2

AMHERST SYSTEMS INC.

132 CAYUGA ROAD • BUFFALO, NEW YORK 14225

ADVANCED
TACTICAL ELECTRONIC WARFARE
ENVIRONMENT SIMULATOR
(ATEWES)

REALTIME SOFTWARE
DESIGN DESCRIPTION

MAY 1983

Contract Number N00014-82-C-2136

AMHERST SYSTEMS INC.

ADVANCED
TACTICAL ELECTRONIC WARFARE
ENVIRONMENT SIMULATOR
(ATEWES)

REALTIME SOFTWARE
DESIGN DESCRIPTION

MAY 1983

Prepared by

AMHERST SYSTEMS, INC.
BUFFALO, NEW YORK 14225

Prepared for

NAVAL RESEARCH LABORATORY
WASHINGTON, D.C. 20375

CONTRACT NUMBER N00014-82-C-2136

132 CAYUGA ROAD • BUFFALO, NEW YORK 14225 • 716-631-0610

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

The software for the Advanced Tactical Electronic Warfare Environment Simulator (ATEWES) resides in the Control Subsystem which includes a CPU with 124K word memory, two 10 Mb moving head disk drives, an operator console, a printer, and interfaces to the TEWES Digital Subsystem ~~(specified in Appendix C)~~, and to a remote simulation processor ~~(specified in Appendix D)~~. The disk contains the computer programs and test data. The data files include the Scenario files ~~(specified in Appendix A)~~ which describe the environment to be simulated, Maximum Available Power, Antenna Gain and Pattern and Threshold files ~~(specified in Appendix D of operator's manual)~~, and Data Extraction (DX) files ~~(specified in Appendix B)~~ which store scenario events as processed, operator actions and simulator status information. The operator's console consists of a 24 line by 80 character CRT display and a standard keyboard. The console is used by the operator to enter commands to control the simulation, modify the scenario and display status messages.

The ATEWES real time software system consists of six separate programs designed to operate under Digital Equipment Corporations's RSX-11M Executive. The programs communicate by means of a system common area. The ATEWES software maintains the Platform file, the Emitter file and various timing counters, status flags and system parameters. The real time software provides all operator communications, as well as Scenario file input and event processing, and DX file output.

The Platform file contains position, velocity, heading, bearing and range attenuation parameters for up to 255 target platforms or sites and the platform for the EW system under test. The Emitter file contains a platform link, an emitter link, and power and scan parameters for up to 1,023 emitters. Once per second the software updates the position of each platform and recalculates each platforms bearing and elevation to the EW system and range attenuation. If ano of these values has changed by an amount equal to or greater than the simulator minimum change capability, an updated bearing, elevation, and attenuation will be output to the Digital subsystem for each associated emitter.

The Scenario file on disk contains a series of events in time order. Events include the appearance of new platforms and emitters and changes in platform motion and emitter status. Each event has a time associated with it which is used by the program to process the event at the proper time.

Operator commands are included to initialize and start the simulation, stop the simulation, speed up or slow down the scenario, set the scenario to a specific time, turn radiation on or off, control the interface to the EW system under test, and end the simulation run. The operator can also enter events to add to or change the simulation as it is running.

While the simulation is running the program records time tagged data on the disk in a data extraction file. The data recorded includes each event and operator command executed, pulse count data and any error indicators from the digital subsystem. The Data Extraction file can be used by offline analysis programs to evaluate the performance of the EW system under test.

The program also transmits a status message containing the position of the platform for the EW system under test, and other status data to the remote simulation processor upon request.

SECTION 2

SYSTEM ARCHITECTURE

The ATEWES real time software is composed of six independent tasks which communicate through system common areas. A simulation run is started by an initialization program, INIRN. This program requests and processes the data necessary to initialize the system. The program initializes the Digital Subsystem and the system common area. INIRN then schedules the remaining real time programs as needed and exits.

All operator interaction is provided by program OTASK, which is initiated by INIRN. When OTASK is initiated, it prompts the operator to enter a command code and is suspended by the Executive until a response is input. Based on the command selected, the operator is prompted to enter any parameters needed to define the command. When all data have been input, the command and its parameters are stored in common area OPERC. Program OTASK is then suspended until the command has been processed. Upon restart, OTASK displays any error code or display data returned by the processing programs and requests the next command code.

Program REALTM maintains a Platform File and an Emitter File in main memory. The Platform File contains position, velocity, range attenuation, bearing, and heading fields for up to 255 target platforms or sites and the platform for the EW system under test. The Emitter File contains platform, power, and scan data for each emitter. Once per second, the program will update the position of each platform and recalculate its range attenuation, bearing, and elevation. If any of these values has changed by an amount equal to or greater than the simulator minimum change capability, an updated bearing, elevation, and attenuation will be output to the Digital Generator for each emitter associated with the platform. Program REALTM processes all scenario events and operator commands. All data to be output to the Digital Generator is formatted and output by this program. REALTM also performs all Data Extraction file processing, including reading and recording pulse counts as well as all other DX data.

Program SFPM runs at an elevated priority. It performs all disc input for the scenario file. Scenario buffers are maintained in a system common area.

Program DEFP also runs at an elevated priority. It performs all disc output for the DX file. DX Buffers are maintained in a system common area.

Program SPCTL performs all input and output for the HP 8566A Spectrum Analyer. All command data are passed through common area ACTRL.

SECTION 3

PROGRAM INIRN

Program INIRN performs the functions necessary to initialize the system for a new simulation run. This includes initializing memory and initiating the remaining real time programs. Program INIRN is the only real time program which is directly initiated by the operator, done by entering a RUN INIRN command on the operator's console. The first action consists of terminating the remaining real time programs. (ABRTR). This insures a normal startup even in situations where the previous simulation run was not terminated normally. This is accomplished via an RSX-11M Executive directive. Programs OTASK, REALTM, SFPM, DEFP, and SPCTL are aborted.

The operator is prompted to enter the Scenario file name to be used for the current simulation run (GNAME). The Scenario file is located and the header block is read (GTHDR). If the file is not found, the input request is repeated. Otherwise, the Scenario file header is displayed on the operator's console (DISCN). The Scenario file header is used as the basis for the DX file header, beginning by adding the Scenario file name to it.

Default values are provided for many of the initialization parameters through a default data file. The file is normally stored under the name DEFALT, and is fully described in Appendix C of the ATEWES Operator's manual. The operator has the option of entering a different file name from which to retrieve default initialization data (GNAME). The initialization data are read from the appropriate file (GDFLT) and displayed on the operator's console. The operator then has the option of editing most of the default values. Editing is accomplished by first displaying a single field, and then allowing the operator to enter a new value using OPGII (integer values), OPGRI (real values), or GNAME (file names) with the edit mode enabled. The operator is then asked to enter a code specifying either further edits, or acceptance of the initialization data (OPGII).

When the initialization data have been accepted by the operator, the initialization is performed. The Power Boost, Frequency Daisable, and Azimuth and Elevation Receiver Pattern files are read into temporary storage (RDDAT). The Maximum Available Power file is read into a work array (RDINI). If any errors are encountered such as a missing or incomplete data file, the operator will still be prompted to enter a new file name as required. The Receiver Antenna Gain file is read (RDINI) and combined with a 20 LOG(F) term and the corresponding value read from the maximum Power file, as well as the Power Boost for the current frequency.The result is stored as an integer in common area OPERC. As the frequency is increasing, data from the antenna pattern files are used to determine pattern codes at 152 points over the frequency range.

The Receiver Sensitivity file is next read into the work array (RDINI). The file represents the receiver antenna sensitivity and must be subtracted from the corresponding Maximum Power file to determine threshold values, which are stored in common area OPERC. The Max Power file is also stored in common area OPERC to be used in realtime to compute received power values.

If usage of IRIG-B time has been selected, the program checks to see that the clock is on-line (TIRIG). If the clock does not respond, an error is output and the the clock is disabled.

The DX file header is now completed. The simulation run time and date are retrieved and inserted (FMDAT), and all initialization data file names are stored (PKFIL). The DX file is created and the header is written to the first block (CRDX). The remaining realtime programs are initiated (START) and INIRN exits.

3.1 SUBROUTINE ABRTR

Subroutine ABRTR is used to terminate all ATEWES realtime programs. This is to insure that all programs are intitiated normally at the beginning of each simulation run. An RSX-11M system directive is issued top abort OTASK, REALTM, SFPM, DEFP, and SPCTL. The routine delays 2 seconds to allow the aborts to take effect and returns to the caller. This routine is stored in module INSBS.

3.2 SUBROUTINE CLDX

Subroutine CLDX is used to close the DX file after the header events are output.

The routine uses standard I/O functions of the file system to close the file. This routine is stored in module MINSBS.

3.3 SUBROUTINE CRDX

Subroutine CRDX is used to create a new DX file. The routine uses standard I/O functions of the file system to create a file with the fixed name XXXXXXXXX.XXX;377. The initial size allocated to the file is determined by the value in the default file. The routine then returns to the caller. This routine is stored in module MINSBS.

3.4. SUBROUTINE GDFLT

Subroutine GDFLT is used to read all data from the default initialization file. The file is accessed and read using standard Fortran formatted I/O. If any error is detected, or insufficient data is found, an error is output. The routine then returns to the caller. Ths routine is stored in module INSBS.

3.5 SUBROUTINE GTHDR

Subroutine GTHDR is used to locate a selected Scenario file. The file is first opened and the header read and stored for future use. The emitter map is then read and stored in common area OPERC. If the file is not found, an error is returned. The routine then returns to the caller. This routine is stored in module MINSBS.

3.6 SUBROUTINE INIDS

Subroutine INIDS is used to initialize the Digital Generator. The DGU is first checked that it is ready to accept commands from the control Subsystem (DGURDY). All emitters are turned off, radiation is disabled, and the system under test mode is output. The routine then routine then returns to the caller,

3.7 SUBROUTINE RDDAT

Subroutine RDDAT is used to read all data from the receiver pattern, power boost and frequency disable files. The files are accessed and read using standard Fortran formatted I/O. If any error is detected, or insufficient data is found, an error is output. The routine then returns to the caller. Ths routine is stored in module INSBS.

3.8 SUBROUTINE RDINI

Subroutine RDINI is used to read all data from the receiver sensitivity and max power files. The files are accessed and read using standard Fortran formatted I/O. If any error is detected, or insufficient data is found, an error is output. The routine then returns to the caller. The routine is stored in module INSBS.

3.9 SUBROUTINE RFRDY

Subroutine RFRDY is used to check the readiness of the RF Generator. A set of frequency and DOA test points are read from files RFRDY.FRQ and RFRDY.DOA. A simple emitter is programmed through each frequency and DOA combination. The operator is notified of all failures. The operator has the option to continue or exit if errors are detected.

3.10 SUBROUTINE START

Subroutine START completes the initialization process begun by program ININN. The DX file is first created using the size specified in the default file (CRDX). The arrays in OPERC are initialized using data previously calculated by the main program. All remaining common data are also initialized. Programs OTASK, REALTM, DEFP, SFPM, and SPCTL are initiated via an executive directive. If all programs cannot be executed, an error is output and the program exits. The program also initiates the situation display processor DISP is it is installed. The routine then returns to the caller. This routine is stored is module INSBS.

3.11 SUBROUTINE TIRIG

Subroutine TIRIG is used to verify the readiness of the IRIG-B time code generator. The routine attempts to read the current time. An error code is returned is the timeout expires before the clock responds. The routine then returns to the caller.

SECTION 4

PROGRAM OTASK

Program OTASK provides all operator communications once the system has been initialized for a simulation run. OTASK is scheduled by START when the initialization process is complete. Program OTASK prompts the operator and then accepts a command code response (OPGII).If a menu request is specified, the command menu is displayed (CMENU) and the next command is requested (OPGII).

If a system status display is requested, OTASK retrieves the system status and displays the information on the operator's console (DSTAT).

If an enter new event command is specified, OTASK requests the event data from the operator (EDTOT) and notifies REALTM to process the data entered (OCMND). If an error is returned, the operator may edit the event and retry (EDTOT).

If a display emitter command is specified, OTASK requests the emitter, retrieves the parameters, and displays the information (DSPLY). If the data are edited to generate an emitter update, REALTM is notified to process the new event.

For SSP mode systems, if an emitter mapping status display is requested, OTASK retrieves the mapping status and displays the information on the operator's console (DSPMP).

For a display platform parameters, the operator is prompted to enter the desired platform number. If the platform is inactive, or is the EW system platform, an error is returned. If a valid platform is requested, the REALTM program is scheduled to return the display data(OCMND). The platform parameters are formatted and output (LISP1). If more emitters are pending, a prompt is issued to the operator. If a backup or cancel is entered, the program returns to the caller. Otherwise, the REALTM program is rescheduled to get the next block of emitters (OCMND). This sequence continues until all emitters for the specified platform have been processed.

For a display the active emitter list request, the REALTM program is scheduled to return the first block of emitters for display (OCMND). The emitter data are formatted and output (LISEM). REALTM is then rescheduled to get the next block of active emitters (OCMND). This process continues until all active emitters have been processed.

For a display the active platform list request, the REALTM program is scheduled to return the block of platforms for display (OCMND). The platform list is formatted and output (LISAP).

For all other commands, OTASK requests the appropriate command parameters (OPGRI,OPGII), and notifies REALTM to process the data returned (OPCMD). Upon restart OTASK checks the error flag, and displays an appropriate error message if set (ERORS). If a display RF detector count or display pulse count command was specified, the return values are displayed. OTASK then requests the next command code from the operator (OPGII).

4.1 SUBROUTINE CMENU

Subroutine CMENU is used to output the menu od ATEWES realtime commands. The is output and the routine returns to the caller.

4.2 SUBROUTINE DSPLY

Subroutine DSPLY is called to display and modify an emitter's parameters. The emitter number is first requested from the operator (OPGII) and is placed in OPERC. Program REALTM is notified to retrieve the linked platform parameters (OCMND). The associated platform's parameters are displayed (LISP). REALTM is rescheduled to get the emitter data (OCMND) which is displayed (ENEE). The operator is then asked if he wishes to edit the emitter's parameters (OPGII). If no edit is specified, the routine returns to the caller. If an edit is specified, a flag is set and the data is edited (ENEE). The edited data is redisplayed, and the edit option requested from the operator. If a further edit is specified, ENEE is invoked again with the edit flag set. When no further edit is specified, REALTM is scheduled to process the update (OCMND) and the routine returns to the caller.

4.3 SUBROUTINE DSPMP

Subroutine DSPMP is used to retrieve emitter mapping status. The map is scanned for all active operator emitters and their corresponding ATEWES DGU slot numbers. This information is displayed on the operator's console as well as the total number of DGU slots currently active. The routine then returns to the caller.

4.4 SUBROUTINE DSTAT

Subroutine DSTAT is called to display the current system status. The REALTM program is first notified to retrieve the system status (GDATA). The elapsed time is converted from double integer to real (DFLT) and displayed. The EW system platform data is output (LISP), followed by the number of active platforms and emitters, and the scenario and RF status. Pulse count totals are converted to reals (DFLT) and displayed, followed by the number of dropped emitters. The routine then returns to the caller.

4.5 SUBROUTINE EDTOT

Subroutine EDTOT is used to enter or edit all data necessary to specify a scenario event. For a new entry, the event menu is output (EMENU) and the operator is prompted to enter the event-type (OPGII). Based on the new event type, or the event type being edited, the appropriate editing routine is called as follows.

EVENT	ROUTINE
1	ENPE
2	DPE
3	ENEE
4	DEE
5	EOFE
6	EONE
.	.
.	.

4.6 SUBROUTINE LISAP

Subroutine LISAP is used to output the active platform list. The number and range for the first 80 platforms are unpacked and output. The operator is then prompted to enter a carriage return to display the next block of active platforms. The cancel command may be used to terminate the display before the entire list is output. When all active platforms have been displayed, the routine returns to the caller. This routine is stored in module LISP.

4.7 SUBROUTINE LISEM

Subroutine LISEM is used to format and display the active emitter list. The three state flags for each active emitter in the buffer are unpacked and converted to their ASCII values (blanks or O=On, B=Below Horizon, D=Detectable). The information is then output to the operator's console. The routine then returns to the caller. This routine is stored in module LISP.

4.8 SUBROUTINE LISP

Subroutine LISP is used to display a set of platform parameters. Platform parameters are unpacked (UNPK1) If the current platform is not the EW system platform, the platform number, emitter number and platform flags are displayed. In any case, platform position speed and heading are displayed. If this is not the EW system platform, the range attenuation, azimuth and elevation AOA are displayed. The routine then returns to the caller.

4.9 SUBROUTINE LISP1

Subroutine LISP1 is used to format and output platform display parameters. The platform position and velocity data are first output (LISP). The routine then unpacks the frequency and power for any active emitters returned by the REALTM program. All emitters are displayed on the operator's console. The routine then returns to the caller. This routine is stored in module LISP.

4.10 SUBROUTINE OCMND

Subroutine OCMND is used to schedule REALTM to process an operator command. KFLAG is set to cause REALTM to process the command residing in the EVENT common. The routine waits for event flag 33 to signify that processing is complete and returns to the caller. For commands directed to the DISP program, the new display parameters are stored in common area OPERC and the mode flag is updatted to flag the new data. The routine then waits for event flag 36 to signify that processing is complete and returns to the caller.

4.11 SUBROUTINE SPECO

Subroutine SPECO is used to control all operaator interaction with the HP 8566A spectrum analyzer. The operator is prompted to select the desired field or command (OPGII). For fields one through ten, the operator is prompted to enter the data field to be transmitted. The data field is formatted along with the appropriate command data for the spectrum analyzer. If no data [illegible] input, the specified field is addressed for use with the step up and d[illegible] commands.

For a manual command, the operator is prompted to enter any AS[illegible] be output to the spectrum analyzer.

For an emitter tune command, the operator is prompted to enter the desired emitter number (OPGII). If the emitter is not active, an error is output and the operator again prompted to enter the desired emitter number.

For an auto tune command, the spectrum analyzer mode is set to auto tune in common area ACTRL.

When all command data have been stored in common area ACTRL, the command is passed to program REALTM (OCMND).

SECTION 5

PROGRAM REALTM

Program REALTM has three main memory files associated with it, in addition to the Main Control Loop and processing segments. The memory files include the Emitter File, the Platform File stored in system common areas, and a Control File. The processing sections include Initialization, Simulated Time, Navigation Computer Output, Dropped Emitter Processing, Status Data Processing, Command Processing, Platform Update, and Event Processing.

5.1 CONTROL FILE

The Control File is an internal memory file used by program REALTM to store status and timing data, as well as operator command parameters. The following fields are included.

a. Simulated Time. A one word field containing the simulated time in seconds since the start of the scenario.

b. Time Increment. A one word field containing the amount by which Simulated Time is incremented. This value is one when the scenario is running at or slower than real time, and 8, 4, or 2 when the scenario is running at 8, 4, or 2 times normal speed.

c. Next Half Second. A one word field containing the number of ticks (sixtieths of a second) left in the current half second interval. Decremented every tick and reset to thirty when zero is reached.

d. Simulated Second Interval. A one word field containing the number of half seconds per Simulated Second. This value is one when the scenarior is running at or faster than normal, and 16, 8, or 4 when the scenario is running at 1/8 1/4 or 1/2 normal speed.

e. Next Simulated Second. A one word field containing the number of half seconds to the next Simulated Second. Decremented every half second and reset according to the Simulated Second Interval when zero is reached.

f. Scenario Speed. A two byte field, with one byte containing the shift code for platform velocities which is 0 unless the scenario is running at 8, 4, or 2 times normal speed, at which times it is 3, 2, or 1. The second byte is the shift code for the Simulated Second Interval, which is 0 unless the scenario is running 1/8, 1/4 or 1/2 normal speed, when it is 3, 2, or 1.

g. New Scenario Speed. a two byte field, with one byte containing a flag signifying a new speed is present, and the other byte containing the new speed code as a power of two.

h. Pulse Counter Periodicity. A one word field containing the periodicity with which the Pulse Detector Counters are to be read, stored in sixtieths of a second, with a maximum of 240.

i. Last Pulse Count Request Time A two word field containing the las pulse count request time to be output to the DX file.

j. Pulse Count Address A one word field containing physical address for the DMA pulse count transfer from the Digital Generator.

k. Read Counter Variables Flag. A one word field which is set to the Counter Variable Periodicity and decremented every tick. Counters are read when Flag is zero or less.

l. Automatic Stop Time. A one word field containing the Simulated Time in seconds at which the scenario is scheduled to be stopped, or -1 if it is to run indefinitely.

m. Repeat Loop Start Time. A one word field containing the start time in simulated seconds of a repeat loop currently in effect. Undefined if no repeat loop is in effect.

n. Repeat Loop Stop Time. A one word field containing the stop time in simulated seconds of a repeat loop currently in effect, or -1 if no repeat loop is in effect.

o. RF Status. A two byte field containing a flag showing the current status of the RF radiation and a flag showing the last operator status set.

p. Scenario Status. A two byte field containing a flag showing whether the scenario is currently running and a flag showing whether the scenario has ever been started.

q. Next Scenario Event. A four word field containing the event number, time, and memory location, and length of the next event of the scenario file.

r. Current Event A two word field to contain the number of the event and length of the event currently being processed. Event number set negative when processing operator events.

s. Emitter Flag Mask A one word field containing the operator's flag for emitter file entries when an operator event is being processed.

t. Platform Flag Mask A one word field containing the operator's flag for platform file entries when an operator event is being processed.

u. Update Flag. A one word field which is set to zero each time Simulated Time is incremented, and set to -1 each time an Update is performed.

v. EW Platform Index. A one word field containing the index address in the Platform File of the EW system platform.

w. EW Heading A one word field to contain the heading of the EW system.

x. EW Climb Angle A one word field to contain the climb angle of the EW system.

y. EW Position A six word field to contain the position of the EW system scaled for usage by the line of sight calculation.

z. Request Pulse Count. A one byte field set to one when a Pulse Count display is requested and cleared when completed.

aa. System Controls A one word field containing the current DOA test poin., system test mode, and RF status command. Output to the Digital Generator once per second.

ac. Number of Dropped Emitters. A one word field containing the number of dropped emitters since the last system status display.

ab. Last Dropped Emitter A one word field to contain the latest dropped emitter read from the Digital Generator.

ac. Latest Pulse Counts. Three two word fields containing the latest totals of pulse counts for all 1023 emitters in the currently active modes.

ad. Real Time Display Filters. A eight word field containing an attenuation high and low limit, a bearing upper and lower limit, and a frequency upper and lower limit (two words each). Limits are used for the System Status Display in determining the filtered number of active platforms and emitters.

ae. Last Display Platform - One word field containing the last platform number processed by the Display Platform command. Used when returning successive blocksof emitters for platforms with more than 64 linked emitters.

af. Last Displayed Emitter - One word field containing the next emitter number to return for the platform stored in LPLAT. If negative, all emitters for the platform have been displayed.

ag. Number of Active Emitters A one word field containing the number of currently active emitters.

ah. Number of Active Platforms A one word field containing the number of currently active platforms.

ai. Tune Next Emitter A one word flag set not zero when it is time to auto tune the spectrum analyzer to the next active emitter.

aj. Pooling Status A two word field to contain the pooling status of all Rf channels. One word contains the initial status as specified by the default file. The other word contains the current status, including changes required by dedicated emitter assignments.

ak. Dedicated Emiter Count An eight byte field containing a one byte count of the number of emitters currently dedicated to each RF channel.

al. Last Sync Emitter A one word field containing the last emitter specified as the scan sync emitter.

am. DX Event Buffer A 256 word field used to build DX events. Used for platform update events, scan update events, and pulse count events.

5.2 MONITR (CONTROL LOOP)

The Control Loop manages the execution of all of the processing subsections. High priority tasks are given preference by placing them near the beginning of the loop. The following cycle is continually executed searching for the next task to be processed.

a. Perform system initialization (INITT). This task is performed only once at program initiation.

b. Wait for event flag 40 to be set, either by the clock routine or by program OTASK.

c. If the last dropped emitter number read by the clock routine is non-zero, execute dropped emitter processing (DREM).

d. Record pulse counts in the DX file if a DMA transfer has completed (REPC).

e. If the spectrum analyzer auto tune flag is set, process the next emitter for tuning (SPNXT). If an emitter is processed, control is returned to point c. Otherwise, execution continues at point f.

f. If simulated time has reached the Automatic Stop time, record a Stop Scenario event in the DX file (RECST).

g. If Simulated Time has reached the Repeat Loop Stop Time, Backup and Start Scenario Commands are executed (REPTL).

h. Process the next operator command if the command flag in OPERC is set (OPEV).

i. Update platform positions if the Update Flag is set (UPDATE). If an Update is performed, control is returned to point c. If no Update is performed, execution continues at point j.

g. If the time of the next scenario event stored in the Control File is less than or equal to Simulated Time, and the scenario has been started at least once, the event is executed (SCENEV). If an event is processed, execution continues at point c. If no events are processed, execution continues at point b.

5.3 CLKAST (TIMING CONTROL)

Program timing is achieved through the use of an Asynchronous System Trap, (AST). An AST is scheduled to occur 60 times a second, and each time an AST occurs the program is interrupted and control is passed to a timing control section. This routine then performs all timekeeping functions. Appendix E contains a diagram of the scheduling of specific processing within a 1 second interval.

For each AST, the latest System Controls command is issued and the return data (the latest dropped emitted) is stored for use by the main control loop. Elapsed time is incremented by one. If spectrum analyzer auto tuning is active, the next tune time is decremented, and the Tune Next Emitter flag is set if the result s zero. The Read Counter Variables flag is decremented. If the result is zero, and the previous pulse count data has been processed, the counter is reset and a pulse count transfer is initiated. The pulse count request time is read and stored (GDXTM). The Next Half Second count is decremented. If the result is not zero, processing is complete. Otherwise, the count is reinitialized and the the Next Simulated Second count is decremented. If the result zero, Simulated Time is updated. If a the new scenario speed flag is set, control variables are updated using the new speed value. If the next Simulated Second will occur in .5 seconds, the Update Flag is set so that the Update begins .5 seconds before the actual time is reached.

5.3 SUBROUTINE ADVSCN

Subroutine ADVSCN is used to process an Advance Scenario operator command. The advance time is firs [illegible] fied that it is greater than the current simulated time. If it is not, an error is returned and the routine returns to the caller.

If the advance time is valid, the scenario stop time is set to the current time, and the routine waits for the scenario to be stopped. When the scenario has been stopped, a Stop Scenario event is recorded in the DX file (RECST), radiation is turned off, and the Advance Scenario command is recorded in the DX file (WRTVNT). All scenario events up to and including the advance target time are then executed. Platform positions are updated locally as time is advanced. When the target time has been reached, a normal platform update is performed to update all range, bearing, and elevation data (UPDATE). Radiation is then turned back on if it was enabled before the start of the advance command. The routine then returns to the caller.

5.5 SUBROUTINE ATAN

Subroutine ATAN is used to compute the arctangent of an angle. The arctan funtcion is determined using a binary search of a tangent table. The angle is not adjusted for quadrant effects. The routine then returns to the caller. This routine is stored in module FANGL.

5.6 SUBROUTINE BACKL

Subroutine BACKL is used to backup the scenario after it has been stopped. The platform file is reinitialized (INIPF), all emitterd are deleted and turned off in the Digital Generator, and the EW system position is reinitialized (INIEW). The scenario file is then reinitialized (INISCN). The scenario is then advanced to the backup target time using the same procedure as the Advance Scenario command. The routine then returns to the caller. This routine is stored in module ADVSCN.

5.7 SUBROUTINE BACKUP

Subroutine BACKUP is used to process an Backup Scenario operator command. The backup time is first verified that it is less than the current simulated time. If it is not, an error is returned and the routine returns to the caller.

If the backup time is valid, the scenario stop time is set to the current time, and the routine waits for the scenario to be stopped. When the scenario has been stopped, a Stop Scenario event is recorded in the DX file (RECST), radiation is turned off, and the Backup Scenario command is recorded in the DX file (WRTVNT). The scenario is then backed up to the target time (BACKL). This routine is stored in module ADVSCN.

5.8 SUBROUTINE CEFIL

Subroutine CEFIL is used to check an emitter against the current system status display filters. The frequency is read back from the Digital Generator and compared against the filter limits. The carry bit is set if the emitter lies outside the limits. otherwise, the carry bit is clear upon return. The routine then returns to the caller.

5.9 SUBROUTINE CKDED

Subroutine CKDED is used to update the pooling status as dedicated emitters are deleted. If the Dedicated flag is not set in the emitter flags word, the routine simply returns. For a dedicated emitter, the program reads the current synthesizer channel from the Digital Generator and updates the dedicated emitter count for that channel. If the count goes to zero, the channel is repooled. The routine then returns to the caller.

5.10 SUBROUTINE CLOSDX

Subroutine CLOSDX is used to initiate the close of the DX file at the end of a simulation run. The close flag is set in the DX file status word and event flag 37 is set to start program DEFP. The routine then returns to the caller. This routine is stored in module DXIO.

5.11 SUBROUTINE CLOSEF

Subroutine CLOSEF is used to initiate the close of the scenario file at the end of a simulation run. The close flag is set in the scenario file status word and event flag 38 is set to start program SFPM. The routine then returns to the caller. This routine is stored in module SCIO.

5.12 SUBROUTINE COS

Subroutine COS is used to compute the cosine of an angle. The angle is first converted to an angle between 0 and 255 BAMS. This angle is then used to look up the corresponding cosine, which is then adjusted based on the quadrant of the angle. The routine then returns to the caller.

5.13 SUBROUTINE CPFIL

Subroutine CPFIL is used to check a platform against the current system status display filters. The current DOA is checked against the DOA limits and the current range attenuation is checked against the current range limits. If either value falls outside the limits, the carry bit is set upon return. Otherwise the carry bit is cleared and the routine returns to the caller.

5.14 SUBROUTINE DAEL

Subroutine DAEL is used to generate display data for the active emitter list. The routine begins with the emitter number specified by OTASK and scans the emitter file until 140 active emitters have been found, or the end of the emitter file is found. For each emitter, the routine determines whether the linked platform is above or below the radar horizon, whether the emitter is on or off, and whether the emitter is detectable or not. These flags are packed with emitter number in the return buffer. When the buffer is full, or the end of the emitter file is reached, the routine returns to the caller.

5.15 SUBROUTINE DAPL

Subroutine DAPL is used to generate data for the active platform list. The routine begins with the first entry in the platform file and scans the entire file for active platforms. The EW system platform is excluded, as well as any platform whose range or AOA lies outside the current limits for the system status display. For each active platform within the limits, the routine returns the platform number and the range attenuation. When the end of the platform file is reached, the routine returns to the caller.

5.16 SUBROUTINE DEE

Subroutine DEE is used to delete an emitter from the simulation. The emitter number is first mapped to its corresponding Digital generator slot number. The link to the associated platform is deleted. The all emitter slots are turned off in the Digital Generator (DELEM), the event is recorded in the Dx file (WRTVNT), and the routine returns to thā caller.

5.17 SUBROUTINE DELEM

Subroutine DELEM is used to reinitialize emitter slots as they are deleted. The routine first checks the pooling status of the emitter to be deleted (CKDED). The slots are then turned off in the Digital Generator and the emitter file entries for the associated slots are reinitialized. The routine then returns to the caller. This routine is stored in module DEE.

5.18 SUBROUTINE DGUBF

Subroutine DGUBF is used to send a series of commands to the Digital Generator. Each command is output (WRTDGU) and the routine returns to the caller.

5.19 SUBROUTINE DISEMT

Subroutine DISEMT is used to gather platform and emitter parameters for real time display. The routine first determines which platform the emitter is associated with. The platform is then reconstructed as an enter new platform event in scenario format. The routine then sends this data to be displayed and waits (STOT).

Upon restart the emitter is reconstructed as an enter new emitter event in scenario format from information in the Digital Generator and the Control Computer. The routine then returns to the calling routine.

5.20 SUBROUTINE DPLC

Subroutine DPLC is used to process an operator request to display pulse counts for an emitter. The routine first checks that the requested emitter is active, and returns an error if it is not. If the emitter is active, the emitter number is stored in the Request Pulse Count flag. If the scenario has been started, the routine returns to the caller. If it has never been started, all zero pulse counts are returned and OTASK is notified of command completion (STOT). the routine then returns to the caller. This routine is stored in module PLSCNT.

5.21 SUBROUTINE DPE

Subroutine DPE is used to delete a platform from the simulation. If the platform is not active, or is the EW system, an error is returned. Otherwise the platform is marked as inactive, and all linked emitters are also deleted (DELEM). The event is recorded in the DX file (WRTVNT) and the routine returns to the caller.

5.22 SUBROUTINE DPLAT

Subroutine DPLAT is used to process a Displat Platform operator command. If the platform is inactive, or the EW system platform, an error is returned. For a valid platform, the position and velocity parameters are retrieved from the platform file. The platform number is then checked to see if it is the same as the last displayed platform. If this is the same platform, the next emitter to display is checked. If the next emitter to display is no longer linked to this platform, or is negative, the emitter list begins with the first emitter on the platform. Otherwise, the first emitter returned will be that stored in common. For a new display platform, the first linked emitter is always the first returned. Starting with the appropriate linked emitter, the routine formats emitter number, frequency, and effective output power into the return buffer for up to 64 emitters. The last displayed platform number and next emitter number are updated. The routine then returns to the caller.

5.23 SUBROUTINE DREM

Subroutine DREM processed the latest dropped emitter number read from the Digital Generator. If the number is zero, the routine simply returns. For a non-zero dropped emitter number, the emitter first determines the first slot number which is part of the operator emitter. The slot number is then mapped back to the corresponding operator emitter number (UNMAP). The emitter is then turned on or off based on its status as stored in the emitter file. The operator emitter number and Digital Generator slot number are recorded in the DX file (WRTVNT). The LAst Dropped Emitter number is cleared, and the routine returns to the caller.

5.24 SUBROUTINE DSBDX

Subroutine DSBDX is used to process an operator Disable DX Recording command. The command is first written to the DX file (WRTVNT). The Disable DX Recording flag is set ant the routine returns to the caller.

5.25 SUBROUTINE ENDSIM

Subroutine ENDSIM is used to process an End Simulation operator command. RF radiation is turned off and the command is recorded in the Dx file (WRTVNT). The Scenario File to be closed (CLOSEF) and the DX file is closed (CLOSDX). The exit flags are set for the situation display processor and the spectrum analyzer control program. The routine then sets event flag 51 to restart the ATEWES Executive Menu program and exits.

5.26 SUBROUTINE ENEE

Subroutine ENEE is used to an enter/update emitter event. The routine first verifies that the emitter's platform is active and not the EW system. The first base frequency is check against the Frequency Disable File. For an update emitter event a test is made to insure the emitter is active. A failure on any of the initial status checks generates an error message and returns to the calling routine.

For an operator enter new event the number of unused scenario slots is compared to the number required. If not enough slots are available an error message is displayed and the routine returns to the caller.

An operator update emitter event cannot use more DGU slots than the number currently is use by the emitter. Otherwise, an error message is displayed and the routine returns to the caller.

Some emitter update events require the emitter to be turned off before updating. Any emitter with a sync emitter link or any edit to an emitter with a sync emitter link must be turned off.

A buffer is built from the non-repeated event fields and the first set of the repeated fields. This buffer completely defines one DGU slot emitter, except for the linking code (See Appendix C.1). The buffer is output to the Digital Generator (DGUBF).

For an emitter using more then one DGU slot only the repeated fields are changed in the buffer before it is output again.

Based upon frequency the chirp rate field is scaled to correspond to maximum chirp attainable in the frequency synthesizer, See Table X in Appendix A.

Emitter linking code is determined based upon dependent signal status, Emitter Link Select (ELS), Link Seed Jitter Code (LSJC) and PRI. Dependent Signal emitters utilize the sync emitter link exclusively. A DGU slot with a non-zero ELS would have a sequence link of its own slot number and pattern switch link of the next slot number. the last slot would pattern switch link the first slot. A DGU slot with a non-zero LSJC will sequence link the first slot.

The remaining DGU slots sequence the next slot until the end of the chain which sequence links the first slot. DGU slots with a PRI less than 2 microseconds are the exception. these slots can only be generated using the sync emitter link. Thus the slots with a PRI 2 microseconds or greater are sequence linked together and the others are sync emitter links from these slots.

5.27 SUBROUTINE ENPE

Subroutine ENPE is used to process an enter new platform scenario event. The routine first determines whether the event is a new platform, velocity change, reposition or turn event.

For a new platform event, the routine returns an error if the platform is already active. Otherwise the positions and velocities are stored in the platform file for the specified platform. The initial range (FNDRA), bearing (FNDBE), and elevation (FNDEL) to the EW system are calculated and stored in the platform file. If an operator event is being processed, the operator's flag is set in the platform file. The event is written to the DX file (WRTVNT) and the routine returns to the caller.

For a velocity change, reposition, or turn event the routine returns an error if the specified platform is inactive. If the platform is active, new positions, velocities, or turn data are stored in the platform file. The heading and pitch change flags are set whenever different values are stored. The operator's flag is set if an operator event is being processed, the event is written to the DX file (WRTVNT), and the routine returns to the caller.

5.28 SUBROUTINE EOFE

Subroutine EOFE is used to process an emitter off scenario event. The routine returns an error if the specified emitter is inactive, or if a scenario event attempts to turn off an operator's emitter. Otherwise the emitter is mapped to its corresponding Digital Generator slot number. Each slot used to simulate the emitter is turned off, the event is written to the DX file, and the routine returns to the caller.

5.29 SUBROUTINE EONE

Subroutine EONE is used to process an emitter on scenario event. The routine returns an error if the specified emitter is inactive, or if a scenario event attempts to turn on an operator's emitter. Otherwise the emitter is mapped to its corresponding Digital Generator slot number. Each slot used to simulate the emitter is turned on, the event is written to the DX file, and the routine returns to the caller.

5.30 SUBROUTINE FNDBE

Subroutine FNDBE is used to compute the true bearing from a platform to the EW system platform using the following formula.

$$\text{Bearing} = \text{Arctan} \quad (X_p - X_o) / (Y_p - Y_o)$$

Delta X and delta Y are used to compute the tangent of the bearing angle. The angle is then determined by taking the arctan of the resultant tangent (ATAN). The angle is then adjusted for the right quadrant. The true bearing is the combined with platform heading to cumpute the relative bearing from the current platform to the EW system platform. The routine then returns to the caller. This routine is stored in module FANGL.

5.31 SUBROUTINE FNDEL

Subroutine FNDEL is used to compute the true elevation from a platform to the EW system platform using the following formula.

$$\text{Elevation} = \text{Arctan } (Z_p - Z_o) / ((X_p - X_o)^2 + (Y_p - Y_o)^2)^{.5}$$

Delta X, delta Y, and delta Z are used to compute the tangent of the elevation angle. The angle is then determined by taking the arctan of the resultant tangent (ATAN). The angle is then adjusted for the right quadrant. The true elevation is the combined with platform pitch to cumpute the relative elevation from the current platform to the EW system platform. The routine then returns to the caller. This routine is stored in module FANGL.

5.32 SUBROUTINE FNDRA

Subroutine FNDRA is used to determine the current range attenuation for a platform. A horizon check is performed for each platform if the line of sight check is enabled. The maximum separation is computed from the altitude of the EW system and the altitude of the current platform. If slant range squared exceeds the maximum allowable range, the range attenuation is set to the maximum value (127). Otherwise SR^2 is then used in a binary table search to determine the Platform Range Attenuation. The routine then returns to the caller.

5.33 SUBROUTINE GDXTM

Subroutine GDXTM is used to determine the recording time for a DX event. If the IRIG-B clock is disabled, the current elapsed time tick count is returned.

Otherwise the routine reads the current time of day from the IRIG-B time generator. The routine then returns to the caller.

5.34 SUBROUTINE GSTAT

Subroutine GSTAT is used to process an operator system status display request. The EW system platform parameters are first copied to the return buffer, followed by the range and bearing filters. The number of active platforms, and detectable emitters both filtered and unfiltered are determined and copied to the return buffer. The elapsed time, scenario time and status, and RF status are copied to the return buffer. The latest pulse count totals are copied. The dropped emitter count is copied to the buffer, and the count is zeroed. The routine then returns to the caller.

5.35 SUBROUTINE INIEW

Subroutine INIEW is used to initialize the EW system platform. The platform is marked as active, and as the EW system. All position and velocity fields are zeroed and the platofrm is linked to emitter zero. The routine then returns to the caller. This routine is stored in module INITT.

5.36 SUBROUTINE INIPF

Subroutine INIPF is used to initialize the platform file. All platforms are marked as inactive. The first emitter link is initialized to show the end of the link, and the routine returns to the caller. This routine is stored in moudule INITT.

5.37 SUBROUTINE INISCN

Subroutine INISCN is used to initialize the scenario file. The file is first set to the beginning of the file (RESETF). All events scehduled to occur at time zero are then executed (SCENEV) and the routine returns to the caller.

5.38 SUBROUTINE INITD

Subroutine INITD is used to initialize DX file access for program REALTM. The buffer pointer is set to the beginning of the first DX buffer. All Dx buffers are marked as free, and the routine returns to the caller. This routine is stored in module DXIO.

5.39 SUBROUTINE INITF

Subroutine INITF is used to initialize scenario file access for program REALTM. The buffer pointer is initialized to the beginning of the first buffer. The routine waits for the first scenario buffer to be filled and returns to the caller. This routine is stored in module SCIO.

5.40 SUBROUTINE INITT

Subroutine INITT performs all initialization for the initial startup of program REALTM. The virtual address of the pulse count DMA buffer is converted to a phisical address (VTR). The scenario file is opened for read (INITF) and the Data Extraction file is opened for append (INITD). Dynamic memory mapping is initialized (MPINI).

The platform file is initialized (INIPF), and all emitters are deleted from the Emitter File and turned off in the Digital Generator. The EW system platform is initialized (INIEW). All parameters in the Control File are initialized and the routine returns to the caller.

5.41 SUBROUTINE LOTIM

Subroutine LOTIM is used to output the current scenario time to the Digital Generator. The time is first broken down into its hours, minutes and seconds components. These values are then converted to BCD and output to the Digital generator. The routine then returns to the caller.

5.42 SUBROUTINE MPDF

Subroutine MPDF is used to map virtual address space to the Display File stored in common DFILE. If common EFILE is mapped on entry, it is unmapped. The routine then returns to the caller. This routine is stored in module MAPSUB.

5.43 SUBROUTINE MPEF

Subroutine MPEF is used to map virtual address space to the Emitter File stored in common EFILE. If common DFILE is mapped on entry, it is unmapped. The routine then returns to the caller. This routine is stored in module MAPSUB.

5.44 SUBROUTINE MPINI

Subroutine MPINI is used to initialize dynamic mapping for common areas DFILE and EFILE. The Active Page Register is computed from the virtual address assigned. The DFEF region is attached and the EFILE portion is mapped. If any errors are encountered, the routine outputs an error message and exits. This routine is stored in module MAPSUB.

5.45 SUBROUTINE NAVAST

Subroutine NAVAST is used to create and output the TEWES Navigation message, in Appendix D. The routine is started when a one byte request is issued across a RS232C serial port. The current contents of the five processing registers is saved on the stack. The message is built and a checksum is generated. The checksum is the exclusive or of all data bytes in the message. The message is output over the same serial port.

5.46 SUBROUTINE OPEV

Subroutine OPEV is used to process an operator command. The routine initializes the return error code, the current event number, and the emitter and platform flag masks if an operator entered scenario is being processed. The routine then decodes the command code and jumps to the appropriate processing subroutine. Unless the last command was a pulse count request, the current scenario time is returned, OTASK is notified of command completion (STOT), and the routine returns to the caller.

5.47 SUBROUTINE RELVNT

Subroutine RELVNT is used to return the next event from the scenario file. If all scenario input buffers are empty, the routine returns an event code of zero. Otherwise the next event in the scenario is returned and buffer pointers are updated. The routine then returns to the caller. This routine is stored in module SCIO.

5.48 SUBROUTINE REPC

Subroutine REPC is used to record the latest pulse count data in the DX file. If the latest pulse count request time is not zero, it is recorded in the DX file (WRTVNT) and reinitialized to zero. If a pulse count transfer has been completed, the pulse counts for all active emitters are formatted and recorded. For each active entry in the emitter map, the pulse counts are indexed by Digital Generator slot number. The counts for all emitter slots used to generate the emitter are totaled. If an overflow occurs for a patricular pulse count type, the total is set to the maximum value. Pulse count events are formatted with operator emitter number, followed by the total for all three pulse count types. Emmitter reports arre packed to a maximum event length of 256 words and output to the Dx file (WRTVNT). The routine then returns to the caller. This routine is stored in module PLSCNT.

5.49 SUBROUTINE REPEAT

Subroutine REPEAT is used to process an operator Initiate Automatic Repeat Loop command. The Start and Stop Times specified with the command are stored in the Control File. The scenario is advanced or backed up as needed to the Start Time. The scenario is then started and will automatically be executed between the Start and Stop Times until a Stop, Advance, or Backup Scenario Command is given.

5.50 SUBROUTINE REPTL

Subroutine REPTL is used to initiate the next pass for an automatic repeat loop. The scenario is backed up to the repeat loop start time (BACKL). The scenario is then started, and scheduled to stop at the repeat loop stop time. The start command is recorded in the DX file (WRTVNT), and the routine returns to the caller. This routine is stored in module REPEAT.

5.51 SUBROUTINE RESEFT

Subroutine RESETF is used to reset the scenario file to the beginning. The reset scenario flag is set and the routine waits for the first scenario buffer to be filled. The buffer pointer is initialized and the routine returns to the caller. This routine is stored in module SCIO.

5.52 SUBROUTINE RFCHK

Subroutine RFCHK is used to set the Rf status to the last status set by the operator. If the last operator status set was enabled, the RF enable flag is set in the System Controls word in the control file. Otherwise the flag is cleared. The routine then returns to the caller. This routine is stored in module RFCON.

5.53 SUBROUTINE RFOFF

Subroutine RFOFF is used to process an operator Disable Radiation command. The RF Status is updated in the Control File, and the event is recorded in the Data Extraction File (WRTVNT). The routine then returns to the caller. This routine is stored in module RFCON.

5.54 SUBROUTINE RFON

Subroutine RFON is used to process an operator Enable Radiation command. The RF status is updated in the control and the event is recorded in the DX file (WRTVNT). If the scenario has never been started, the elapsed time counter is started (STRTC). The routine then returns to the caller. This routine is stoerd in module RFCON.

5.55 SUBROUTINE SCENEV

Subroutine SCENEV is used to process a single scenario event. If the last event read from the scenario is type zero, the routine attempts to get the next event from the file. If a zero is still returned, the routine returns to the caller. Otherwise the event time is copmared to the current time, and the routine returns to the caller if the event time is greater. Otherwise, the current event and platform and emitter masks are initialized in the control file and the appropriate event processing routine is called. The routine then releases the next event from the scenario file (RELVNT) and returns to the caller.

5.56 SUBROUTINE SDFIL

Subroutine SDFIL is used to process an operator Set System Status Display Filters command. The new display filters are stored in the Control File, for use by all succeeding display commands. The routine then returns to the caller.

5.57 SUBROUTINE SETCP

Subroutine SETCP is used to process an operator Set Pulse Counter Periodicity command. The new periodicity is stored in the Control File and the routine returns to the caller. This routine is stored in module PLSCNT.

5.58 SUBROUTINE SETDOA

Subroutine SETDOA is used to process an operator Set DOA Test Point command. The old test point is cleared from the current system controls. The new test point is then merged with the remaining system controls. The routine then returns to the caller.

5.59 SUBROUTINE SETSPD

Subroutine SETSPD is used to process an operator Set Scenairo Speed command. The New Speed Flag is set and the speed code is stored in the Control File, to be implemented by the timing routine at the next Simulated Second.

5.60 SUBROUTINE SIN

Subroutine SIN is used to compute the sin of an angle. The angle is decreased by 256 BAMS and the cosine of the result is computed (COS). The routine then returns to the caller.

5.61 SUBROUTINE SPNXT

Subroutine SPNXT is used to process the next emitter for automatic tuning of the spectrum analyzer. The routine first checks that there is at least one active emitter, and returns if there are none. If there are any active emitters, the routine searches the emitter file biginning at the last emitter which was automatically tuned. When an active emitter is found, the DOA test point is automatically set based on the DOA of the linked platform. The routine then reads the emitter's PRI, scan type, scan rate, frequency, frequency agility, and frequency modulation amplitude from the Digital Generator. These values are stored in common block ACTRL and event flag 42 is set to start program SPCTL. The routine then returns to the caller. This routine is stored in module SPRTM.

5.62 SUBROUTINE SPRTM

Subroutine SPRTM is used to process an operator Spectrum Analyzer control command. The routine first waits for the spectrum analyzer busy flag to be cleared. For an emitter tune command, the routine first checks to see that the emitter is active, and returns an error if it is not. If the emitter is active, the routine loads the required data for the tune command (SPNXT) and returns to the caller. For a manual spectrum analyzer command, the routine sets event flag 42 to start program SPCTL, sets the spectrum analyzer busy flag, and returns to the caller.

5.63 SUBROUTINE STCEN

Subroutine STCEN is used process an operator Start Scenario command. The Scenario Status is first checked to see if the scenario has previously been started. If so, the Scenario Status is updated to show the scenario running. The timing routine will then automatically begin updating Simulated Time, and real time processing will take place. If the scenario has never been started, all events at time zero are executed (INISCN) and the AST request is initiated (STRTC). The radiation is turned on unless a Radiation Off command has been processed (RFCHK). If a stop time has been included, the value is stored as the Automatic Stop Time in the Control File. If no stop time is specified, this value is set to the highest possible number so the scenario will run indefinitely. The event is written to the DX file (WRTCNT) and the routine returns to the caller. This routine is stored in module SCCON.

5.64 SUBROUTINE STOT

Subroutine STOT is used to notify program OTASK of command completion. Event flag 33 is set to notify OTASK of the completion status. The routine then returns to the caller.

5.65 SUBROUTINE STPSCN

Subroutine STPSCN is used to process an operator Stop Scenario command. The current Simulated Time is placed in the Automatic Stop Time field in the Control File. The Repeat Loop Start and Stop Times are reinitialized. The routine then returns to the caller. This routine is stored in module SCCON.

5.66 SUBROUTINE STRTC

Subroutine STRTC is used to perform initial scenario startup processing. A pulse count transfer is initiated to clear all counters in the Digital Generator. The initial clock AST is requested and the scenario status is updated. The routine then returns to the caller. This routine is stored in module SCCON.

5.67 SUBROUTINE TURNZ

Subroutine TURNZ is used to update a platform which is executing a turn. The turn rate is scaled by the current scenario speed and added to the the platform heading. The resultant heading is scaled to BAMS and compared to the old heading value in BAMS. If a change is detected, the heading change flag is set. The new X and Y velocities are computed from the speed and heading (SIN,COS). The remaining time in the turn is updated, and the routine returns to the caller.

5.68 SUBROUTINE UNMAP

Subroutine UNMAP is used to convert a Digital generator slot number to an operator emitter number. The routine searches the operator to Digital Generator slot mapping table to determine the operator emitter number. The routine then returns to the caller.

5.69 SUBROUTINE UPDATE

Subroutine UPDATE performs position updates and calculates geometry for all active platforms. The platform Update is executed each time the Update Flag is set by the timing routine. Execution is once per second when the scenario is running at or faster than normal speed. Execution is every 2 seconds for 1/2 speed, 4 seconds for 1/4 speed, and 8 seconds for 1/8 speed. If the scenario is running at 2, 4 or 8 times normal speed, velocities are shifted left 1, 2 or 3 bits respectively before being used in any calculations.

If a turn is in progress, the heading and velocities are updated (TURNZ). Positions are updated by adding velocity components to their associated position coordinates. This is first done for the EW system platform. The coordinates are truncated to 16 bits of accuracy, with the least significant bit representing 128 meters, and stored for later use in range calculatons. If the change heading of pitch flags are set, new values are output to the Digital Generator.

For each of the other active platforms, the position is updated in the same way. The new coordinates are then scaled to match the stored coordinates for the EW system platform. These values are used to calculate the Slant Range Squared with the following formula.

$$SR^2 = (X_p - X_o)^2 + (Y_p - Y_o)^2 + (Z_p - Z_o)^2$$

The p subscripts represent the current platform and the o subscripts represent the EW system platform. Slant Range is then used to compute the range attenuation (FNDRA). New bearing (FNDBE) and elevation (FNDEL) angles are calculated.

The new Range Attenuation, Bearing, and Elevation are compared to the values stored in the Platform File. If neither has changed, processing for the platform is complete. If either has changed, the new values are stored in the Platform File and output to the Digital Generator for each emitter linked to the platform. The Bearing and Elevation output for each emitter are the values calculated for the platform. The Attenuation output for each emitter is the sum of the Platform Range Attenuation and the Minimum Range Attenuation field stored in the Emitter File. If the emitter is tracking the EW system in either azimuth or elevation, the sector width is read back from the Digital Subsystem. Half the sector width is subtracted from the bearing or elevation to the EW system and the result is output as the new scan lower limit.

If Update Recording is enabled, the platform number, range attenuation, and azimuth AOA and elevation AOA for all platforms which changed are stored in a temporary buffer. When all platforms have been updated, the buffer is written to the DX file.

The program then updates the scan lower limit for all emitters tracking platforms other than the EW system. For each tracking emitter, DELTA X, DELTA Y, and DELTA Z are determined using the linked platform and the tracked platform. The relative bearing (FNDBE) and elevation (FNDEL) to the tracked platform are computed. One half the sector width is subtracted to compute the new scan lower limit. The old scan lower limit is read from the Digital Generator. If both values are the same, processing for the current emitter is complete. If the value has changed, the new value is output to the Digital Generator. The emitter number and scan limit are stored in a DX event buffer. If the buffer is full, a partial event is recorded in the DX file (WRTVNT) and the buffer is cleared. When all emitters have been processed, the final event is recorded in the DX file.

5.70 SUBROUTINE VTR

Suroutine VTR is used to convert a virtual address to a physical address. The three MSB of the virtual address are used to determine the corresponding Active Page Register. The Active Page Register is then read to determine the physical base address of the page. This base is added to the offset in the twelve LSB of the virtual address to determine the final eighteen bit physical address.

5.71 SUBROUTINE WRTDGU

Subroutine WRTDGU is used to output a single command value to the Digital Generator. The command value is output to the DR11-M parallel interface. The routine then delays approximately twelve microseconds to insure proper command output timing. The routine then returns to the caller.

5.72 SUBROUTINE WRTVNT

Subroutine WRTVNT is used to output an event to the DX file. The routine first checks that buffer space is available for the event, and waits until a buffer is free if there is none. The event is then copied to the DX buffer and the buffer pointers are updated. If any DX buffery are filled, event flag 37 is set to start program DEFP to perform the disc output. The routine then returns to the caller. This routine is stored in module DXIO.

SECTION 6

PROGRAM DEFP

Program DEFP performs all DX output to the disc. All DX events are buffered in common block DXBF. Each time program REALTM fills a 256 word buffer, the buffer status flags are updated and DEFP is scheduled.

Program DEFP appends to a DX file with a fixed name of XXXXXXXXX.XXX;377 when it is initiated. The program then waits for event flag 37 to be set by program REALTM. Each time the flag is set, DEFP checks status flags in OPERC to determine the required action. If the exit flag is set, all full buffers are output. The file is then truncated to release any space allocated to it which is not required. If no output file name was specified by the operator, the temporary file is deleted. Otherwise, the file is renamed using the operator specified name from common area OPERC. If any error is detected, an error code is returned in KERR and the program waits for event flag 37 to be set again. When the DX file has been successfully deleted or renamed, the program exits.

If the exit flag is not set, the program outputs any full DX buffers using standard I/O functcions of the file system. As each buffer is output, its status flag is updated in DXMAP. Additional disc blocks are allocated in segments of 50 blocks as required. When all disc buffers have been emptied, the program returns to the wait state for event flag 37.

SECTION 7

PROGRAM SFPM

Program SPFM performs all scenario file input from the disc. All scenario buffers are stored in common block SCBF.

When the program is initiated, it opens the file name specified in OPERC and sets its block pointer to 6, to bypass the scenario header and emitter map. All scenario buffers are filled, and the program waits for evbent flag 38 to be set.

When event flag 38 is set, the program checks status flags in OPERC to determine the required action. If the exit flag is set, the program closes the scenario file and exits. If the rewind flag is set, the program resets the current block pointer to 6 and fills all scenario buffers starting at the beginning of the file. The program then returns to the wait state for event flag 38.

If no special flags are set, the program fills any empty scenario buffers from the disc. As each buffer is filled, its status flag in SCMAP is updated. When all input buffers have been filled, the programm returns to the wait state for event flag 38.

SECTION 8

PROGRAM SPCTL

Program SPCTL performs all communications with the HP 8566A spectrum analyzer. The program outputs all command strings and reads back and displays required data.

The progrm normally waits for event flag 42 to be set. Each time the flag is set, the program processes the command currently stored in common block ACTRL. For manual analyzer commands, either an operator entered ASCII string or a general menu selection, the program outputs the command string and returns to the wait state for event flag 42.

For an emitter tune command, the program resets the spectrum analyzer, and encodes and outputs a new center frequency based on the frequency of the specified emitter (TALK). The frequency span is then set based on the amount of periodic modulation amplitude, frequency agility, or to a minimum of 5 MHZ. (TALK). The program then sets a reference level of +10 dBm (TALK). The program then reads the peak frequency and power measured by the spectrum analyzer (RFRPW). These values, along with the emitter number, PRI, and scan period, are formatted and output to the spectrum analyzer display (TALK). The scan period is then used to compute the dwell time before the sequencing to the next active emitter if auto tuning is enabled. The program then returns to the wait state for event flag 24.

8.1 SUBROUTINE LISTEN

Subroutine LISTEN is used to input ASCII strings from the spectrum analyzer. A system QIO request is issued to the IEEE interface device handler to perform the input. The routine then returns to the caller.

8.2 SUBROUTINE RFRPW

Subroutine RFRPW is used to read the current peak power and associated frequency from the spectrum analyzer. The analyzer is instructed to perform a peak search (TALK). The peak power and frequency are then input (LISTEN). the routine then returns to the caller.

8.3 SUBROUTINE TALK

Subroutine TALK is used to output ASCII command strings to the HP 8566A spectrum analyzer. A system QIO is issued to the IEEE interface device handler to perform the output. The routine then returns to the caller.

APPENDIX A

SCENARIO FILE DESCRIPTION

The Scenario File is made up of a series of variable length records, each describing a single event. Events are stored in chronological order, with platform events in order of increasing platform number preceeding emitter events in order of increasing emitter number when events have equal time fields. The records are packed into 256 word blocks, with each event record contained completely in one block. Any unused words at the end of a block are filled with zeroes, so that no event crosses a block boundary. Blocks of data are written to a FILES-11 format file, with one 256 word record comprising a data block. The Scenario File is produced by an offline program and stored on a diskpack to be read as needed by the real time software.

The first block (256 words) of the Scenario File contains the file header, which includes the following data.

WORD	CONTENTS
2	File Type = 500 + Revision Level
3	Scenario Creation Date
4	Scenario Creation Time (Minutes)
4-39	Scenario Operator Comment

All fields not described are reserved for use by future programs. All ASCII fields are stored with the first character in the high byte and the second in the low byte. Date is packed as follows:

15 10	9 6	5 0
YEAR	MONTH	DATE

The next four blocks (1024) words contain the operator emitter to Digital

Generator slot map table. The table is indexed by operator emitter number and contains the corresponding slot number. The first scenario event is stored at the beginning of block number six.

There are six valid event types which may be used to describe platform motions and emitter characteristics. The first field of each event record contains the single word event type, which is an integer between one and six. The second field of each event record contains the single word event time expressed in seconds at which the event is to be executed.

All two word fields are 32 bit signed twoes complement integers, with the first word containing the most significant half and the second word the least significant half. All one word fields are 16 bit signed twoes complement integers. All byte and bit defined fields are unsigned integers or codes.

The two types of Platform Event Records are as follows:

1. Enter/Update New Platform.
2. Delete Platform (and Associated Emitters).

All calculations in the simulator assume a flat earth with an x,y, z coordinate system, where the +X direction is East, the +Y direction is North, and the +Z direction is up. The reference origin (0,0,0) is an arbitrary scenario center point. Headings are measured clockwise from true North.

Each platform must appear in the scenario before any of its associated emitters.

There are two methods of describing and updating platform motion, once a platform has been "entered". The Program method allows calculation of a new platform position each second using current position and velocities. This method requires a file event only for changes in velocity (X,Y or Z) and utilizes the velocity Change Event and Turn subtypes. The second method, scenario method, directly positions the platform to a specific point (X,Y,Z), thus not utilizing the velocity vectors. It is used in conjunction with the Platform Reposition Event subtype to approximate curved path motion, or to describe stationary sites.

The motion of any platform can be describd by either update method or a combination of the two methods. In general, the combination of methods is used to describe platform motion which exhibits both straight and curved line paths.

Unless operator specified, the observer platform is always platform number zero. Platform zero has default position and velocity parameters of zero and is the only platform with default parameters. Following Event Type and Time, Platform Number occupies a byte length field in Word 2 of all platform events. Platform Numbers, therefore, must not exceed 255.

ATEWES Enter/Update Platform Event Format

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

	Event Type = 1		
	Event Time		
	Event Length		
1		Platform Number	(ALL)
2	Heading (LSB = 360/32768 Degrees)		(ALL)
3 4	X Position (LSB = .125 Meters)		(N,R)
5 6	Y Position (LSB = .125 Meters)		(N,R)
7 8	Z Position (LSB = .015625 Meters)		(N,R)
9	X Velocity (LSB = .125 Meters/Second)		(N,V)
10	Y Velocity (LSB = .125 Meters/Second)		(N,V)
11	Z Velocity (LSB = .015625 Meters/Second)		(N,V)
12	Event Type Flags		(ALL)
13	Turn Rate (LSB = 360/32768 Degrees/Second)		(T)
14	Turn Duration (Seconds)		(T)
15	Platform ID Characters 1-3 (Radix 50)		(N)
16	Platform ID Characters 4-6 (Radix 50)		(N)
17	Speed (LSB = .125 Meters/Second)		(N,V,T)
18	Roll Angle (LSB = 360/1024 Degrees)		
19	Pitch Angle (LSB = 360/1024 Degrees)		

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

N = New; R = Reposition; V = Velocity Change; T = Turn

Word 9 - Heading

Heading is measured clockwise from true North in the units shown (360 degrees/1024)

A.2 Delete Platform (and Associated Emitters)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT TYPE = 2															
TIME IN SECONDS															
0								PLATFORM NUMBER							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The four types of Emitter Event Records are as follows:

1. Enter/Update Emitter
2. Delete Emitter
3. Emitter On
4. Emitter Off

All emitters to be simulated are entered at their first turn-on time. Each Enter/Update Emitter Event Record contains the emitter number, associated platform number and various descriptive parameters. The Delete Emitter Record removes the emitter and all its parameters from the simulator operating files. The Emitter Off Event Record places the emitter in an inactive status, allowing an Emitter On Event to restore activity without re-sending all parameters.

The length of the Enter/Update Emitter Event Record is variable dependent upon the number of repeated fields associated with the emitter. The other emitter event records all are fixed length format.

The Enter/Update Emitter event can be used to introduce emitters to be generated by special and regular sources within the simulator. The format shown defines all regular emitter parameters.

Enter/Update Emitter Event Format

<table>
<tr><th></th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>Flag</th></tr>
<tr><td></td><td colspan="16">Event Type = 3</td><td></td></tr>
<tr><td></td><td colspan="16">Event Time</td><td></td></tr>
<tr><td></td><td colspan="16">Event Length</td><td></td></tr>
<tr><td>1</td><td colspan="6"></td><td colspan="10">Emitter Number</td><td></td></tr>
<tr><td>2</td><td colspan="16">Spare</td><td></td></tr>
<tr><td>3</td><td colspan="8">Emitter Type</td><td colspan="8">Platform Number</td><td></td></tr>
<tr><td>4</td><td colspan="6"></td><td colspan="10">Azimuth Scan Type</td><td></td></tr>
<tr><td>5</td><td colspan="6"></td><td colspan="10">Azimuth Lower Sector Limit</td><td></td></tr>
<tr><td>6</td><td>C/D</td><td colspan="5"></td><td colspan="10">Azimuth Sector Width</td><td></td></tr>
<tr><td>7</td><td colspan="16">Azimuth Scan Rate (IPU)</td><td></td></tr>
<tr><td>8</td><td colspan="6"></td><td colspan="10">Azimuth Scan Master Emitter</td><td></td></tr>
<tr><td>9</td><td colspan="6"></td><td colspan="10">Elevation Scan Type</td><td></td></tr>
<tr><td>10</td><td colspan="6"></td><td colspan="10">Elevation Lower Sector Limit</td><td>Bit 8</td></tr>
<tr><td>11</td><td>C/D</td><td colspan="5"></td><td colspan="10">Elevation Sector Width</td><td></td></tr>
<tr><td>12</td><td colspan="16">Elevation Scan Rate (IPU)</td><td></td></tr>
<tr><td>13</td><td colspan="6"></td><td colspan="10">Elevation Scan Master Emitter</td><td></td></tr>
<tr><td>14</td><td colspan="2"></td><td>TTW</td><td>SPS</td><td>TOE</td><td>TOA</td><td>TE</td><td>TA</td><td colspan="8">Track Platform</td><td></td></tr>
<tr><td>15</td><td colspan="8">Missing Pulse Factor</td><td colspan="2"></td><td colspan="3">Rel El Jit</td><td colspan="3">Rel Brg Jit</td><td></td></tr>
<tr><td>16</td><td colspan="8">Sensitivity Override</td><td colspan="8">Effective Radiated Power</td><td>Bit 7</td></tr>
<tr><td>17</td><td colspan="6"></td><td colspan="10">Pulse Width</td><td>Bit 5</td></tr>
<tr><td>18</td><td colspan="3">User Flags</td><td colspan="5">Fine Pulse Pos</td><td></td><td>AWI</td><td>EWI</td><td>PWI</td><td>FWI</td><td>Dis</td><td>Syn</td><td>On</td><td>Bit 4</td></tr>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
</table>

<table>
<tr><th></th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>Flag</th></tr>
<tr><td>19</td><td colspan="6"></td><td colspan="10">Frequency 10 MSB</td><td>Bit 0</td></tr>
<tr><td>20</td><td>DED</td><td colspan="5"></td><td colspan="10">Frequency 10 LSB</td><td>Bit 0</td></tr>
<tr><td>21</td><td colspan="6">RF Agility Code</td><td colspan="10">RF Master Emitter</td><td></td></tr>
<tr><td>22</td><td colspan="3"></td><td colspan="13">RF Agility Limit</td><td></td></tr>
<tr><td>23</td><td colspan="4">Ch Numb</td><td>FM</td><td>U/D</td><td colspan="10">Chirp Rate</td><td></td></tr>
<tr><td>24</td><td colspan="16">RF Pattern Rate</td><td>Bit 1</td></tr>
<tr><td>25</td><td>T/P</td><td>C/D</td><td>U/D</td><td colspan="3">Pat Typ</td><td colspan="10">RF Pattern Multiplier</td><td>Bit 1</td></tr>
<tr><td>26</td><td colspan="4">Neg. Long Count</td><td colspan="12">PRI</td><td>Bit 2</td></tr>
<tr><td>27</td><td colspan="5">PRI Jitter Code</td><td colspan="11">PRI Jitter Limit</td><td></td></tr>
<tr><td>28</td><td colspan="16">PRI Pattern Rate</td><td>Bit 3</td></tr>
<tr><td>29</td><td>T/P</td><td>C/D</td><td>U/D</td><td colspan="3">Pat Typ</td><td colspan="10">PRI Pattern Multiplier</td><td>Bit 3</td></tr>
<tr><td>30</td><td>UTO</td><td colspan="15">Spare</td><td></td></tr>
<tr><td>31</td><td colspan="6">Link Jitter Code</td><td colspan="10">Repeat Count</td><td>Bit 6</td></tr>
<tr><td>32</td><td>LPE</td><td>Pri</td><td colspan="2">ELS</td><td colspan="2">LJM</td><td colspan="10">Link Seed Jitter Limit</td><td>Bit 6</td></tr>
<tr><td>33</td><td colspan="8"></td><td colspan="8">Number of Slots</td><td></td></tr>
<tr><td>34</td><td colspan="6"></td><td colspan="10">Associated Emitter Number</td><td>Bit 9</td></tr>
<tr><td>35</td><td colspan="5">ID Char 1</td><td colspan="5">ID Char 2</td><td colspan="6">ID Char 3</td><td></td></tr>
<tr><td>36</td><td colspan="5">ID Char 4</td><td colspan="5">ID Char 5</td><td colspan="6">ID Char 6</td><td></td></tr>
<tr><td>37</td><td colspan="4">AZ Scan Type</td><td colspan="4">EL Scan Type</td><td colspan="4">RF Type</td><td colspan="4">PRI Type</td><td></td></tr>
<tr><td>38</td><td colspan="16">Bit Map for Repeated Fields</td><td></td></tr>
<tr><td>39</td><td colspan="16">Additional Data Defined by Bit Map</td><td></td></tr>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
</table>

Additional description for emitter event fields is contained in the following sections.

Word 1 - Emitter Number

Bits 0 - 9

This field contains the operator entered emitter number.

Word 2 - Spare

Word 3 - Emitter Type/Platform Number

Bits 0 - 7 Platform Number 0-255

No emitters are assigned to the observer platform.

Bits 8 - 15

The emitter type defines the emitter operation to be performed.

1 - Enter New Emitter
2 - Update Emitter
3 - Enter Dependant Emitter
4 - Update Dependant Emitter

Dependant emitters rely on a master emitter for PRI triggering.

Word 4 - Azimuth Scan Type

Bits 0 - 4 Azimuth Antenna Pattern Number

The Antenna Pattern Number is determined from the Standard Scan Code Tables. All omni-directional emitters are assigned a Code 15. All standard emitters are assigned a code from 0 to 7. The proper code is the one which specifies a Reference Beamwidth with a binary multiple closest to the actual beamwidth (3dB) of the emitter. The reference beamwidths and multiples are shown in the Standard Emitter Antenna Scan Table. Special Scan Patterns may be used to incorporate nonstandard antenna patterns.

Bits 4 - 6 Azimuth Beamwidth Factor

The closest antenna beamwidth from the Antenna Scan tables designate both the reference beamwidth and a multiplier. The code for the Beamwidth factor is derived from the Beamwidth Factor Code Table table.

Bit 7 Azimuth Directional Mode

0 - unidirectional
1 - bidirectional

Emitters with full circular scan coverage are designated as undirectional.

Bit 8 Azimuth Blank Flag

The blank flag specifies whether bidirectional emitters should radiate during the return pass. A value of zero allows radiation, a value of one inhibits radiation. For unidirectional or stationary emitters, or conical scanning emitters, the flag is clear.

Bit 9 External Azimuth Synchronization

0 - Scan period is continuous
1 - Scan period is discrete

Externally synced azimuth scans are reset to the lower sector boundary whenever the elevation scan sector limit is reached.

Word 5 Azimuth Lower Scan Sector Boundary

Bits 0 - 9

For sector scanning emitters, a lower boundary and sector width are established which define the azimuthal limits of the antenna mainbeam with respect to the heading of the carrying platform. The boundaries are measured clockwise from platform heading and are expressed in binary angular measurement with the following least significant bit (LSB) units.

LSB = (Multiplier) (360 degrees)/4096 increments

The multiplier (1,2,4,...,128) is the value referenced by the Beamwidth Factor (Word 4) as discussed previously.

Emitters with full circular scan coverage should have a lower boundary of 512 and an upper boundary of 1024, adjusted by the scan multiplier. Stationary emitters should have a scan sector width of zero.

Word 6 - Azimuth Sector Width/Discrete Flag

Bits 0-9 Sector Width

Sector width defines the size of the sector to be swept for scanning emitters. The format is identical to that for lower sector limit. For stationary emitters, the field is set to zero. For circular scanning emitters, the sector width is set to 1024, adjusted by the scan multiplier.

Bit 15 Discrete Flag

0 - Continuous
1 - Discrete

Discrete scanning emitters have their scan periods forced to be a multiple of the emitters PRI. When the sector limit is reached, the current position is reset to zero and the overflow is lost. For continuous emitters, the overflow is retained.

Word 7 - Scan Rate Index

Bits 0 - 15 Increments Per Microsecond (IPU)

The IPU field describes the rate at which scanning emitters move through their scan pattern. The value is the number of increments to move through the pattern for each pulse output. The value is an unsigned 16 bit number, with an LSB of 1/65536 IPU. The following algorithm is used to derive the field value.

IPU = (Sector Width) (Direction Code + 1)*(4096/360)*(65536) / (SRI * MUL)

where: SRI is scan rate interval (in USEC)
Multiplier is a function of beamwidth
Direction Code is 0 for unidirectional, 1 for bidirectional

Word 8 - Azimuth Scan Master Emitter

For emitters with scan rates dependant on other emitters, this field contains the operator emitter number of the emitter to provide scan rate control.

Word 9 - Elevation Scan Type

Bits 0 - 4 Elevation Antenna Pattern Number

The Antenna Pattern Number is determined from the Standard Scan Code Tables. All omni-directional emitters are assigned a Code 15. All standard emitters are assigned a code from 0 to 7. The proper code is the one which specifies a Reference Beamwidth with a binary multiple closest to the actual beamwidth (3dB) of the emitter. The reference beamwidths and multiples are shown in the Standard Emitter Antenna Scan Table. Special Scan Patterns may be used to incorporate nonstandard antenna patterns.

Bits 4 - 6 Elevation Beamwidth Factor

The closest antenna beamwidth from the Antenna Scan tables designate both the reference beamwidth and a multiplier. The code for the Beamwidth factor is derived from the Beamwidth Factor Code Table table.

Bit 7 Elevation Directional Mode

0 - unidirectional
1 - bidirectional

Emitters with full circular scan coverage are designated as undirectional.

Bit 8 Elevation Blank Flag

The blank flag specifies whether bidirectional emitters should radiate during the return pass. A value of zero allows radiation, a value of one inhibits radiation. For unidirectional or stationary emitters, or conical scanning emitters, the flag is clear.

Bit 9 External Elevation Synchronization

0 - Scan period is continuous
1 - Scan period is discrete

Externally synced Elevation scans are reset to the lower sector boundary whenever the azimuth scan sector limit is reached.

Word 10 - Elevation Lower Scan Sector Boundary

Bits 0 - 9

For sector scanning emitters, a lower boundary and sector width are established which define the Elevation limits of the antenna mainbeam with respect to the heading of the carrying platform. The boundaries are measured clockwise from platform heading and are expressed in binary angular measurement with the following least significant bit (LSB) units.

LSB = (Multiplier) (360 degrees)/4096 increments

The multiplier (1,2,4,...,128) is the value referenced by the Beamwidth Factor (Word 4) as discussed previously.

Emitters with full circular scan coverage should have a lower boundary of 512 and an upper boundary of 1024, adjusted by the scan multiplier. Stationary emitters should have a scan sector width of zero.

Word 11 - Elevation Sector Width/Discrete Flag

Bits 0-9 Sector Width

Sector width defines the size of the sector to be swept for scanning emitters. The format is identical to that for lower sector limit. For stationary emitters, the field is set to zero. For circular scanning emitters, the sector width is set to 1024, adjusted by the scan multiplier.

Bit 15 Discrete Flag

Discrete scanning (1) emitters have their scan periods forced to be a multiple of the emitters PRI. When the sector limit is reached, the current position is reset to zero and the overflow is lost. For continuous scanning (0) emitters, the overflow is retained.

Word 12 - Scan Rate Index

Bits 0 - 15 Increments Per Microsecond (IPU)

The IPU field describes the rate at which scanning emitters move through their scan pattern. The value is the number of increments to move through the pattern for each pulse output. The value is an unsigned 16 bit number, with an LSB of 1/65536 IPU. The following algorithm is used to derive the field value.

IPU = (Sector Width) (Direction Code + 1)*(4096/360)*(65536) / (SRI * MUL)

where: SRI is scan rate interval (in USEC)
Multiplier is a function of beamwidth
Direction Code is 0 for unidirectional, 1 for bidirectional

Word 13 - Elevation Scan Master Emitter

For emitters with scan rates dependant on other emitters, this field contains the operator emitter number of the emitter to provide scan rate control.

Word 14 - Track Platform/Track Flags/Video Flags

Bits 0-7 Track Platform

This field contains the platform number whih the emitter will automatically track in realtime. For non-tracking emitters, this is set to the platform number to which the emitter is linked. As the emitter is entered in realtime, the platform number is checked to see whether the emitter is tracking the EW system, based on the EW system platform number for the current simulation. Once per simulated second, a new scan lower limit is computed and output for each tracking emitter.

Bit 8 TA

This bit signifies that the track platform is to be tracked in Azimuth.

Bit 9 TE

This bit signifies that the track platform is to be tracked in elevation.

Bit 10 TOA

This bit signifies that the observer platform is to be tracked in azimuth.

Bit 11 TOE

This bit signifies that the observer platform is to be tracked in elevation.

Bit 12 SPS

This bit signifies that the emitter is to trigger the SPS video channel. SPS pulse width and offset are stored as additional fields.

Bit 13 TTWS

This bit signifies that the emitter is to trigger the TTWS video channel. TTWS pulse width and offset are stored as additional fields.

Word ~~14~~ 5 - Missing Pulse Factor/Rel El Jitter/Rel Brg Jitter

Bits 0 - 2 Relative Bearing Jitter

This field is used to designate the amount of random pulse-to-pulse variation in the relative bearing of an emitter. The code is a function of the limit of relative bearing jitter, assuming a uniform double sided distribution around the true bearing. The relative bearing jitter limit in BAMS should be used to derive the code by locating the closest value in the Random Number Code Table.

Bits 3 - 6 Relative Elevation Jitter

This field is used to designate the amount of random pulse-to-pulse variation in the relative elevation of an emitter. The code is a function of the limit of relative elevation jitter, assuming a uniform double sided distribution around the true elevation. The relative elevation jitter limit in BAMS should be used to derive the code by locating the closest value in the Random Number Code Table.

Bits 8 - 15 Missing Pulse Factor

This field contains a missing pulse factor which is used to randomly drop a percentage of the pulses which would normally be generated for an emitter. The value represents the percent of pulses to be intentionally dropped, where the least significant bit equals .390625 percent.

Word ~~15~~ 6 - Sensitivity Override/Effective Radiated Power

Bits 8 - 15 Sensitivity Override

This field contains an optional sensitivity value which is used to inhibit pulses with lower power levels. The sensitivity value is given in Db. The sensitivity value is subtracted from the maximum available power of the RF Generator to determine an attenuation threshold. Any pulses with an attenuation greater than the threshold value will be inhibited at RF generation. If a sensitivity of zero is specified, a default threshold is determined during real time processing of the event.

Bits 0 - 7 Radiated Power

The field defines the effective radiated mainbeam power of an emitter in milliwatts, expressed directly in dBm. The field should take on positive values only.

Word 17 - Pulse Width

Bits 0 - 9

The pulse width is the duration of each pulse expressed with an LSB weight of 0.1 usec. The ten bit field restricts the maximum value of 102.3 usec. CW emitters should have a pulse width of zero.

Word 18 - Fine Position/Flags

Bit 13 Flag 1
Bit 14 Flag 2
Bit 15 Flag 3

Bits 13-15 represent three independent user-definable flags, which are associated with each pulse for a given emitter. The usage of the flags is entirely dependent upon the system under test. Flag 1 is stored in bit 13, Flag 2 is stored in bit 14 and Flag 3 is stored in bit 15.

Bits 8 - 12 Fine Pulse Position

This field is used to specify a time displacement of the emitter pulses. It allows pulses to occur at times other than the even microsecond. The offset is measured in increments of 50 nanoseconds.

Bit 6 - Azimuth Scan Write Inhibit

Bit 5 - Elevation Scan Write Inhibit

Bit 4 - Pri Modulation Write Inhibit

Bit 3 - Frequency Modulation Write Inhibit

Bits 3-6 represent flags used to inhibit updating dynamic position for scan and periodic modulations. Emitters with these flags set will not advance the current scan or modulation pattern position for the master emitter. All associated scan and frequency signals, and dependant emitters use these flags to inbibit interferance with period generation.

Bit 2 - PRI Display

This flag is used to enable output of the PRI display for an emitter. When the flag is set, the pulse train may be ovserved at the video output of the Digital Generator.

Bit 1 - Scan Sync

This flag is used to enable generation of the scan period trigger. When the flag is set, a transition is generated for the AZ and EL sync outputs whenever the emitter reaches a scan sector limit. Scan sync is automatically removed from the previously selected emitter when a new emitter is specified.

Bit 0 - On

The On flag determines whether an emitter will generate pulses. Emitters may be entered in a non radiating state, and later turned on by an Emitter On event.

Word 19 - Frequency MSB

Bits 0 - 9

This field contains the ten most significant bits of frequency.

Word 20 - Frequency LSB/Dedicated Flag

Bits 0 - 9 Frequency LSB

This field contains the ten least significant bits of frequency. The field has a least significant bit of .125 Mhz. When combined with word 19, a twenty bit frequency word is created.

Bit 15 Dedicated Flag

For emitters with direct assign RF channels, this flag specifies that the emitter is to have dedicated usage of the channel. As the emitter is entered in realtime, the assigned channel is removed from the pool. The channel is returned to the pool when the emitter is deleted.

Word 21 - RF Master Emitter/RF Agility Code

Bits 11-16 RF Agility Code

This field is used to designate the maximum number of bits of frequency to incorporate random pulse-to-pulse frequency agility. The code is a function of the frequency deviation limit, assuming a uniform double-sided distribution around the base frequency. The frequency deviation should be translated to units of .125 Mhz and the code derived from the Random Number Code Table. The code for the smallest table value which meets or exceeds the desired frequency deviation should be selected.

Bits 0 - 9 RF Master Emitter

For emitters with associated frequency modulation, this field specified the operator emitter number of the master emitter. The periodic modulation will depend upon the master emitter for its timing.

Word 22 - RF Agility Limit

Bits 0 - 12

This field contains the exact amount of random pulse to pulse frequency agility to be applied to the frequency. The field uses a least significant bit of .125 Mhz, and is used to further qualify the RF Agility Code (Word 21)

Word 23 - Channel Number/FM/Chirp

Bits 12 - 15 Channel Number

This field conatins the RF channel to which the emitter is assigned. A code of zero signifies that the emitter is to be generated using the pooled channels as determined by the RF Management logic. A non-zero value forces all pulses for the emitter to be generated by the specified RF channel.

Bit 11 FM

This flag enables the sunthesizer FM noise input port whenever the emitter outputs a pulse.

Bit 10 Up/Down

This flag determines whether the specified chirp is to be applied in the positive (0) or negative (1) direction.

Bits 0-9 Chirp Rate

This field contains the rate at which chirp is to be applied to the RF pulse. The rate is copmuted by dividing the chirp limit by the pluse width, and has a least significant bit of 500/1023 Mhz/Usec.

Word 24 RF Pattern Rate

This field contains the rate at which to move through a periodic frequency modulation pattern. The field is calculated in one of two ways, depending upon the Time/Pulse flag (Word 25). For time based modulations, the field is referred to as IPU, has a least significant bit of 1/65536 increments/usec, and is calculated as follows.

IPU = (1024) * (65536) / (Period)
Where Period is in Usec

For pulse based modulations, the field is referred to as IPP, has a least significant bit of 1/256 increments/pulse, and is calculated as follows.

IPP = (1024) * (256) / (Period)
Where Period is in Usec

Word 25 - Modulation Flags/Pattern Multiplier

Bit 15 Time/Pulse

The Time/Pulse flag determines whether the periodic modulation is updated based on the number of pulses output, or the elapsed time. For time based (0) modulations, the IPU field is multiplied by the current PRI for each pulse to determine the amount to advance through the modulation pattern. For pulse based (1) modulations, the IPP field is used directly to advance through the modulation pattern.

Bit 14 Discrete Flag

Discrete modulation (1) emitters have their modulation periods forced to be a multiple of the emitters PRI. When the pattern limit is reached, the current position is reset to zero and the overflow is lost. For continuous modulation (0) emitters, the overflow is retained.

Bit 13 Up/Down

The Up/Down flag controls which direction the modulation moves through the pattern. For Up (0), the pattern begins at position zero, and advances to the upper limit. For Down (1), the pattern begins at the upper limit, and advances to position zero. This allows the Digital Generator to generate inverse modulation patterns using the same pattern type.

Bits 10 - 12 Pattern Type

This field selects the pattern type to be applied to a periodic modulation. Patterns are selected from the following types.

Code	Pattern Type
0	Ramp
1	Concave Increasing Exponential (e^{At})
2	Convex Increasing Exponential ($1-e^{At}$)
3	Unassigned
4	Unassigned
5	Unassigned
6	Trapezoidal
7	Sinusoidal

Bits 0 - 9 RF Pattern Multiplier

This field determines the amplitude of the periodic modulation. This represents the maximum excursion from the base value. The field has a least significant bit of 255/256 Mhz. For sinusoudal pattern types, the base value is set to the lower limit of the sine wave, and the modulation amplitude is set to twice the sinusoidal excursion.

Word 26 - Long Count/PRI

Bits 12 - 15

For emitters with PRI values exceeding 4095, two Digital Generator slots are allocated. This field specifies the number of times to repeat the dummy PRI value to generate the desired effective PRI. The value is stored as a negative number.

Bits 0 - 11 PRI

This field contains the base PRI for the emitter, with a least significant bit of 1 usec.

Word 27 - PRI Jitter Code/PRI Jitter Limit

Bits 11 - 15 PRI Jitter Code

This field is used to designate the maximum number of bits of PRI to incorporate random pulse-to-pulse PRI jitter. The code is a function of the jitter limit, assuming a uniform double-sided distribution around the base PRI. The PRI deviation in usec should be used to derive the code from the Random Number Code Table. The code for the smallest table value which meets or exceeds the desired PRI deviation should be selected.

Bits 0 - 10 PRI Jitter Limit

This field contains the exact amount of random pulse to pulse PRI jitter to be applied to the emitter. The field uses a least significant bit of 1 usec, and is used to further qualify the PRI Jitter Code.

Word 28 PRI Pattern Rate

This field contains the rate at which to move through a periodic PRI modulation pattern. The field is calculated in one of two ways, depending upon the Time/Pulse flag (Word 29). For time based modulations, the field is referred to as IPU, has a least significant bit of 1/65536 increments/usec, and is calculated as follows.

IPU = (1024) * (65536) / (Period)
Where Period is in Usec

For pulse based modulations, the field is referred to as IPP, has a least significant bit of 1/256 increments/pulse, and is calculated as follows.

IPP = (1024) * (256) / (Period)
Where Period is in Usec

Word 29 - Modulation Flags/Pattern Multiplier

Bit 15 Time/Pulse

The Time/Pulse flag determines whether the periodic modulation is updated based on the number of pulses output, or the elapsed time. For time based (0) modulations, the IPU field is multiplied by the current PRI for each pulse to determine the amount to advance through the modulation pattern. For pulse based (1) modulations, the IPP field is used directly to advance through the modulation pattern.

Bit 14 Discrete Flag

Discrete modulation (1) emitters have their modulation periods forced to be a multiple of the emitters PRI. When the pattern limit is reached, the current position is reset to zero and the overflow is lost. For continuous modulation (0) emitters, the overflow is retained.

Bit 13 Up/Down

The Up/Down flag controls which direction the modulation moves through the pattern. For Up (0), the pattern begins at position zero, and advances to the upper limit. For Down (1), the pattern begins at the upper limit, and advances to position zero. This allows the Digital Generator to generate inverse modulation patterns using the same pattern type.

Bits 10 - 12 Pattern Type

This field selects the pattern type to be applied to a periodic modulation. Patterns are selected from the following types.

Code	Pattern Type
0	Ramp
1	Concave Increasing Exponential (e^{At})
2	Convex Increasing Exponential ($1-e^{At}$)
3	Unassigned
4	Unassigned

5	Unassigned
6	Trapezoidal
7	Sinusoidal

Bits 0 - 9 PRI Pattern Multiplier

This field determines the amplitude of the periodic modulation. This represents the maximum excursion from the base value. The field has a least significant bit of 255/256 usec. For sinusoudal pattern types, the base value is set to the lower limit of the sine wave, and the modulation amplitude is set to twice the sinusoidal excursion.

Word 30 - Turn Off

Bit fifteen of word thirty is a flag which identifies an emitter which must be turned off in order to perform a mode change. Certain emitters, notably those using sync emitters to generate short PRI's and those using Link Code Jitter, must be turned off in order to perform an emitter update without creating dropped emitters.

Word 31 - Link Jitter Code/Repeat Count

Bits 10 - 15 Link Jitter Code

This field is used to designate the maximum number of bits of Link Seed/Jitter to incorporate random pulse-to-pulse jitter. The code is a function of the jitter limit, assuming a uniform positive only distribution around the base value. The deviation should be used to derive the code from the Random Number Code Table. The code for the smallest table value which meets or exceeds the desired deviation should be selected.

Bits 0 - 9 Repeat Count

This field contains the number of pulses to output before using the Pattern Switch Link emitter number. The number of pulses is derived from the desired dwell time and the PRI of the emitter.

Word 32 - Emitter Link Select/Link Jitter Mode/Link Seed Jitter Limit

Bit 15 Link Previous Emitter

This flag signifies that the normal link for this slot is to be the previous slot. For emitters with desired repeat counts exceeding the capability of the Digital Generator, two slots are allocated, with the second slot having a normal sequence link of the previous slot. The resulting repeat count is the product of the repeat counts of the two slots.

Bit 14 Priority

This field contains a priority flag for use by the Digital Generator in resolving emitter conflicts for pulse requests. An emitter with priority will bump other emitters from their desired time slots when more than four simulatenous pulses are attempted to be generated digitally. In addition, priority emitters will have first access to available RF channles when the pulses for a given microsecond are allocated to the available resources.

Bits 12 - 13 Emitter Link Select

This field controls the emitter linking method to be used, defined as follows.

0 = Add link code jitter to sequence link
1 = Use switch link on PRI pattern overflow
2 = Use switch link on RF pattern overflow
3 = Use switch link on repeat count underflow

Bits 10 - 11 Link Jitter Mode

This field controls the application of Link Seed Jitter, defined as follows.

0 = Link code jitter
1 = Repeat count jitter
2 = Seed PRI jitter random number

3 = Seed frequency agility random number

Bits 0 - 9 Link Seed Jitter Limit

This field specifies the amount of Link Seed Jitter to be generated. It is used to further qualify the Link Seed Jitter Code.

Word 33 - Number of Slots

Complex emitters require more than a single Digital Generator slot. This field specifies the number of slots required. When more than one slot is used, words 1 through 34 are used to format the parameters for the first slot. The bitmap in word 38 then specifies which fields are repeated for each additional slot. All other parameters remain the same for all slots.

Word 34 - Associated Emitter Number

This field contains the operator emitter number of an emitter to be generated as a dependant emitter. The specified emitter is included as a sync emitter for the emitter currently being entered.

Words 35/36 - Emitter ID

This field contains an emitter identification. The ID is stored as up to six ASCII characters, packed in Radix 50 notation as defined by Digital Equipment Corporation.

Word 37 - Signal Types

Bits 12 - 15 Azimuth Scan Type

This field defines the azimuth scan type as follows.

Code	Type
1	Circular

2	Sector
3	Raster
4	Conical
5	Helical
6	Steady
7	OMNI
8	Associated

Bits 8 - 11 Elevation Scan Type

This field defines the elevation scan type as follows.

Code	Type
1	Sector
2	Steady
3	OMNI
4	Associated

Bits 4 - 7 RF Signal Type

This field defines the frequency signal type as follows.

Code	Type
1	Continuous Wave
2	Single Frequency
3	Sequence
4	Periodic Modulation
5	Switching
6	Discrete Agility
7	Multibeam
8	Associated

Bits 4 - 7 RF Signal Type

This field defines the PRI signal type as follows.

Code	Type
1	Single PRI
2	Stagger, Pulse Group, Sequence
3	Periodic Modulation
4	Switching
5	Discrete Jitter

Word 38 - Bitmap

Word thirty eight defines the fields which are repeated for emitters requiring additional Digital Generator slots. Where multiple bits are set, fields are repeated in order of increasing bit number. For example, if bits 0 and 1 are both set, fields repeated for bit 0 are stored before fields repeated for bit 1. Complete sets of repeated data are stored, with each set fully defining the next Digital Generator slot. Repeated fields follow exactly the same format as corresponding fields storen in the base event. Bit values are defined as follows.

Bit 0 - Frequency list (Words 19/20)
Bit 1 - Frequency modulation list Words (24/25)
Bit 2 - PRI list (Word 26)
Bit 3 - PRI modulation list (Words 28/29)
Bit 4 - Fine position list (Word 18)
Bit 5 - Pulse width list (Word 17)
Bit 6 - Repeat count list (Words 31/32)
Bit 7 - ERP list (Word 16)
Bit 8 - Elevation start angle list (Word 10)
Bit 9 - Associated emitter list (Word 34)

A.3.1 Antenna Pattern Code Table

Code	Description
0	Standard Reference Beamwidth = .70
1	" " " = .77
2	" " " = .84
3	" " " = .92
4	" " " = 1.00
5	" " " = 1.09
6	" " " = 1.20
7	" " " = 1.31
8	Conical Modulation
9	Special Scan Pattern
10	" " "
11	" " "
12	" " "
13	" " "
14	" " "
15	Omni-directional

A.3.2 Standard Emitter Antenna Scan Table

Multiplier	Antenna Pattern Code							
	0	1	2	3	4	5	6	7
1	.70	.77	.84	.92	1.00	1.09	1.20	1.31
2	1.41	1.54	1.68	1.83	2.00	2.19	2.39	2.61
4	2.81	3.07	3.36	3.67	4.01	4.38	4.78	5.22
8	5.63	6.15	6.71	7.33	8.01	8.75	9.56	10.45
16	11.25	12.29	13.43	14.67	16.03	17.51	19.13	20.90
32	22.50	24.58	26.86	29.34	32.05	35.02	38.26	41.80
64	45.00	49.16	53.71	58.68	64.11	70.04	76.51	83.59
128	90.00	98.33	107.42	117.36	128.21	140.07	153.03	167.18

A.3.3 Beamwidth Factor Code Table

Code	Multiplier
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

A.3.4 Random Number Code Table

Code	Maximum Random Number
0	0
1	1
2	3
3	7
4	15
5	31
6	63
7	127
8	255
9	511
10	1023
11	2047
12	4095
13	8191

A.4 Delete Emitter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT TYPE = 4															
TIME IN SECONDS															
EMITTER NUMBER															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

A.5 Emitter Off

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT TYPE = 5															
TIME IN SECONDS															
EMITTER NUMBER															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

A.6 Emitter On

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT TYPE = 6															
TIME IN SECONDS															
EMITTER NUMBER															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

APPENDIX B

DATA EXTRACTION FILE DESCRIPTION

The Data Extraction file is made up of a series of variable length records each describing a single scenario event, operator command or status event. Data is recorded in the order of command or event execution. The records are packed into 256 word blocks, which are written sequentially to a FILES-11 format file in 256 word records. Events or commands are allowed to cross block boundaries when necessary. Any extra words in the last block are filled with zeroes when the Data Extraction File is closed. Data analysis files are read and processed offline as required for performance evaluation.

The Data Extraction file contains all scenario events, operator commands, and status data pertinent to a simulation run. The first field of each record is the event or command code, as described below. The second field of each record is the actual execution time in clock ticks, defined in the DX header.

The third field of each record contains the simulated time in seconds at which the event occurred. The fourth field contains the total number of words in the record.

The first block (256 words) of the DX file contains the file header. The header contains the following data.

WORD	CONTENTS
1	Tick size > 0 milliseconds/tick < 0 ticks/second
2	File Type = 700 + Revision Level
43	Simulation Run Date
44	Simulation Run Time (minutes)
45-51	Scenario File Name
52-58	Maximum Power File Name
59-65	Receiver Gain File Name
66-72	Sensitivity File Name
73	EW System Platform Number
74	Test Mode 0 = Fixed Antenna/Digital Output 1 = Rotating Antenna/Digital Output 2 = Fixed Antenna/RF Output 3 = Rotating Antenna/RF Output
158-164	Azimuth Receiver Pattern Code File Name
237-243	Elevation Receiver Pattern Code File Name

All fields not described are reserved for use by future programs. All ASCII fields are stored with the first character in the high byte and the second in the low byte.

Date is packed as follows:

15 10	9 6	5 0
YEAR	MONTH	DATE

CODE	EVENT OR COMMAND
1	Enter/Update Platform
2	Delete Platform
3	Enter/Update Emitter
4	Delete Emitter
5	Emitter Off
6	Emitter On
30	Dropped Emitter
31	Pulse Count Request Time
32	Record Pulse Counts
33	Record Platform Update Information
34	Record Scan Update Information
64	Start Scenario
65	Stop Scenario
66	Change Scenario Speed
67	Radiation Off
68	Radiation On
69	Advance Scenario
70	Backup Scenario
71	Disable All DX Recording
72	End Simultaion Run
95	Maximum Power File
96	Receiver Gain File
97	Receiver Sensitivity File
98	Azimuth Receiver Pattern File
99	Elevation Receiver Pattern File

B.1 Codes 1-6

Event codes one through six correspond to the six event types specified in the scenario file. The parameters for these events, words 2 - N from the scenario are copied to the DX file following the length parameter as described above. Data for operator events are stored in the same fashion, except that the event number is negated to signify an operator event.

B.2 Code 30 - Dropped Emitter

WORD	
0	EVENT TYPE = 30
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 7
5	EMITTER NUMBER *
6	DIGITAL GENERATOR SLOT NUMBER

* The Sign bit of word 5 reflects the status of the Emitter on flag in the Emitter File at the time the emitter was dropped.

B.3 Code 31 - Pulse Count Request Time Event

WORD	
0	EVENT TYPE = 31
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 7
5	REQUEST TIME
6	

B.4 Code 32 - Pulse Counts

WORD	
0	EVENT TYPE = 32
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH
5	FIRST OPERATOR EMITTER NUMBER
6	TOTAL PULSES OUTPUT
7	TOTAL AMPLITUDE INHIBITED
8	TOTAL RF DROPPED
9	NEXT OPERATOR EMITTER NUMBER
10	TOTAL PULSES OUTPUT
11	TOTAL AMPLITUDE INHIBITED
12	TOTAL RF DROPPED

Pulse counts for up to 64 emitters are packed in a single pulse count event. If more emitters are to be recorded, additional events are output.

B.5 Code 33 - Record Platform Update

WORD		
0	EVENT TYPE = 33	
1	TICK COUNT	
2		
3	SIMULATED TIME	
4	RECORD LENGTH	
5	RANGE ATTENUATION	1ST PLATFORM #
6	REL BEARING FROM EW SYSTEM	
7	REL ELEVATION FROM EW SYSTEM	
8	RANGE ATTENUATION	2ND PLATFORM #
9	REL BEARING FROM EW SYSTEM	
10	REL ELEVATION FROM EW SYSTEM	
N	RANGE ATTENUATION	NTH PLATFORM #
N+1	REL BEARING FROM EW SYSTEM	
N+2	REL ELEVATION FROM EW SYSTEM	

B.6 Code 34 - Record Scan Update

WORD	
0	EVENT TYPE = 34
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH
5	1st EMITTER NUMBER
6	LOWER SECTOR BOUNDARY
7	2nd EMITTER NUMBER
8	LOWER SECTOR BOUNDARY
9	3rd EMITTER NUMBER
10	LOWER SECTOR BOUNDARY
11	4th EMITTER NUMBER
12	LOWER SECTOR BOUNDARY
13	5th EMITTER NUMBER
14	LOWER SECTOR BOUNDARY
N	Nth EMITTER NUMBER
N+1	LOWER SECTOR BOUNDARY

Lower Sector Boundary is in units of 360/1024 degrees.
Bit 15 of each sector bpoundary is set for elevation limit, clear for azimuth limit.

B.7 Code 64 - Start Scenario

WORD	
0	EVENT TYPE = 64
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 6
5	STOP TIME

B.8 Code 65 - Stop Scenario

WORD	
0	EVENT TYPE = 65
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 5

B.9 Code 66 - Set Scenario Speed

WORD	
0	EVENT TYPE = 66
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 6
5	SPEED CODE (POWER OF TWO)

B.10 Code 67 - Radiation Off

WORD	
0	EVENT TYPE = 67
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 5

B.11 Code 68 - Radiation On

WORD	
0	EVENT TYPE = 68
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 5

B.12 Code 69 - Advance Scenario

WORD	
0	EVENT TYPE = 69
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 6
5	ADVANCE TARGET TIME

B.13 Code 70 - Backup Scenario

WORD	
0	EVENT TYPE = 70
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 6
5	BACKUP TARGET TIME

B.14 Code 71 - Disable All DX Recording

WORD	
0	EVENT TYPE = 71
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 5

B.15 Code 72 - End Simulation Run

WORD	
0	EVENT TYPE = 72
1	TICK COUNT
2	
3	SIMULATED TIME
4	RECORD LENGTH = 5

B.16 Code 95 - Maximum Power File

WORD	
0	EVENT TYPE = 95
1 2	TICK COUNT
3	SIMULATED TIME
4	EVENT LENGTH = 256
5	MAX POWER FIRST FREQUENCY
6	MAX POWER SECOND FREQUENCY
	⋮
157	MAX POWER LAST FREQUENCY
158	0
	⋮
255	0

This event is used to record an image of the Maximum Power file used at initialization for the current run. The Max Power file data are copied in order to words 5 through 158 of the event. The remainder of the event is zero filled to have the event occupy one disc block. The max power values are stored as integer data, with an LSB of 1/128 Db.

B.17 Code 96 - Receiver Gain File

WORD	
0	EVENT TYPE = 96
1 2	TICK COUNT
3	SIMULATED TIME
4	EVENT LENGTH = 256
5	RECEIVER GAIN FIRST FREQUENCY
6	RECEIVER GAIN SECOND FREQUENCY
	⋮
157	RECEIVER GAIN LAST FREQUENCY
158	0
	⋮
255	0

This event is used to record an image of the Receiver Gain file used at initialization for the current run. The Receiver Gain file data are copied in order to words 5 through 158 of the event. The remainder of the event is zero filled to have the event occupy one disc block. The max power values are stored as integer data, with an LSB of 1/128 Db.

B.18 Code 97 - Receiver Sensitivity File

WORD	
0	EVENT TYPE = 97
1	TICK COUNT
2	
3	SIMULATED TIME
4	EVENT LENGTH = 256
5	RECEIVER SENSITIVITY FIRST FREQUENCY
6	RECEIVER SENSITIVITY SECOND FREQUENCY
	⋮
157	RECEIVER SENSITIVITY LAST FREQUENCY
158	0
	⋮
255	0

This event is used to record an image of the Receiver Sensitivity file used at initialization for the current run. The Receiver Sensitivity file data are copied in order to words 5 through 158 of the event. The remainder of the event is zero filled to have the event occupy one disc block. The max power values are stored as integer data, with an LSB of 1/128 Db.

B.19 Code 98 - Azimuth Receiver Pattern File

WORD	
0	EVENT TYPE = 98
1 2	TICK COUNT
3	SIMULATED TIME
4	EVENT LENGTH = 256
5	AZIMUTH PATTERN CODE FIRST FREQUENCY
6	AZIMUTH PATTERN CODE SECOND FREQUENCY
	• • •
157	AZIMUTH PATTERN CODE LAST FREQUENCY
158	0
	• • •
255	0

This event is used to record an image of the azimuth receiver codes used at initialization for the current run. The receiver pattern codes are copied in order to words 5 through 158 of the event. The remainder of the event is zero filled to have the event occupy one disc block.

B.20 Code 99 - Elevation Receiver Pattern File

WORD	
0	EVENT TYPE = 99
1 2	TICK COUNT
3	SIMULATED TIME
4	EVENT LENGTH = 256
5	ELEVATION PATTERN CODE FIRST FREQUENCY
6	ELEVATION PATTERN CODE SECOND FREQUENCY
	⋮
157	ELEVATION PATTERN CODE LAST FREQUENCY
158	0
–	⋮
255	0

This event is used to record an image of the elevation receiver codes used at initialization for the current run. The receiver pattern codes are copied in order to words 5 through 158 of the event. The remainder of the event is zero filled to have the event occupy one disc block.

APPENDIX C

DIGITAL SUBSYSTEM INTERFACE

The interface between the Digital Generator and the Control Subsystem consists of three 16 bit paths plus associated status lines. The three paths are Command Data, Status Data and Pulse Count Data.

C.1 Command Data

The Command Data path consists of 16 data lines from the Control Subsystem to the Digital Generator plus a control line. The interface is terminated in the Control Subsystem by a DR-11M interface card supplied by DEC. The sixteen data lines of port B are used, and a Unibus address of 764020 is selected. The DATA READY OUT B signal is asserted whenever new data is loaded into the output register. All signals are TTL compatible levels.

Commands are output at a maximum rate of one every 12 microseconds. No acknowledgment is required from the Digital Generator.

The data path is divided into two fields: a Command Field in the most significant six bits and a Value Field in the ten least significant bits. When all of the Value bits are not used, the field is right justified and the unused bits are undefined.

When an emitter number is loaded, all subsequent commands reference this emitter until a new emitter number is loaded. All angles (beaings and headings) are ten bit fields in units of 360/1024 Degrees, referred to as Binary Angular Measure (BAMs), divided by the Scan Multiplier. A list of valid commands and Value fields follows.

COMMAND NAME	DATA FIELD	COMMAND CODE
LFRQJC	Load Frequency Agility Limit MSB(3),Frequency Agility Code(4)	40
LCHIRP	Load Chirp Rate	41
LFCHAN	Load Frequency Chirp Up/Down(1),External FM Enable(1), RF Channel Code(4)	42
LFIPUL	Load Frequency Pattern Rate (IPU) LSB	43
LFIPUM	Load Frequency Pattern Rate (IPB) MSB	44
LFRQTP	Load Frequency Pattern Type(3), Up/Down Count(1), Continuous/Discrete(1),Time/Pulse(1)	45
LFRQML	Load Frequency Pattern Multiplier	46
LFRQEN	Load Frequency Base Emitter Number	47
LPUWID	Load Pulse Width	48
LFNPOS	Load Fine Pulse Position(5),User Flags(3)	49
LATTEN	Load Base Attenuation	50
LTHRSH	Load Threshold	51
LMPFAC	Load Missing Pulse Factor	52

C.1.2 System Commands

COMMAND NAME	DATA FIELD	COMMAND CODE
LOWNHD	Load Observer Heading	33
LCLIMB	Load Observer Climb Angle	36
LSYCON	Load RF BIT DOA Select(3),Rotating Receiver Enable(1), RF Mode Enable(1), RF Enable(1)	53
LRFPOL	Load RF Channel Pool Selection	54
LRESET	Reset Emitter Active Flags(1), Reset Digital/RF Generators	55
LMINHR	Load Scenario Time BCD Minutes(7),Hours(3)	56
LSECND	Load Scenario Time BCD Seconds	57
RPCNTS	Read Emitter Pulse Counts	62

C.2 Status Data

The Status Data path consists of sixteen data lines from the Digital Subsystem to the Control Subsystem. The sixteen lines are driven by a sixteen bit status register which always contains the data selected by the last Read command received from the Control Subsystem. The information is right justified with unused bits set to zero. Data is selected by command code 63, and is identical in format to data in the value fields of the corresponding Load commands (0-62). Other data formats are described in Sections C.2.1 and C.2.2.

The interface is terminated in the Control Subsystem by a DR-11L interface card supplied by DEC. The sixteen data lines of port A are used, and a Unibus address of 764030 is selected. The 16 bit status register is updated by the

Digital Subsystem within 12 microseconds of receipt of a Read command. All levels are TTL compatible.

C.2.1 Dropped Emitter Number

When the Digital Subsystem discovers that it has gone .1 seconds without generating a pulse for an active emitter, it saves the emitter number to be read by the Control Subsystem. When the Control Subsystem issues a Read Dropped Emitter Number Command, the Digital Generator will respond by moving the ten bit emitter number to the Status Register.

C.3 Pulse Counter

The Digital Generator maintains three sixteen bit counter for each emitter number. The counters are incremented each time a pulse is generated or dropped. Upon receipt of an Initiate Transfer of Pulse Counts command (Command 55), the Digital Generator transfers all of the pulse counters to the Control Subsystem via the Pulse Counts data path which is a Direct Memory Access channel into the main memory of the Control Subsystem. Each counter is cleared after it is transferred. The Digital Generator will prevent any of the counters from counting past the maximum number of 65536.

The Direct memory Access channel is implemented by a KIT11-D interface system supplied by DEC, using a Unibus address of 764000. The DATA AVAILABLE IN signal is asserted each time a data word is to be transferred. All levels are TTL compatible.

APPENDIX D

TEWES - AISF NAVIGATION MESSAGE

The TEWES Navigation message is sent periodically using a standard RS 232C serial communications line at 9600 baud. Message transmission begins when the first Start Scenario command is issued at the TEWES operator's console. The AISF computer shall transmit a one byte request message to the TEWES computer each time a navigation message is desired. The TEWES shall immediately transmit the latest navigation data. Transmission continues at the normal rate when freeze mode is entered, with static parameters output for the EW system. Pulse count data continues to be updated during freeze mode. Data shall be sent as 8-bit binary codes, with no parity bits and one stop bit. Within each data byte, the LSB of the byte corresponds to the LSB of the specific field. Where the field contains less than 8 bits, the field is right justified within the transmitted byte. The message shall contain the IRIG-B time code read as transfer of the data initiated, followed by the current TEWES scenario time. This is followed by the current position, heading, latitude, longitude, pitch and roll angles for the EW system. The scenario speed power of two code, scenario and RF status, and pulse count data complete the message. Pulse count data includes the totals of pulses generated, pulses amplitude inhibited, and pulses inhibited due to RF contention. Totals include all active emitters except CW signals, and represent the latest data read from the TEWES DGU. All positions, latitude, longitude and scenario speed are sent in 2's complement integer format. All other fields are unsigned integer codes. The final byte of the message is a checksum calculated by generating the exclusive "or" of all data bytes within the message. Detailed formats for the entire message are defined as follows.

TEWES - AISF

NAVIGATION MESSAGE

BYTE FORMAT

BYTE	CONTENTS			LSB
0	IRIG-B	TIME	BITS 16-23	
1	IRIG-B	TIME	BITS 24-31	
2	IRIG-B	TIME	BITS 0-7	1 MSEC
3	IRIG-B	TIME	BITS 8-15	
4	SCENARIO	TIME	BITS 0-7	1 SEC
5	SCENARIO	TIME	BITS 8-15	
6	EW SYSTEM	X POSITION	BITS 16-23	
7	EW SYSTEM	X POSITION	BITS 24-31	
8	EW SYSTEM	X POSITION	BITS 0-7	1/8 METER
9	EW SYSTEM	X POSITION	BITS 8-15	
10	EW SYSTEM	Y POSITION	BITS 16-23	
11	EW SYSTEM	Y POSITION	BITS 24-31	
12	EW SYSTEM	Y POSITION	BITS 0-7	1/8 METER
13	EW SYSTEM	Y POSITION	BITS 8-15	
14	EW SYSTEM	Z POSITION	BITS 16-23	
15	EW SYSTEM	Z POSITION	BITS 24-31	
16	EW SYSTEM	Z POSITION	BITS 0-7	1/64 METER
17	EW SYSTEM	Z POSITION	BITS 8-15	
18	EW SYSTEM	TRUE HEADING	BITS 0-7	360/1024 DEG
19	EW SYSTEM	TRUE HEADING	BITS 8-9	
20	EW SYSTEM	MAGNETIC HEADING	BITS 0-7	360/1024 DEG
21	EW SYSTEM	MAGNETIC HEADING	BITS 8-9	
22	EW SYSTEM	LATITUDE	BITS 0-7	360/65536 DEG
23	EW SYSTEM	LATITUDE	BITS 8-15	
24	EW SYSTEM	LONGITUDE	BITS 0-7	360/65536 DEG
25	EW SYSTEM	LONGITUDE	BITS 8-15	
26	EW SYSTEM	PITCH	BITS 0-7	360/1024 DEG
27	EW SYSTEM	PITCH	BITS 8-9	
28	EW SYSTEM	ROLL	BITS 0-7	360/1024 DEG
29	EW SYSTEM	ROLL	BITS 8-9	

BYTE	CONTENTS		LSB
30	NUMBER OF ACTIVE EMITTERS	BITS 0-7	
31	NUMBER OF ACTIVE EMITTERS	BITS 8-15	
32	LATEST TOTAL PULSES GENERATED	BITS 16-23	
33	LATEST TOTAL PULSES GENERATED	BITS 24-31	
34	LATEST TOTAL PULSES GENERATED	BITS 0-7	
35	LATEST TOTAL PULSES GENERATED	BITS 8-15	
36	LATEST TOTAL PULSES AMPLITUDE INHIBIT	BITS 16-23	
37	LATEST TOTAL PULSES AMPLITUDE INHIBIT	BITS 24-31	
38	LATEST TOTAL PULSES AMPLITUDE INHIBIT	BITS 0-7	
39	LATEST TOTAL PULSES AMPLITUDE INHIBIT	BITS 8-15	
40	LATEST TOTAL PULSES RF INHIBIT	BITS 16-23	
41	LATEST TOTAL PULSES RF INHIBIT	BITS 24-31	
42	LATEST TOTAL PULSES RF INHIBIT	BITS 0-7	
43	LATEST TOTAL PULSES RF INHIBIT	BITS 8-15	
44	LATEST TOTAL PULSES RF INHIBIT	BITS 0-7	
45	SCENARIO SPEED CODE	BIT 0-7	$-3 \leq$ CODE ≤ 3
46	RADIATION STATUS	BIT 0	0=OFF, 1=ON
47	SCENARIO STATUS	BIT 0	0=STOP, 1=RUN
48	EW SYSTEM PLATFORM NUMBER	BITS 0-7	
49	CHECKSUM	BITS 0-7	

APPENDIX E

TYPICAL SECOND TIMING DIAGRAM
NORMAL SCENARIO SPEED

T = X	Update Simulated Time Output to Observer Process Scenario Events at Current Time
T = X+.5	Set Update Flag Output to Observer
T = X+1	Update Simulated Time Output to Observer Process Scenario Events at Current Time

Decrement Pulse Counter Flag Once per Tick

If result zero and previous requested data has been processed, request new data and reinitialize flag. If previous data has not been processed, leave Pulse Counter Flag set to zero.