

DESCRIPTION OF SOFTWARE  
FOR THE EBR-II DIGITAL DATA ACQUISITION SYSTEM

by

R. W. Hyndman, J. M. Allen,\* R. A. Call, E. W. Laird,  
M. R. Tuck, and K. D. Tucker

EBR-II Project  
Argonne National Laboratory  
Argonne, Illinois - Idaho Falls, Idaho

**LEGAL NOTICE**

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

December 1970

\* U. S. Atomic Energy Commission, Division of Reactor Development and Technology.

Work performed under the auspices of the U. S. Atomic Energy Commission.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT . . . . .	4
I. GENERAL DESCRIPTION . . . . .	4
A. Function of Software . . . . .	4
B. Realtime Environment . . . . .	5
C. Interrupts . . . . .	5
D. Foreground and Background Areas of Memory . . . . .	5
E. Protection of Software . . . . .	6
F. Special Features . . . . .	6
1. Floating-point Arithmetic . . . . .	6
2. Separation in Input/Output Processor . . . . .	7
3. Storage Capability . . . . .	7
G. Software Format . . . . .	7
II. SYSTEM CONFIGURATION . . . . .	8
A. Realtime Batch Monitor . . . . .	8
1. Job Control Processor (JCP) . . . . .	8
2. RAD Editor (RAEDIT) . . . . .	8
3. Overlay Loader (OVLOAD) . . . . .	9
4. Compilers or Processors . . . . .	9
B. RAD Area Allocation . . . . .	9
III. SOFTWARE ALLOCATION . . . . .	11
A. Software Residing in Foreground . . . . .	11
1. Time-actuated Software . . . . .	11
2. Internal-event-actuated Software . . . . .	13
3. External-event-actuated Software . . . . .	14
B. Background Support Programs . . . . .	14
C. Assignment of Resident and Nonresident Foreground . . . . .	15
IV. SOFTWARE MAINTENANCE . . . . .	16
APPENDIXES	
A. DAS Programs Generated and Proposed to Date by EBR-II Project . . . . .	17
B. List of Applicable Reference Documents on Software for the Sigma 5 Computer . . . . .	23
REFERENCE . . . . .	24

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	RAD Area Allocation and Organization . . . . .	10
2.	Time-actuated Software . . . . .	12
3.	Internal-event-actuated Software . . . . .	13
4.	Examples of External-event-actuated Software . . . . .	14
5.	Background Run Sequence . . . . .	15

DESCRIPTION OF SOFTWARE  
FOR THE EBR-II DIGITAL DATA ACQUISITION SYSTEM

by

R. W. Hyndman, J. M. Allen, R. A. Call, E. W. Laird,  
M. R. Tuck, and K. D. Tucker

ABSTRACT

This report describes system configuration, allocation, protection, and maintenance of software associated with the EBR-II digital data acquisition system. Software prepared both by Xerox Data Systems (the manufacturer of the Sigma 5 computer of the system) and by the EBR-II Project is considered.

I. GENERAL DESCRIPTION

A. Function of Software

The software associated with the EBR-II digital data acquisition system (DAS\*) provides the necessary interface and communication link between the DAS and the user. It is through the software that the computer and its peripheral equipment are controlled and information is gathered from the plant, processed, and conveyed to the reactor operator or programmer/analyst. Software is supplied both by Xerox Data Systems (XDS) and the EBR-II Project. The software supplied by XDS is primarily concerned with the system operation, such as utilization and management of storage and input/output (I/O) processing, whereas software generated by the EBR-II Project is user-oriented, such as programs for sampling plant parameters. The software generated and proposed to date by the EBR-II Project is listed in Appendix A of this report.

---

\*The DAS is a Sigma 5 computer, manufactured by Xerox Data Systems, which has been specifically configured to perform data acquisition and detailed on-line data reduction. A functional description of the system, which can be considered a "companion" of this report, has been published.<sup>1</sup>

B. Realtime Environment

The primary feature of the DAS is its realtime capability, which can be defined as the ability to respond immediately (within microseconds) to internal events such as clocking pulses or to external events such as interrupts. The response to these events must be fast enough to keep up with the realtime process that the computer is following.

C. Interrupts

Interrupts--of which there are both external (provided by equipment other than the Sigma 5) and internal (provided by the Sigma 5)--cause the normal program sequence to be modified. For example, a scram will trigger an interrupt that will halt any program in execution and start a program to determine the cause of the scram and the sequence of events leading up to it. After the scram program is executed, the system will resume execution of the original program at the point it was interrupted. Each interrupt is assigned a priority. Those with a lower priority cannot be triggered until the program triggered by an interrupt with a higher priority has been completed. The higher-priority interrupts are assigned to the more critical programs associated with plant operation and safety.

D. Foreground and Background Areas of Memory

The realtime batch monitor (RBM), discussed in Sect. II.A, divides the magnetic-core memory (CORE) and the fast bulk storage (RAD) into areas referred to as "foreground" and "background." All realtime programs are assigned to the foreground area for both storage and execution. This area can be further subdivided into "resident foreground" and "nonresident foreground." Resident foreground is stored in the core permanently, and nonresident foreground is stored on the RAD and transferred to the core only when needed. Programs assigned in this manner are discussed in Sect. III.

Background area is allotted for programs not associated with the realtime environment. Programs run in this area are referred to as "batch" or background programs. Batch programs are run only when foreground programs are not in operation and are executed serially. Examples of batch programs

are analytic programs such as rod-calibration and core-loading programs. Allocation of a background area provides an efficient method for using idle computer time when the foreground is not in operation. Background area can be used by the foreground by a process known as "checkpointing." The monitor automatically stores the background area needed by the foreground. When the foreground program has been completed, the background area is restored to its original state.

#### E. Protection of Software

Several different methods are used to prevent inadvertent modification of the software.

RAD write protection is provided both by manually actuated toggle switches and by key-in statements. Sixteen toggle switches are provided on the RAD, each one capable of inhibiting 32 tracks on the bulk storage area. Background programs are allowed to write only in the background data areas or background temporary areas of the bulk storage. (See Sect. II.B for organization of bulk storage.) For writing into the foreground area, the system operator must enter via the teletype a command to override the write protection.

Memory is protected by a write-lock feature. To write in a specific area of memory protected by a write-lock, the programmer must furnish a write-key that corresponds to the write-lock.

To write on a tape, a write-ring must first be put on the tape reel. This plastic ring fits in a groove on the back of the reel. When the ring is inserted, the tape depresses a microswitch to disable the write protection for the tape transport.

#### F. Special Features

##### 1. Floating-point Arithmetic

The system is implemented with floating-point hardware. Both short and long formats can be used. The short format consists of a 32-bit word (24-bit mantissa), and the long format consists of a 64-bit double word

(56-bit mantissa). The short format is used for high-speed processing, and the long format for extremely high precision. A complete discussion of the floating-point option can be found in the Sigma 5 Computer Reference Manual. (This manual and other pertinent documents are listed in Appendix B.)

2. Separation in Input/Output Processor

The Sigma 5 DAS is equipped with a multiplexing input/output processor (MIOP) which allows concurrent operation of I/O devices. The MIOP controls all I/O operations (with the exception of the digital I/O), hence leaving the central processing unit (CPU) free to control program execution and perform calculations. Both input and output operations can be carried on simultaneously.

3. Storage Capability

The storage capability of the DAS is:

<u>Device</u>	<u>Storage Capacity</u>
RAD (2 units)	3 megabytes each (6 megabytes total)
Magnetic Tape (3 units)	9 tracks @ 800 bits/in.-track (tape length variable)
Magnetic Core	32 k <sup>a</sup>

---

$$^a k = 2 \times 10^{10} = 1024.$$

G. Software Format

The instruction-word format of the Sigma 5 software consists of a 32-bit word. This word can also be broken into two 16-bit halfwords or four 8-bit bytes. A double word which is two 32-bit words combined into 64 bits is also available.

The memory-addressing instruction consists of a 32-bit word sectioned as follows: Bit 0 is used for indirect addressing. A zero in this position

implies that indirect addressing is not to be performed; "1" implies indirect addressing is to be performed. Bits 1 through 7 are used to designate the operational code to be performed. For example, a "30" in bit locations 1 through 7 would indicate an Add Word instruction is to be performed. Bits 8 through 11 are used to designate any one of 16 general registers as an operand source, a destination, or both. Bits 12 through 14 designate any of registers 1 through 7 as an optional indexing register. Bits 15 through 31 indicate the initial address of the instruction operand. A complete discussion of the instructions can be found in the Sigma 5 Reference Manual.

## II. SYSTEM CONFIGURATION

### A. Realtime Batch Monitor

The major software-control element of the operating Sigma 5 system is the realtime batch monitor (RBM), which was supplied by XDS. The RBM performs all supervisory and control functions. It acts as a coordinator that controls all program execution and flow both in the foreground and the background. The monitor preserves the priority of realtime tasks, runs background programs when the foreground is idle, and provides I/O services for the entire system. The RBM is discussed fully from the standpoint of the programmer/operator in the RBM Reference Manual and RBM Operations Manual (see Appendix B). Several other software packages furnished by XDS are closely related to the RBM and are briefly discussed below.

#### 1. Job Control Processor (JCP)

The JCP reads and processes control command input from an input device until a command is encountered to load a compiler or user program. It then loads the compiler or program from the RAD into core and then causes compilation or execution to start.

#### 2. RAD Editor (RAEDIT)

The RAEDIT does as its name implies: it edits or controls all information going to or from the RAD. It allocates areas for storage of



foreground and background data, saves or deletes data, and performs other utility functions as requested by the RBM. RAD allocation is discussed under Sect. II.B.

### 3. Overlay Loader (OVLOAD)

The OVLOAD creates programs in segmented form, referred to as overlay form, for later execution in the foreground or background. This segmentation allows programs to be run that are much larger than the available core space. An overlay program consists of a root, which loads the code, and several segments, which are created from one or more object-modules output from a compiler or processor such as MACRO-SYMBOL or Fortran IV-H.

### 4. Compilers or Processors

Two compilers, MACRO-SYMBOL and Fortran IV-H, are incorporated into the software provided by XDS. The first is a two-pass compiler that outputs programs in Sigma 5 standard object language, sometimes referred to as assembly language. Fortran IV-H, a one-pass compiler, compiles in the background, but is capable of realtime operation.

## B. RAD Area Allocation

The RAD units are set up and maintained by the system software. The area allocation and organization are shown in Fig. 1. The numbers "0" and "1" indicate the RAD unit to which the area is assigned.

The FP Area, or foreground program area, contains the following software:

1. Foreground Programs (realtime software)
2. User Libraries
3. Public Libraries

The SP Area, or system program area, contains the software of the XDS systems. Specifically, the SP Area contains:

1. The RBM
2. MACRO-SYMBOL and Fortran IV-H Compilers

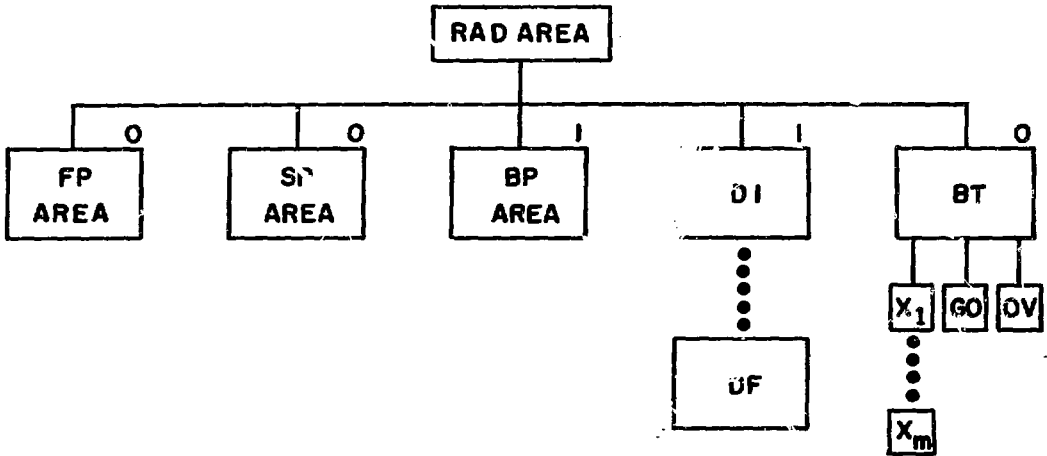


Fig. 1. RAD Area Allocation and Organization

3. Systems Library
4. RAD Editor (RAEDIT)
5. Overlay Loader (OVLOAD)
6. Job Control Processor (JCP)

The BP Area, or background program area, is that area allocated for background utilization.

The D1-DF Areas are 15 data-storage areas used for both background and foreground storage. D1-DF is in hexadecimal notation.

The BT Area, or background temporary area, contains several temporary scratch files ( $X_1-X_m$ ) used for temporary background storage during processing of background programs. These files are destroyed at the end of each job step unless they are requested to be saved.

The BT area also contains a file referred to as the GO file, which contains relocatable object modules formed by a compiler such as Fortran IV-H. It is used primarily for assembly and "go" types of programs.\* Another temporary file in the BT area, the OV file, contains executable programs formed by the overlay loader.

---

\* In a "go" program, compilation and execution are combined in one operation.

### III. SOFTWARE ALLOCATION

#### A. Software Residing in Foreground

The DAS-prepared software residing in the foreground can be classified into three major groupings: (1) time-actuated software, (2) internal-event-actuated software, and (3) external-event-actuated software. The following discussions present a brief insight into the classifications of this software.

##### 1. Time-actuated Software

Time-actuated software is that software clocked to execute on specified time intervals. For example, the data-acquisition program (DACQ) is, among other things, clocked to sample the outlet temperature of the subassembly in the 1A1 core position of the reactor every second. The time-actuated software is illustrated in Fig. 2. Basically the software functions as follows.

Program TIMER, a realtime scheduler, provides all realtime clocking and interrupt control. TIMER is responsible for controlling sampling of plant parameters, updating stored data, and controlling data display.

The first step in data acquisition is the gathering, or retrieval, of data by the multiplexers (MUX). After the data have been retrieved, they are stored in a data table (DATATAB) in the resident foreground in the core. The data in DATATAB are classified according to sample intervals (e.g., 0.5 sec, 1.0 sec) and are then checked to determine if they are within limits. If the data are outside specified limits, appropriate action is taken as shown in Fig. 2.

Several calculations, using data from DATATAB, are made and displayed on the alpha-numeric (A/N) displays in the reactor control room. The calculations are continuously updated as the data are updated. The calculations made are shown in Fig. 2. The dotted lines indicate direct input from other calculations. Utility programs such as least-squares fitting are available to other programs upon request. Raw data are stored continuously

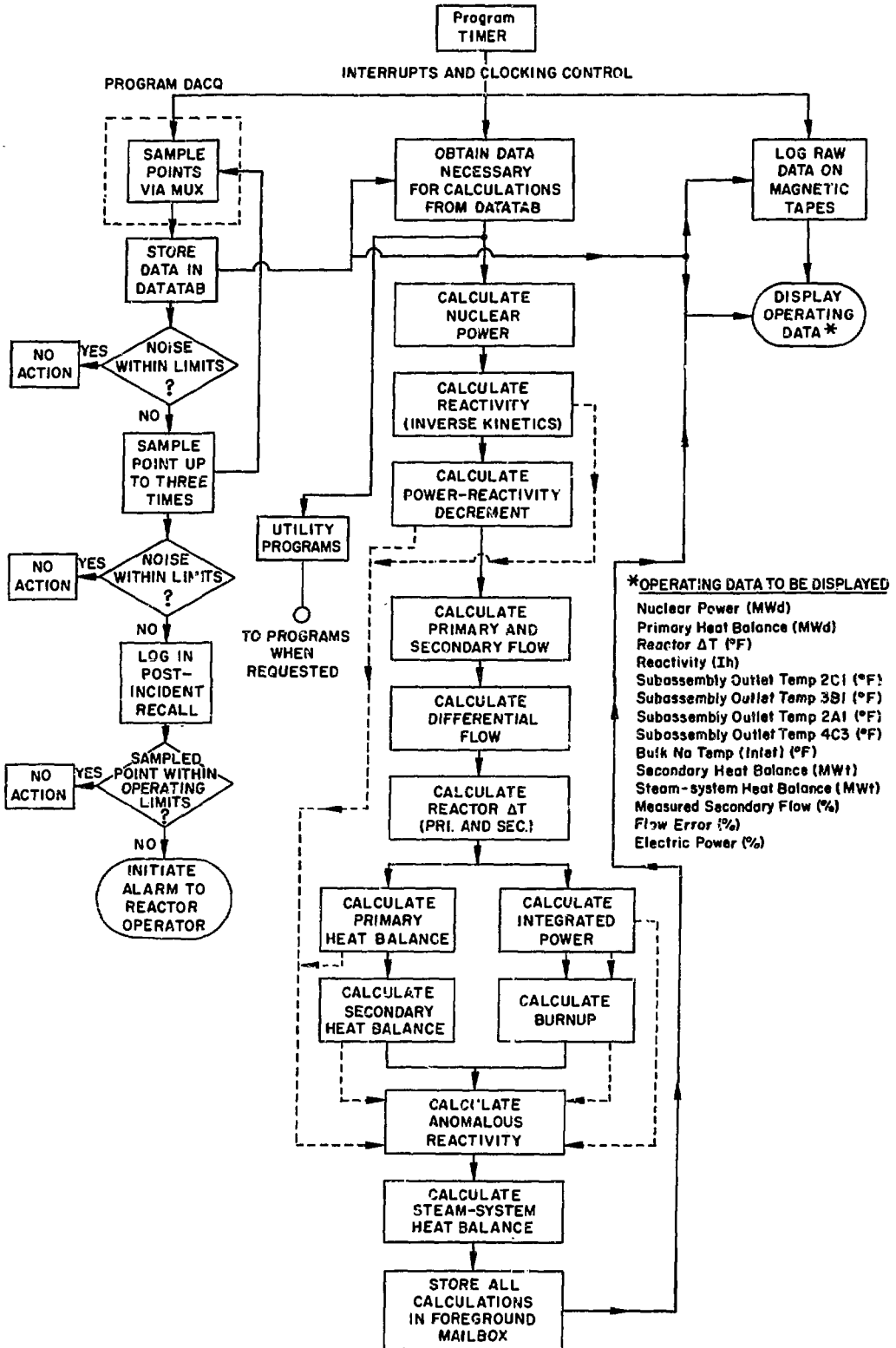


Fig. 2. Time-actuated Software

on magnetic tape so they will be available for later examination and analysis if required.

Data are taken both from DATATAB and the foreground mailbox and are displayed on a lock-out keyboard display for the reactor operators. The data displayed are listed in Fig. 2. These data are continuously updated, and the date and time of each updating are listed. This assures the operator that his operating data are being continuously revised.

## 2. Internal-event-actuated Software

Internal-event-actuated software are those programs the reactor operator may request via an A/N keyboard display. There are two of these displays (see Fig. 3), both with open keyboard (i.e., not locked out). In

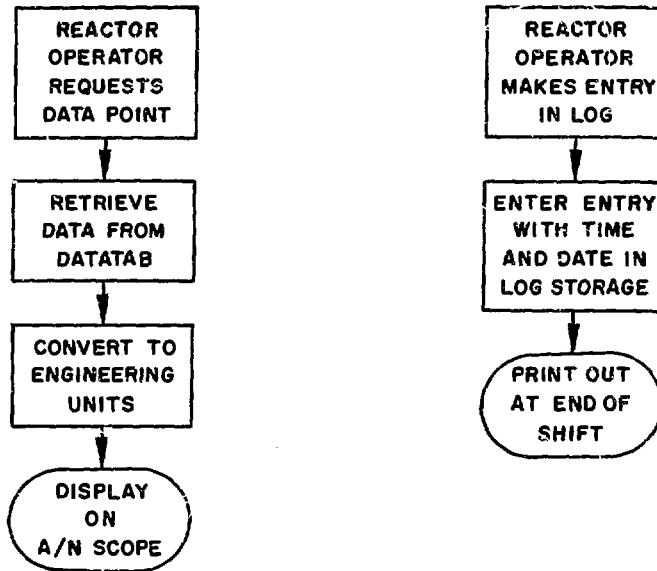


Fig. 3. *Internal-event-actuated Software*

the first, the reactor operator may request that any data point be displayed on the A/N scope. Up to five points may be displayed at one time. In the second display, the reactor operator may log entries into the shift-log program. These are entered with the date and time. The operator may at any time ask for a printout of the log on the line printer. Editing and modification of log entries are possible.

### 3. External-event-actuated Software

External-event-actuated software may be categorized into two major subgroups: (1) software actuated by an external event and (2) software actuated by manually triggering an interrupt. An example of the first subgroup is the SCRAM program, which is actuated by a reactor scram. An example of the second is the software used by experimenters. Both of these examples are shown in Fig. 4. The SCRAM program could branch to an optional program such as a fission-gas-monitoring program for supplementary analysis.

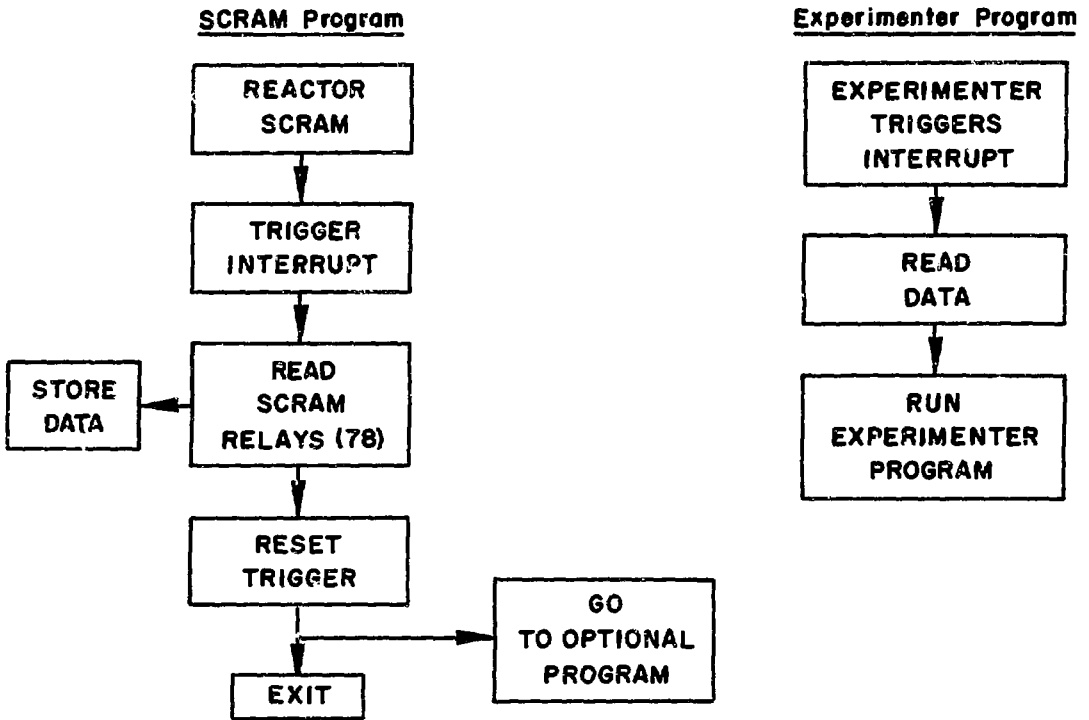


Fig. 4. Examples of External-event-actuated Software

### B. Background Support Programs

Up to this point, all software allocation discussed has been to the foreground. As mentioned in Sect. II.B, a background area (of approximately 11 k) has been allocated by the RBM for running batch programs when the foreground is not active. The background will be used primarily for programs

not requiring immediate execution. Analytic programs can be run after data have been stored on tape or on punched cards. The foreground, however, can interrupt the background at any time. The background run sequence is shown in Fig. 5.

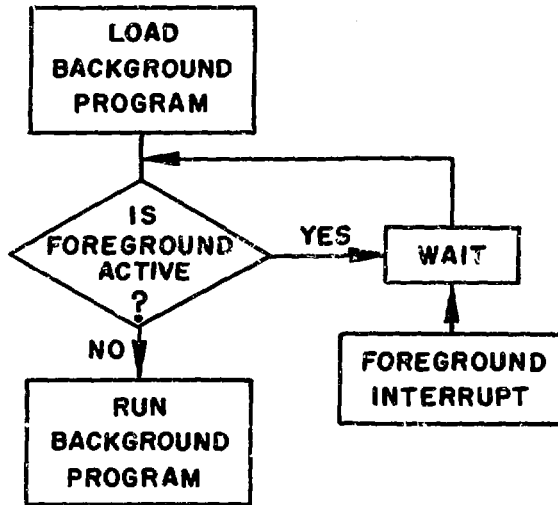


Fig. 5. Background Run Sequence

The priorities for background runs are:

1. Experiment Programs
2. Reactor-analysis Support Programs
3. General Programs

The computer operator is the final authority on the priorities of background runs. He may change them if necessary.

C. Assignment of Resident and Nonresident Foreground

As previously mentioned, the foreground is divided into resident foreground (stored in the core) and nonresident foreground (stored on the RAD).

Listed below are the program assignments for the foreground.

Resident Foreground

1. RBM
2. Program TIMER
  - a. Program DACQ (data acquisition)
  - b. Nuclear Power

- c. Reactivity (all codes)
  - d. Heat Balances (primary and secondary)
  - e. Reactor  $\Delta T$
  - f. Flow Codes
  - g. Post-incident Recall
  - h. Data Logging
  - i. Power-Reactivity Decrement
  - j. Utility Routines
  - k. Burnup
  - l. Integrated Power
3. DATATAB (foreground data table)
  4. Sequence of Events
  5. Operating Data Display Code (locked-out A/N scope)

Nonresident Foreground

1. Listing of Data for Display (MENU)
2. Shift Log
3. Control-rod Calibration
4. Extended Data Table
5. Steam-system Heat Balance
6. Data-point-display Code (unlocked A/N scopes).

#### IV. SOFTWARE MAINTENANCE

Software is continuously being modified and generated by the EBR-II Project. New and modified software will be documented and distributed to the users.

XDS software is also modified for correction and improvement. These modifications are included in an XDS Software Maintenance Report distributed periodically.



APPENDIX A

DAS Programs Generated and Proposed to Date  
by EBR-II Project

<u>Program Name</u>	<u>Core Area</u>	<u>Comment</u>	<u>Resident ?</u>
ASCON	Foreground	Converts hexadecimal floating-point numbers to decimal and then to ASCII. Format is either fixed-point or scientific notation, depending on the magnitude of the number converted.	No
AUTO	Background	Auto-correlation of data stored on tape.	No
BIG	Foreground	Debugs handler for graphics display.	No
CEU	Foreground (publib)*	Converts any data stored in DATATAB into engineering units.	No
COCH	Foreground (publib)	<u>COC handler</u> . Transmits and receives characters for the A/N scopes. Converts from EBCDIC to ASCII and vice versa.	Yes
CONEB	Foreground (publib)	Converts hexadecimal floating-point numbers to decimal and then to EBCDIC scientific notation--maximum resolution of six decimal digits.	Yes
CROSS	Background	Auto-correlation of data stored on tape.	No
DACQ	Foreground (publib)	Subroutine to get data from low-level multiplexer	Yes
DATATAB	Foreground (publib)	Resident in-core data table in which data from low-level multiplexer are stored.	Yes
FANAL	Background	Fourier-analysis code.	No
FFT	Background	Fast Fourier transform.	No
GDATA	Foreground	Samples one channel on the low-level multiplexer. Uses the SAMP program with a MUX data word at location X'504C'. Sample rate can be varied by inserting the desired frequency in Hz in location X'7052'. The program takes 8191 samples of the channel selected by SAMP at the rate specified in location X'7052'. The data are then dumped on magnetic tape (unit 9TAB0) for subsequent evaluation by other programs.	No

\*"Publib" is a library in storage that users may call as subroutines or functions.

<u>Program Name</u>	<u>Core Area</u>	<u>Comment</u>	<u>Resident ?</u>
HIST	Foreground	Run by use of the SAMP program with MIX data word at location X'670E'. This program is used for noise analysis of incoming signals on the low-level multiplexer. On a given channel, 128 samples are taken and used to calculate an expected mean; sample rate is 500/sec. The program then takes 10,000 more samples and constructs a histogram table in memory, calculates the mean deviation, the true mean, and the standard deviation of the samples taken.	No
INK	Background	Takes rod-drop data off tape and calculates worth of drop rod using the inverse-kinetics calculation.	No
INVK	Foreground	inverse-kinetics calculation for use by assembly-language users.	No
MEAN3	Foreground	Calculates mean and standard deviation for a channel in the front end. Prints the mean and standard deviation on printer and displays them on scope.	No
MUXTEST	Foreground	Determines offset and drift of low-level multiplexer, 128 channels at a time.	No
PCFT	Background	Polynomial curve fit.	No
PHONY	Foreground	Samples high-level multiplexer at a 10-kc rate for 1 to 10 channels and then writes data on tape.	No
PLOT1	Background	Used in conjunction with HIST, this background program retrieves the data calculated by HIST and converts and plots these data on the plotter. The data can also be placed on magnetic tape for easy storage.	No
PRNT	Foreground	Checkpoints background and prints shift log on line printer.	When Needed

<u>Program Name</u>	<u>Core Area</u>	<u>Comment</u>	<u>Resident ?</u>
RDRP	Foreground	Code used to sample two channels of data on the high-level multiplexer and store on tape when triggered by an external interrupt.	No
KODCAL	Foreground	Designed for use by reactor operators to measure the worth of two control rods by the rod-bump method. The program is loaded by use of the RCAL command on the A/N scopes.  Reactor power (ion chamber) and two rod positions are measured every 1/2 sec. Reactivity is calculated using inverse kinetics. As the rods are bumped, position and reactivity associated therewith are stored in memory, and the last reactivity increment is displayed on the A/N scope. When the calibration is done, the program releases itself and displays the total worth of both rods on the scope.	No
RODPLOT	Background	Used in conjunction with RODCAL to retrieve the data from memory and make a plot of rod worth vs position of the two rods.	No
RODROP	Foreground	Realtime version of rod-drop code. Calculates reactivity as soon as a data point is sampled.	No
ROSCILL	Foreground	Determines the transfer function and the feedback components using the rod oscillator.	No
RTFT	Foreground	On-line Fourier transform. Handles a maximum of 7 points/sec. Calculates 4097 coefficients.	No
RTFT2	Foreground	On-line Fourier transform. Handles 100 points/sec. Calculates 257 coefficients.	No
TIMER	Foreground	Software-timing routine to trigger interrupts--resolution of 0.05 sec.	Yes
TINT	Foreground	Main software driver for data acquisition. Includes drivers for sampling data at specified time intervals plus all limit checking on these data.	Yes

<u>Program Name</u>	<u>Core Area</u>	<u>Comment</u>	<u>Resident ?</u>
PVH	Foreground	Primary vocabulary handler for scopes. Checks to see if code word entered from scope is in the vocabulary; if it is, the appropriate overlay (see below) to handle the associated function is loaded into core. Maintains a 30-sec timer for use in realtime displays.	Yes
<u>PVH Overlays</u>			
DATA		Allows display of reactor parameters along with alphabetic description--30-sec update.	
DISP		Displays entries from shift log.	
EDIT		Allows operator to edit messages in shift log.	
INIT		Displays alphabetic description of parameters on reactor monitor scope.	
MESS		Handles entry of messages in shift log.	
MNEM		Dumps core in assembly-language mnemonics on scope.	
OURS		Handler for on-line debugging.	
RCAL		Loads rod-calibration program.	
SAMP		Allows entry of gain and channel number for sampling data from front end.	
UPDTRI		Displays and updates numeric information on reactor monitor scope.	
<u>Utility Routines</u>			
BCDEB	Background	Reads BCD (binary coded decimal) cards, converts to EBCDIC, and punches converted cards.	No
BINPLOT	Background	General-purpose plotting routine.	No

<u>Program Name</u>	<u>Core Area</u>	<u>Comment</u>	<u>Resident ?</u>
CONTEST	Foreground	Tests program CONEB.	No
LISTPR	Background	Reads cards (any format) and prints them on printer.	No
RTFIVH	Foreground	Tests realtime features of Extended Fortran IV-H.	No
TET	Foreground	Tests program TIMER.	No
WLRAD	Background	Writes alphabetic description of reactor parameters on RAD for use by PVH overlay DATA.	No
WRAD	Background	Writes assembly-language mnemonics on RAD file for use by the PVH overlay MNEM.	No

APPENDIX B

List of Applicable Reference Documents on Software for the  
Sigma 5 Computer

<u>Document</u>	<u>Content</u>
1. RBM Reference Manual No. 901581B (April 1970)	RBM procedures, methods, and programming requirements from the standpoint of the programmer analyst.
2. RBM Operations Manual No. 901647A (Feb. 1970)	RBM/operator communications and procedures from the standpoint of the Sigma 5 operator.
3. Sigma 5 Computer Reference Manual No. 900959C (Sept. 1968)	Software procedures, requirements, methods, and instructions for the Sigma 5.
4. Extended Fortran IV-H Reference Manual No. 900966E (March 1970)	Fortran programming manual for the Sigma 5.
5. Extended Fortran IV-H Operations Manual No. 901144C (Sept. 1969)	Operator information for use of Fortran IV-H compiler.
6. MACRO-SYMBOL Reference Manual No. 901578A (Oct. 1967)	MACRO-SYMBOL programming methods and instructions.
7. MODEL 7450 Buffered Line Printer Reference Manual No. 901601A (Nov. 1969)	Instructions for operating and programming.
8. Model 7121 Card Reader Reference Manual No. 900970D (May 1970)	Instructions for operating and programming.

<u>Document</u>	<u>Content</u>
9. Model 7012 Keyboard/ Printer Reference Manual No. 900974C (Dec. 1968)	Instructions for operating and programming.
10. Models 7611-7616 C0C Reference Manual No. 900981B (Oct. 1969)	Instructions for programming.
11. Model 7204 RAD Reference Manual No. 900979C (Oct. 1969)	Instructions for programming.
12. Models 7320-7322 9-Track Magnetic Tape System No. 900977B (Aug. 1968)	Instructions for operating and programming.
13. Model 7165 Card Punch Reference Manual No. 901567A (July 1969)	Instructions for operating and programming.
14. Fortran IV-H Library Technical Manual No. 901138B (Sept. 1969)	Description of and calling sequence for Fortran IV-H library.

REFERENCE

1. J. M. Allen et al., Functional Description of the EBR-II Digital Data Acquisition System, ANL/EBR-029 (November 1970).