A computer system for bedside medical research*

by STEVEN E. WIXSON, EUGENE M. STRAND and H. WILLIAM PERLIS

The University of Alabama Medical Center
Birmingham, Alabama

INTRODUCTION

The University of Alabama Myocardial Infarction Research Unit (MIRU) supports clinical research on patients who have sustained a myocardial infarction (heart attack). MIRU is a contract from the National Heart Institute whose goal is the reduction of mortality and morbidity from myocardial infarction. Patient rooms provide the environment for intensive coronary care and research. Two laboratories facilitate study of critically ill patients with complicating conditions, such as, shock, congestive heart failure and severe arrhythmia. The digital computer housed adjacent to the patient rooms is dedicated to on-line real-time MIRU research (see Figure 1., MIRU).

The Shock Research Unit of Los Angeles County Hospital pioneered the application of digital computers for on-line clinical research of cardiovascular functions.[1] Following that effort, electronic data processing techniques are being used increasingly for patient monitoring.[2-5] These applications emphasize clinical care of postoperative patients, and they primarily monitor only a few variables, such as, blood pressure, heart rate, respiratory rate, temperature, and urine flow. The patient monitoring programs developed by Sheppard, et al., at the University of Alabama[5] are used in the clinical care of patients in the MIRU.

The variety of MIRU research protocols (e.g., thermal dilution cardiac output, assisted circulation, ECG rhythm analysis) demand a changeable support system both in the bedside instrumentation and in the computer software. The MIRU research system emphasizes flexibility in facility allocation and ease of programming and operation.

* Supported in part by U.S. Public Health Service Contract No. PH43-67-1441

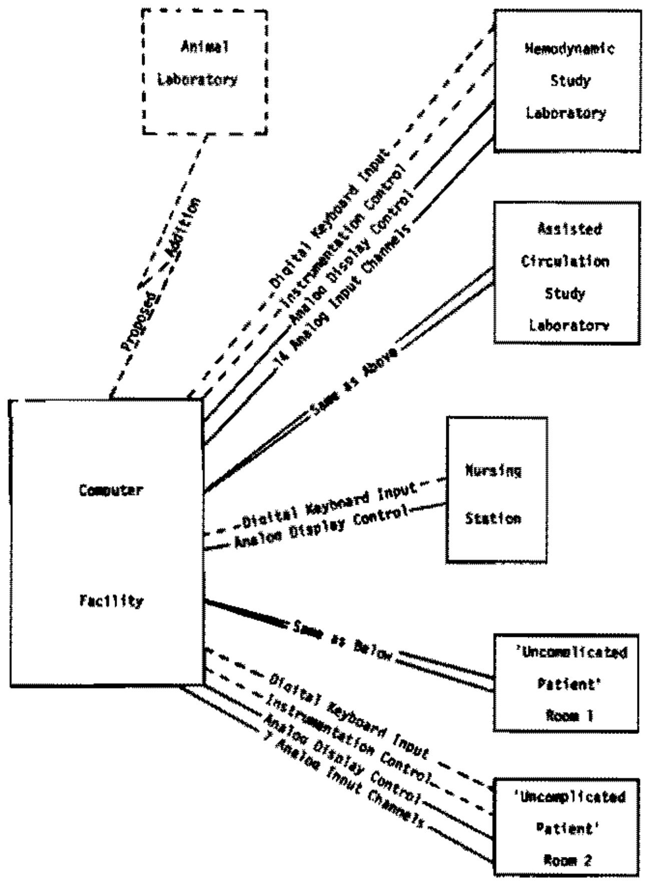


Figure 1—Myocardial infarction research unit

Computer Requirements

MIRU research requires the following computer capabilities:

- Acquisition and analysis of data from multiple beds,

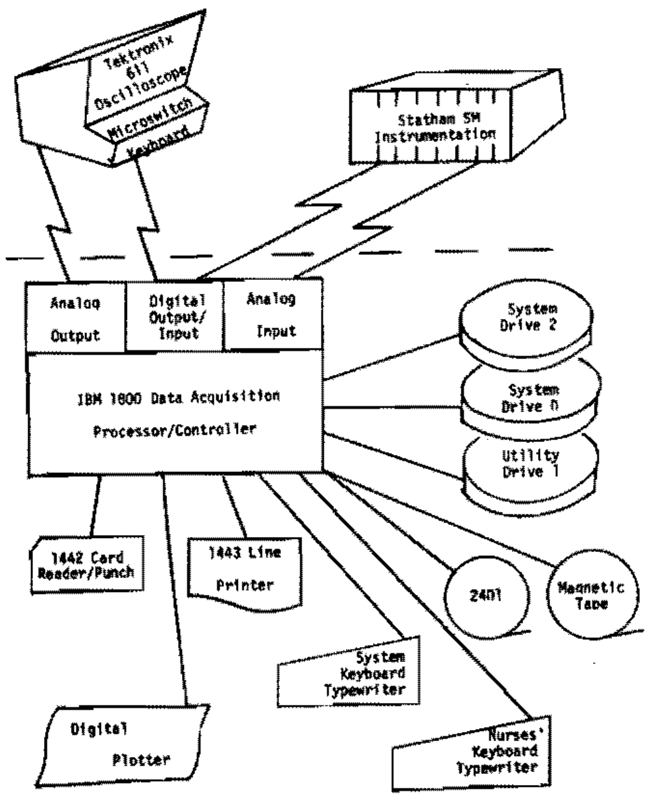


Figure 2—System design

- Control of measurement and therapy devices,
- Real-time display of information, and
- Concurrent development of computer programs.

The patient data for research studies include physiological, historical, physical examination, laboratory, clinical observation, intervention, and pathological data. Intervention data include medication, therapy, changes in position, and research protocols. Physiological data acquired in analog form include sampled data and derived parameters.

Computer configuration

The IBM 1800 Data Acquisition and Control System[6] with the Multiprogramming Executive (MPX)[7] provides the foundation for the system which meets these requirements. Supplementing MPX, a MIRU executive supports data entry at multiple terminals and allocates system facilities to multiple patients. This executive requires minimal alteration to MPX.

The IBM 1800 processor controller is a 32K word, 2-microsecond cycle time computer. Each word is 16 bits plus 2 bits for parity checking and storage protection. Digital input, digital output, and analog output features allow connection of special devices, e.g., remote terminals (see Figure 2., System Design).

Three direct access disk drives provide 1.5 million words total of on-line storage for programs and data (see Table I, Disk Allocation). Two drives are reserved for the real-time parts of the MIRU system. The third is a utility drive serving as one of the following:

- backup for the system drives,
- storage for source and object programs during development,
- storage of the MIRU master patient index for statistical studies, and
- large volume disk storage for a particular research protocol.

Two 60kb tape drives provide high volume storage for research studies. One, the real-time tape drive, logs data from patient files for retrospective study. The other, the special study tape drive, provides temporary data storage in scheduled research procedures and backup for the real-time tape drive.

Two 20kc analog-to-digital converters provide continuous and noncontinuous modes for analog data acquisition. One, the real-time converter, converts data

TABLE I—Disk Storage Allocation

SECTORS	*DRIVE 0*	*DESCRIPTION*
300		IBM system programs.
800		TASK working storage.
293		Coreload storage.
60		Temporary patient file.
2		Terminal control file.
54		Interrupt save area.
27		Batch save area.
56		Executive Director.
8		Cold start program.
SECTORS	*DRIVE 1*	*DESCRIPTION*
24		Disk index table.
350		IBM relocatable programs.
100		General relocatable subroutines.
286		Batch work storage.
40		Test process work storage.
50		Test coreload area.
750		Source program file.
SECTORS	*DRIVE 2*	*DESCRIPTION*
8		Disk index table
1000		Patient active file.
592		Coreload area.

which is processed continuously by core-resident routines. The second, the special study converter, is used for converting bursts of data for routine clinical processing and research experiments. MPX schedules the second converter by queueing conversion requests. The queued requests are serviced according to a priority assigned to the research experiment. Experiments demanding immediate response receive a high priority. The special study converter serves as back-up for the real-time converter.

Multiprogramming executive

The IBM 1800 Multiprogramming Executive operating system is the real-time monitor for the computer. MPX provides automatic handling of interrupts from data input-output (I/O) devices and user sources, automatic program scheduling, on-line hardware diagnostics, and time sharing for real-time routines, process programs, and background processing. The MIRU system provides these areas for program execution: Special coreload area (SPAR), coreload area, and variable core. These areas service, respectively, programs of high response and short execution (1 millisecond), medium response and medium execution (1 second), and slow response with variable execution (see Table II, Core Allocation).

The MIRU executive features: (1) Remote terminal control of system facilities, (2) Computer controlled medical instrumentation, (3) Task concept for allocation of system facilities, (4) FORTRAN programming environment with multiple entry points, (5) Flexible program communication including program control of exception conditions, and (6) Standardized handling for data storage in the patient file.

TABLE II—Core Storage Allocation

PARTITION OR AREA	*SIZE*	*DESCRIPTION*
Inskel Common	3 K*	Core resident storage area for program communications. Terminal control block, bed and parameter control blocks, task block and working storage.
Executive I/O and Director	10 K*	Core resident parts of the IBM MPX System. Interrupt Handler, Program Scheduler, Disk I/O Routine, Error Routine.
MIRU Executive and FORTRAN Subroutines	3 K*	Keyboard entry routine, instrumentation handler, MIRU housekeeping routine, task timer control. Frequently used FORTRAN subroutines.
SPAR	4 K*	Special coreload area for fast response (Millisec). Continuous signal processing.
Core Load Area 1	4.5 K	Medium response (1–5 sec). Short execution times (Up to 1 sec). Disk loaded programs to handle remote terminals.
Core Load Area 2**	4 K	Slow response (10–20 sec). Longer executing time (5 sec). Work horse area used by most MIRU processing programs.
Variable Core	7.5 K	1st priority—high response processing on a core exchange basis for programs too large for Areas 1, 2. 2nd priority—long executing, large programs with no response required. 3rd priority—batch processing.

*These areas are storage protected.

** This area is planned for the future when additional core storage can be obtained. These programs are now executed as 1st priority in variable core.

REMOTE TERMINALS

Remote terminals control the work load of the disk oriented computer system. The remote terminal consists of a storage oscilloscope and a keyboard for display and entry of information. The storage oscilloscope produces excellent graphic and alphanumeric displays for review of information. High quality graphic plots are essential in the MIRU environment where large quantities of analog data are processed. The storage oscilloscope also offers visual quality control of signals sampled by the computer.

The keyboard keys and lights are connected to the computer's digital input and output points. Data entries (numerics, minus sign, blank, decimal point) are displayed on the top line of the scope for visual verification as the key is depressed. Action keys (clear, enter, respond, reset) cause the computer to perform the specified function. The lights (attached, busy, message) show the terminal's status.

The keyboard and oscilloscope terminals facilitate communication between programs and researchers. Through audio-tone and lights on the terminal, alarm conditions, alert conditions, and routine messages can

TABLE III—Terminal Control Block (TCB)

WORDS	*P**	*DESCRIPTION*
1	P	Terminal number.
2	P	Address of task block in control of terminal.
3	P	Bed control block number.
4	P	Hardware bit mask for lights and display.
5–8		Display "mode" scale factors.
9–10		Display origin for keyboard entry.
11–13		Display origin for FORTRAN IOCR.
14–21		Display scale factors.
22–31	P	Special function program names.
32–47		Keyboard buffer and pointer.
48		Time of data entry.
49	P	Digital input address.
50		Reserved.

* NOTE: "P" means storage protected, and a blank space means not protected.

be signalled. The user can enter data at the keyboard for a program, and the program can display textual, numeric, and graphic information.

A fixed in-core table of parameters exists for each terminal. This terminal control block (TCB) contains information pertinent to the generating of displays and keying of data. MIRU executive plot routines reference the TCB for scaling factors. The keyboard entry routine buffers characters in the TCB (see Table III, Terminal Control Block).

The patient's bedside or nurses' station terminal can be used to call programs into execution and to enter data during program execution. A terminal is normally in program call mode (unattached). A program request through a MIRU executive subroutine dedicated (attaches) a terminal for data entry.

A three-digit name identifies each process program. The first digit is the hardware priority[6] level at which the program will execute. Digits 2 and 3 provide identification for programs which execute on that level. Entry through the keyboard of a program name queues the program for execution. Function buttons on the keyboard map through the TCB into the ten programs most frequently called from a particular terminal. Selecting a function button queues a specific program for execution. Thus a program can be selected either by keying the three digit name or by selecting a function button.

When a keyboard has been attached to a program, data can be entered for that program. Up to 15 characters of information (numerics, minus sign, blank, decimal point) can be buffered in the TCB. Individual data words are separated by the blank, so more than a single value can be entered. For example, the string [5 12.7 1 −53 15] represents five distinct data entries. MIRU executive subroutines move the character string from the TCB and convert it to FORTRAN real or integer values.

BEDSIDE INSTRUMENTATION

Commercially available medical instrumentation provides 7 to 14 channels of analog signals from each patient room. The instrumentation has been modified to permit computer identification of transducers and modules and to allow computer control of bedside devices. The modular computer-controlled instrumentation meets the research requirements of MIRU, as the bedside instrumentation requirements vary with different protocols being conducted. In patient rooms, computer-connected cabinets accept up to 7 channels of instrumentation. In the two laboratories, up to 14 computer-connected module positions are available (see Figure 1, MIRU). Transducer panels above the patient's bed provide one computer-coded transducer connector for each module position in the cabinet. Above each connector is the "transducer active" button used to signal the computer to put a transducer on-line.

When a analog signal is needed, the nurse connects the transducer, plugs in an amplifier module, and presses the transducer active button. The computer reads digital information from the amplifier module and transducer connector. A computer program checks the 5-bit code to insure proper setup of the instrumentation. Using the remote terminal, the program guides the calibration of the amplifier. A light in the "transducer active" button signifies the signal is on-line.

A second pressing of the button sets the signal off-line removes the calibration tables, and turns off the transducer active light.

In the computer, tables store module addressing and calibration information for all programs using a signal. A bed control block (BCB) for each patient in the research unit contains identification and physiological information used frequently by programs. Programs access the data in the BCB through MIRU executive routines which load or save values. A program can only address the BCB of the bed for which it is active.

Since instrumentation requirements vary, a block of storage called the parameter control block (PCB) is dynamically allocated when a parameter (e.g., blood pressure, surface ECG) is placed on-line. A fixed section of the PCB contains addressing and calibration information. A variable section holds the derived data related to the parameter. The derived values are usable for all programs and are referenced through system load/store routines (see Table IV, BCB's and PCB's).

TASK CONCEPT

The basic unit of work in the MIRU system is the 'task.' A task is defined as a disk loadable program that is active for a specific patient and is uniquely identified by a program name and a bed number. It may be of short duration (a few hundred milliseconds), such as, a summary display; or it may be of long duration (the entire patient stay), such as, a monitoring program executed at periodic intervals. Tasks can be initiated from the remote terminals or through other tasks.

Since a new task is started for each patient, one program is serially reusable for all patients on the unit. This conserves disk storage; since only one copy of a program need be stored on disk.

The task concept dynamically provides system facilities to each patient in the research unit. The bedside terminal will be dedicated to any task for entry or display of information. Core and disk working storage are dynamically allocated for intermediate storage of parameters. Lights and audio alarms are available for signalling alarm and message conditions. Real-time measurements from parameters are available to all programs, and any program can log information to the patient file.

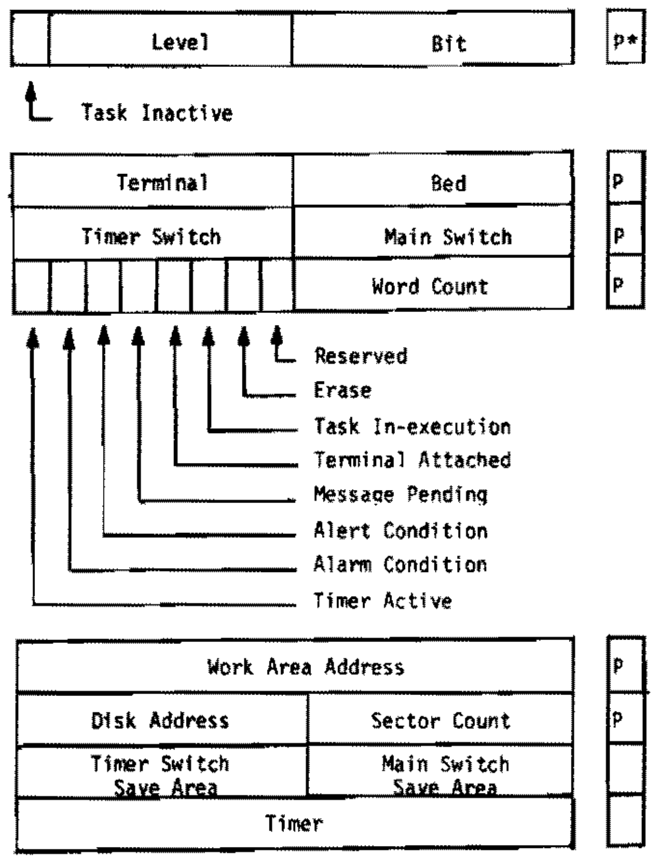
A programmable timer is provided for each task. The 1-sec. time base timer can be used to recall a program at a set interval, check data entry or response to alarm conditions, or schedule a program execution on a periodic basis.

At task request time, the MIRU executive allocates eight words of core called a task block and initializes task parameters. The parameters include the program name, the terminal number, the bed number, two program switches, eight program status flags, the word count and address of in-core storage, the sector address and count of disk storage, a program switch save area, and a task timer (see Figure 3, Task Block).

TABLE IV—Bed Control Block (BCB) and Parameter Control Block (PCB)

WORDS	*P**	*BED CONTROL BLOCK DESCRIPTION*
0	P	BCB length.
1	P	Bed control block number.
2–5	P	Patient number.
6–12	P	Patient name.
13	P	Digital input/output addresses.
14–40		Physiological data storage.
41	P	Maximum number of PCB's.
42–**		Parameter block addresses.
WORDS	*P**	*PARAMETER CONTROL BLOCK DESCRIPTION*
0	P	PCB length.
1	P	Multiplexor address of module.
2	P	Address of task block in control of module.
3	P	Slope calibration.
4	P	Zero calibration.
5+		Derived physiological parameters.

* NOTE: "P" means storage protected, and a blank space means not protected.
** NOTE: = 73 for laboratories.
= 47 for uncomplicated patient rooms.



*NOTE: "P" means storage protected, and blank space means not protected.

Figure 3—Task block

Figure 4—MIRU Sample Task Program

```
SAMPLE TASK PROGRAM
// FOR STASK
C*************************************************************
C*    NAME          STASK                                   *
C*    TITLE         SAMPLE MIRU TASK PROGRAM                *
C*    DATE          4/1/70                                  *
C*    FUNCTION      THIS ILLUSTRATES HOW PRO-               *
C*                    GRAMS ARE STRUCTURED IN               *
C*                    THE UNIVERSITY OF ALABAMA             *
C*                    MIRU SYSTEM. THE WRITEUP              *
C*                    IN THE MIRU PROGRAMMERS               *
C*                    GUIDE EXPLAINS THIS PRO-              *
C*                    GRAMMING IN DETAIL.                   *
C*    ENTRY         CALL TASK (NAME, IERR) FROM             *
C*                    ANOTHER PROGRAM OR CALL               *
C*                    FROM REMOTE TERMINAL                  *
C*                    (PROGRAM SWITCH SET TO 1)             *
C*                    CALL TASKP (NAME, WDCNT,              *
C*                    NOPAR, ARRAY, IERR) FROM              *
C*                    ANOTHER PROGRAM (PRO-                 *
C*                    GRAM SWITCH SET TO 2) CALL            *
C*                    TENTR (NAME, ICODE, IERR)             *
C*                    BY ANOTHER PROGRAM (PRO-              *
C*                    GRAM SWITCH SET TO 3)                 *
C*                  ERROR RECALL (PROGRAM                   *
C*                    SWITCH SET TO 4)                      *
C*    EXIT          CALL MEXIT (TYPE, NEXT                  *
C*                    SWITCH, TIMER SWITCH,                 *
C*                    TIMER INTERVAL)                       *
C*************************************************************
      DATA NAME /Z***/
C*****PROGRAM NAME IS A THREE DIGIT NUMBER.
      CALL INITL (NAME, IPRSW, IALTS, IBED, ITERM,
      ISAVM, ISAVT)
      GO TO (100, 200, 300, 400, 500) IPRSW
C
C*****PROGRAM INITIALIZATION WITHOUT
        PARAMETERS
  100 CALL WORK (10, IERR)
C*****INSURE WORK AREA IS ALLOCATED (IERR IS
        '-' OR '+')
      IF (IERR) 110, 1000, 110
  110 CONTINUE
  CALL MEXIT (ITYPE, IPRSW, IALTS, INTVL)
C
C*****PROGRAM INITIALIZATION WITH PARAMETERS
  200 CONTINUE
C*****WORK STORAGE HAS BEEN ASSIGNED BY
        CALLER
C*****USE 'LDTSK' TO RETRIEVE INITIAL
        PARAMETERS
      CALL MEXIT (ITYPE, IPRSW, IALTS, INTVL)
C
C*****EXTERNAL ENTRY SECTION
  300 CONTINUE
C*****'IALTS' CONTAINS ONE WORD FROM THE
        EXTERNAL CALLER
      GO TO 1000
C
C*****PROGRAM ERROR RECALL SECTION
  400 CONTINUE
C*****'IALTS' CONTAINS THE ERROR CODE
      GO TO 1000
C*************************************************************
  500 CONTINUE
C
C*****PROCESSING COMPLETE EXIT
 1000 CALL MEXIT (0, 0, 0, 0)
      CALL EXIT
      END
```

PROGRAM ORGANIZATION

To enhance response times, core utilization, and disk storage capacity, the MIRU programming system provides multiple entry points to a FORTRAN program. Thus, without being rolled-out (saved) onto disk, a program can be exited and entered at a later time. In-core storage is dynamically available to each program for the preservation of intermediate parameters. While a program is not active (executing), its coreload area is used by other programs. On recall for execution a fresh copy of the program is read from disk and the execution continues from the designated entry point.

This programming system conserves the overhead time that would otherwise be required to save an interrupted coreload on disk and frees a coreload area while a program is inactive. While one task waits for the entry of data from a keyboard, another task program is executing.

Task program

A task program is many program segments connected together under the control of two program switches, the main and timer switches. The switch mechanism is the FORTRAN computed-go-to statement.[8] A program segment begins at one of the statement numbers in the computed-go-to statement and ends with a call to the MIRU executive exit routine. This gives a FORTRAN program multiple entry points.

In the computed-go-to statement, GO TO (100, 200, 300, 400, 500), IPRSW, the numbers are the entry points to the program and "IPRSW" is the main program switch (see Figure 4, MIRU Sample Task Program). By selectively setting "IPRSW" as 1, 2, . . ., N, the program can begin execution at entry points 100, 200, . . . respectively. This provides a powerful but easy mechanism to select the same or different entry point each time the program is executed.

The switches provide the programmer with two independent control paths for each task. The first or main switch is the entry point following data input via a remote terminal or response to an alarm, alert, or message signal.

The second or timer switch is the entry point following a time-out of the task timer. Since the two switches are independent, each may point to a different segment of the program. A program requesting data can regain control after a specified time interval, whether or not data has been entered.

Entry point selection

The MIRU executive initializes the switches at task setup time. At the end of each program segment, the MIRU exit routine sets the switches for the next entry. A task program determines its entry point through a call to the task initialization routine.

A task program begins with the task initialization routine. This routine identifies the task being processed and places the task block address on the MPX level work area, that is, the MPX mechanism for reentrant coding.[7] Subsequent MIRU executive routines reference the task block through the saved address. The initialization routine sets the task active flag and returns the program switches.

The last executable statement of every program segment is a 'CALL MEXIT' (i.e., MIRU executive exit routine). Through this routine, a program communicates with the task block. The user can specify why he is exiting the program (exit type), where he wishes to return (main switch), where to return on a timer recall (timer switch), and the increment of time to remain inactive. The MIRU executive saves this information in the task block, designates the task as inactive, and removes the task block address from the MPX level work area.

The following are task exit types:

0—complete task processing,
1—signal alarm condition,
2—signal alert condition,
3—signal message pending,
4—remain inactive for specified time interval,
5—request data from terminal without erasing display, and
6—request data from terminal and erase display.

Certain main entry points are reserved and defined as follows:

1—task initialized without parameters,
2—task initialized with parameters,
3—task reentered from external source, and
4—task recalled through error condition.

Initial entry without parameters

A task initialized through the terminal has no in-core or disk work storage; and hence, it can have no initial parameters. Working storage can be attained through a request to the MIRU executive. The task program can request up to 255 words of in-core storage and up to 128 sectors of disk storage. This storage space is found by the MIRU executive, which returns its address to the task block. The executive also returns status indicator: Work storage already exists; no work area remains; request successfully filled.

Initial entry with parameters

A task initialized from another task program can be in one of the following states:

- Neither disk nor in-core storage,
- Disk but not in-core storage,
- In-core but not disk storage, or
- Both in-core and disk storage.

A task program starting a new task can transfer data through in-core working storage. The MIRU executive selects a task block, initializes the program switches to entry point 'two', allocates task working storage, moves the data to the storage, and queues the new task for execution.

Using another MIRU executive routine, a program can transfer both in-core and disk working storage to the new task. This routine uses the existing task block for the new task, thus completing the processing by the first task. The parameters contained in in-core and disk working storage are passed to the second task. The program switches are set to entry point 'two'; the new task is queued for execution; and the existing program is exited.

Task working storage cannot be referenced directly. It is accessed indirectly through the work area address stored in the task block. On request, MIRU executive routines load or store values in the working storage. The load and store functions validate the relative

TABLE V—MIRU Abort Messages and Restart Codes

MIRU TASK ABORT MESSAGES

MESSAGE	*MEANING*
'MULT ER ABT'	Multiple error abort.
'WHOOPS BATCH CALL'	System routine called invalidly from BATCH
'WHOOPS TSK AD INV'	No 'CALL INITL' has been made prior to a call to a system routine.
'WHOOPS INV LVL/BT'	No active task found for name input to 'INITL.'
'DAMN DAMN DAMN'	Programmed status dump.
Word	*Definition*
1	Level/Area
2	I (User variable)
3	Terminal/Bed
4	Alternate switch/Main switch
5	Flag/Word count
6	Audit trail switches (Previous entry point)

MIRU ERROR RESTART CODES

4–1	Multiple error recall.
4–2	MPX program restart (FORTRAN error detected).
4–3	Task load/store relative word error.
4–4	Invalid switch in 'MEXIT' call.
4–5	Invalid type in 'MEXIT' call.
4–6	No interval in type 4 'MEXIT' call.
4–7	Invalid word count in 'WORK' call.
4–8	Attempted use of terminal attached to another 'TASK' (e.g., 'CALL TEXT').
4–9	Invalid terminal call in 'MEXIT.'
4–10	Terminal number error in 'ATACH' call.
4–11	Error in code in 'TENTR' call.
4–12	Error in word count in 'TASKP' call.
4–13	Sector count error in 'DWORK' or 'MADDR' call.
4–14	Invalid name in 'TASK,' 'TASKP,' 'TASKT' call.
4–15	Sector count outside reserved area in 'MADDR' call.
4–16	Display number is zero, negative, or greater than number of displays on system.

MIRU EXTERNAL ENTRY CODES

3–1	System reload (Storage protect, op code, or parity error).
3–2	Forced terminal separation (Keyboard reset).

address of the word affected. A task restart occurs if a program attempts to address an invalid core storage location.

External entry

A task reentered through a system reload, through the keyboard reset, or from another task receives a communication code in the alternate (timer) switch. This external entry disrupts the normal flow of the task execution. The MIRU executive saves the former switches; and the task initialization routines returns these to the program. The user can reestablish the flow of execution.

Restart entry

A task restarted through an exception or error condition receives a code in the alternate switch. The MIRU executive leaves control with the program whenever possible. Only the program knows the significance of an exception condition and what corrective action is necessary. On restart, the program can shut down a process or inform the user of the error condition. Only as a program loops in multiple restarts is it forceably aborted. Abort conditions are logged on the system printer (see Table V, MIRU Abort Messages and Restart Codes).

DATA MANAGEMENT

A disk file for each patient's data contains all information needed in real-time. A variety of data types can be saved in the patient file: Raw analog-to-digital values, derived parameters, coded information and narrative data. Information saved in the patient file is used to generate summary displays and for retrospective studies.

MIRU executive routines log each piece of information to a temporary disk file, sort data for individual patient files, and dump disk files to magnetic tape.

When entered into the logging files, information is identified with patient's number, a data type code, and time stamp. A retrieval program collects information from the patient file by code and patient number. Using the code, retrieval routines can extract the requested data without reference to an external source for attributes, e.g., word count. The code also allows logging a group of data, each element identified only by its relative position in the group and the code of the record. This fixed position format is used to log the multiple results of a program without coding each piece of data within the record.

A log request moves the data to a core buffer area. After sixteen requests the core buffer is transferred to the temporary log file on disk. Periodically, a program sorts the temporary disk file (which has data mixed from all patients) for the real-time tape, and individual patient disk file (the patient active file), or the nurse's printer. The tape records are blocked five disk records to one tape record (see Figure 5, Information Logging). When a patient is dismissed from the research unit, a program extracts the data for that patient and gener-

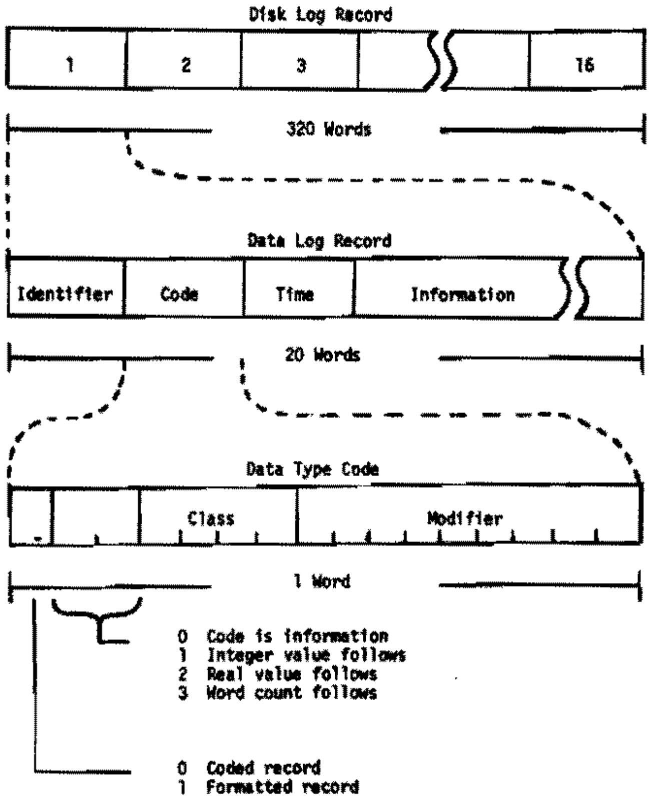


Figure 5—Information logging

ates an individual tape file. This forms the permanent record for retrospective studies.

Analog-to-digital values and large arrays of derived data may be written directly to the real-time tape. In logging data directly to the real-time tape, the user must provide a tape header including patient identification, data type, and time of acquisition. Otherwise, the record will be discarded when the real-time tape is sorted into individual patient files.

CONCLUSION

The research system has been operational since July, 1969, with a single prototype of the medical instrumentation and two keyboard/oscilloscope terminals. The Myocardial Infarction Research Unit opened in October, 1969, so the computer and instrumentation system is being used for research studies from two patient rooms and two laboratories. Instrumentation of an animal laboratory is projected for 1970.

We have not had sufficient experience in a multiple bed research environment to note the strengths and weaknesses of the system. These will be reported at the conference.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the efforts of Mr. Jeary Vogt, now of Shared Medical Systems Corporation, Philadelphia, Pa., in the selection of monitoring equipment and the design of the digital computer interface for the bedside instrumentation. The patient cooperation of Mr. Louis Sheppard of the Department of Surgery in providing consultation, program test time, and proven monitoring programs is appreciated.

REFERENCES

1 H SHUBIN M H WEIL
Efficient monitoring with a digital computer of cardiovascular function in seriously ill patients
Annals of Internal Medicine 65:3 1966

2 H R WARNER R M GARDNER A F TORONTO
Computer-based monitoring of cardiovascular functions in post-operative patients
Supplement II to Circulation 37 1968

3 J J OSBORN J O BEAUMONT S C A RAISON J RUSSELL F GERBODE
Measurement and monitoring of acutely ill patients by digital computer
Surgery 16:6 1968

4 D W CHAAPEL G RASTELLI R WALLACE
On-line computer care of post-operative cardiac patients
Digest of IEEE Computer Group Conference June 1969

5 L C SHEPPARD N T KOUCHOUKOS M A KURTTS J W KIRKLIN
Automatic treatment of critically ill patients following operation
Annals of Surgery 168:4 1968

6 *IBM 1800 data acquisition and control system: Functional characteristics*
IBM Systems Reference Library Form A26-5918

7 *IBM 1800 multiprogramming executive operating system: Programmers guide*
IBM Systems Reference Library Form C26-3720

8 *IBM 1130/1800 basic FORTRAN IV language*
IBM Systems Reference Library Form C26-3715

Patient Room with computer terminal and bedside instrumentation

Special procedures room with computer terminal and rack instrumentation

Remote terminal display using storage oscilloscope (left) and beside monitor via scan converter storage oscilloscope (right)