Measurement.	76	"	
X					Wage Incentives using Work Measurement.	"		
		X			Production Scheduling using Work Measurement.	"		
		X			Justifying Numerical Control Machines using Work Measurement.	"		
		X			Justifying Robotics using Work Measurement.	"		
		X			Equipment Utilization using Work Measurement.	"		
					Work Measurement for other usage.(Specify Capacity planning)	"		
		X	X		Video Tape/TV Monitor.	"		
X					Memomotion or Micromotion using Camera.			

What problems should Industrial Engineers be addressing in Work Measurement?
How Micro-electronics and Computer Devices may be helpful in your Work Measurement and Work Design?
-Please use the other side for the answers. Thank you.

Name Michael V Smith Company Freightliner

Title MN ER Corp I.E. Number of Employees ~~3~~ 500

Address 4747 N CHANNEL P.O Box 3849 Ptd, OR 97208

Main Product Diesl Truc's

N. of Industrial Engineers 25

We never have used or done it before.	We use or do it but manually.	We use or do it using computer programs developed inside the company.	We use or do it using computer programs purchased outside the company.	We would like to see it computerized.		When was it introduced ?	Satisf. or Unsatis. with Results	It could be improved with Com. Programs
	X		X		Time Study of Direct labor using Stop-watch. Computer 1979	Manual 1966	S	N/A
			X		Time Study of Machine Utilization using Stop-watch.	1979	S	
	X				Work Sampling of Direct labor.	1963	S	Yes
	X				Work Sampling of Indirect labor.	19871	S	Yes
X					M.T.M. in Direct labor.			
X					Work-factor in Direct labor.			
X					Other Predetermined Time Systems in Direct labor. (Specify)			
X					M.T.M. in Indirect labor.			
X					Work-factor in Indirect labor.			
X					Other Predetermined Time Systems in Indirect labor. (Specify)			
					Standard Data Systems. (Specify Used before but discontinued)		U	Yes
X					Electronic Time Measurement Devices. (Specify)			
	X	X	X		Estimating and Costing using Work Measurement. 1960, 1977, 1978	—	S	
X					Wage Incentives using Work Measurement.			
	X	X			Production Scheduling using Work Measurement. 1975, 1978	→	U, S	
	X	X	X		Justifying Numerical Control Machines using Work Measurement.	1973, 1979	S	
		X	X		Justifying Robotics using Work Measurement.	1979	S	
	X	X	X		Equipment Utilization using Work Measurement.	1966, 1980	S	
X					Work Measurement for other usage. (Specify)			
	X		X		Video Tape/~~TV Monitor~~.	1976	U	
X					Memomotion or Micromotion using Camera.			

What problems should Industrial Engineers be addressing in Work Measurement?

How Micro-electronics and Computer Devices may be helpful in your Work Measurement and Work Design?

-Please use the other side for the answers- Thank you.