

APOLLO GUIDANCE COMPUTER

Information Series

ISSUE 17

KEYRUPT, UPRUPT, MARK  
AND DSKY FUNCTIONS

FR-2-117

14 August 1964

Changed 20 April 1965

## CONTENTS (cont)

Paragraph		Page
17-225	The DSPLIST Waitinglist . . . . .	17-93
17-228	Decoding and Executing the Verb-Noun Combination . . . . .	17-95
17-240	Executing a Non-CHARIN Load Request . .	17-100
17-245	Subsection ENTER . . . . .	17-101
17-248	Organization . . . . .	17-102
17-251	Initiating Group . . . . .	17-102
17-255	Continuing Group . . . . .	17-102
17-260	Initiating the Execution of a Verb-Noun Combination . . . . .	17-106
17-261	Routine ENTER . . . . .	17-106
17-263	Routine ENTPASO . . . . .	17-106
17-273	Decoding the Verb Code . . . . .	17-111
17-277	Entering a Data or Address Word . . .	17-116
17-279	Subsection REGULAR VERBS . . . . .	17-117
17-283	Display Data in Specified Locations . .	17-118
17-285	Octal Display Verbs . . . . .	17-119
17-288	Routines DSPA, DSPB, DSPC, DSPAB, and DSPABC . . . . .	17-119
17-302	Routine DSPBANK . . . . .	17-123
17-305	Decimal Display Verbs . . . . .	17-124
17-307	Sealing . . . . .	17-124
17-311	Routine DECDSP . . . . .	17-130
17-343	Routine DSDPDEC . . . . .	17-148
17-346	Shift Display Verb . . . . .	17-148
17-347	Functional Description . . . . .	17-148
17-350	Routine BUMP . . . . .	17-149
17-354	Monitor Data Contained in Specified Locations . . . . .	17-150
17-355	The Monitor Function . . . . .	17-150
17-357	Organization of the Monitor Function .	17-151
17-359	Routine MONITOR . . . . .	17-152
17-362	Task MONREQ . . . . .	17-153
17-364	Job MONDO . . . . .	17-153
17-368	Request Data Entry Into Specified Locations . . . . .	17-155
17-378	Requesting Routines . . . . .	17-159
17-379	Routine ALOAD . . . . .	17-159
17-381	Routine BLOAD . . . . .	17-159

## CONTENTS (cont)

Paragraph		Page
17-383	Routine CLOAD . . . . .	17-160
17-385	Routine ABLOAD . . . . .	17-160
17-387	Routine ABCLOAD . . . . .	17-161
17-390	Scaling . . . . .	17-162
17-397	Routine PUTCOM . . . . .	17-167
17-403	Routine PUTDECSF . . . . .	17-170
17-410	Routine DSPALARM . . . . .	17-172
17-411	Routine BINROUND . . . . .	17-172
17-416	Routine ARTHINSF . . . . .	17-173
17-417	Routine DEGINSF . . . . .	17-174
17-423	Routine OPTDEGIN . . . . .	17-176
17-428	Routine DPINSF . . . . .	17-177
17-431	Routine LOADVL . . . . .	17-178
17-437	Proceed . . . . .	17-180
17-441	Routine VBPROC . . . . .	17-181
17-443	Terminate . . . . .	17-181
17-445	Routine VBTERM . . . . .	17-182
17-447	Request Execution of a Job or Task . . .	17-182
17-455	Routine VBQEXEC . . . . .	17-185
17-459	Routine VBQWAIT . . . . .	17-186
17-463	Re-initialize the System . . . . .	17-188
17-464	Routine SLAPB . . . . .	17-188
17-470	Routine DSPMM . . . . .	17-191
17-473	Change Major Mode . . . . .	17-191
17-479	Routine MMCHANG . . . . .	17-196

## ILLUSTRATIONS

Number		Page
17-1	Primary Functions . . . . .	17-4
17-2	Primary Functions, Functional Flowchart (2 sheets) . . . . .	17-5
17-3	Character Displays . . . . .	17-18
17-4	Program Section Keyboard and Display . . .	17-55
17-5	Job CHARIN . . . . .	17-64
17-6	Multiplying by 00012 in Routine DECTOBIN . .	17-70
17-7	Decimal to Binary Conversion, Example 1 . .	17-73
17-8	Decimal to Binary Conversion, Example 2 . .	17-74
17-9	Subsection NVSUB . . . . .	17-85
17-10	Use of DSPLIST Waitinglist . . . . .	17-94
17-11	Decimal Display Verbs . . . . .	17-125
17-12	Decimal Display . . . . .	17-142
17-13	Load Routines Functional Flowchart . . . .	17-157
17-14	Routines VBRQWAIT and VBRQEXEC . . . .	17-187
17-15	Verb 36 (Fresh Start) . . . . .	17-189
17-16	Routine DSPMM . . . . .	17-193

## TABLES

Table		Page
17-1	Storing of MARK AGC Information in a Work Area . . . . .	17-14
17-2	DSPTAB Registers . . . . .	17-19
17-3	Verbs . . . . .	17-20
17-4	Nouns . . . . .	17-21
17-5	Data Input Buffer Registers . . . . .	17-27
17-6	Verb Definitions . . . . .	17-29
17-7	Noun Definitions . . . . .	17-41
17-8	DSPCOUNT Numbers . . . . .	17-59
17-9	Table INRELTAB . . . . .	17-67

## TABLES (cont)

Table		Page
17-10	Table CRITCON. . . . .	17-68
17-11	TESTNN Actions . . . . .	17-103
17-12	Addresses of Normal Nouns . . . . .	17-108
17-13	Index Quantities for Mixed Nouns . . . . .	17-112
17-14	Subaddress for Mixed Nouns . . . . .	17-113
17-15	VERBTAB Table . . . . .	17-115
17-16	LST2FAN Table. . . . .	17-116
17-17	Interpretation of Displayed Decimal Data . . .	17-126
17-18	Scaling Procedure Prior to Display . . . . .	17-131
17-19	Scaling Procedure Prior to Load Specified Address. . . . .	17-132
17-20	NNADTAB . . . . .	17-134
17-21	NNTYPTAB . . . . .	17-136
17-22	SFOUTAB1 Table . . . . .	17-138
17-23	SFOUTAB2 Table . . . . .	17-138
17-24	SFOUTABR Table . . . . .	17-139
17-25	Low Order Part of Augument . . . . .	17-144
17-26	High Order Part of Augument . . . . .	17-145
17-27	Decimal Data Scaling Routines. . . . .	17-163
17-28	Exiting From Routine PUTCOM . . . . .	17-169
17-29	Load Routine Scale Factors - High Order Part .	17-170
17-30	Load Routine Scale Factors - Low Order Part .	17-171
17-31	SFINTABR Table . . . . .	17-172
17-32	Data Entry . . . . .	17-179
17-33	Program Displays . . . . .	17-197

## ATTACHMENTS

## Number

17-1	Primary Functions, Detailed Flowchart. . . .
17-2	Subsection DSPIN Detailed Flowchart. . . .
17-3	CHARIN Routine (2 sheets) . . . .
17-4	Subsection NVSUB Detailed Flowchart . . . .
17-5	Subsection ENTER Detailed Flowchart . . . .
17-6	Octal Display Routines . . . .
17-7	Decimal Display Routines . . . .
17-8	Bump and Monitor Routines . . . .
17-9	Load Routines (2 sheets) . . . .
17-10	Routine MMCHANG. . . .

17-361. After program section K&D has been released for internal use, instruction INHINT is executed and the content of register MONSAVE is tested whether or not register GRABLOCK contains 00002. If register MONSAVE contains 00000, indicating Task MONREQ (paragraph 17-362) is not contained in the Waiting List (i. e. , there is no Monitor activity within the computer), then a request is made to the Waitlist to schedule Task MONREQ with a 10 msec delay. If register MONSAVE contains a positive quantity, indicating there is monitor activity (although in the suspended state) within the computer, then the request for scheduling Task MONREQ is bypassed. After register MONSAVE is tested and the appropriate action taken, the content of register MPAC (the Verb and Noun code) and the content of register MPAC+1 (the Noun address) is transferred to registers MONSAVE and MONSAVE1 respectively. Then instructions RELINT and TC ENTRET are given. Register ENTRET contains instruction TC ENDOFJOB or TC NVSUBEND.

#### 17-362. Task MONREQ

17-363. Task MONREQ requests the execution of a Job to display the content of one, two, or three locations within the computer. Since Task MONREQ requests rescheduling of itself with a 0.5 sec delay, a monitor function results. A failure of Task MONREQ to request rescheduling terminates the monitor function. The monitor function is terminated if, during the execution of intervening Jobs, a non-CHARIN Job passes the operator/internal interlock which enters a ONE into bit position 15 of register MONSAVE1 or if the operator keys in VERB, 3, 4, and ENTER. If Task MONREQ finds that bit position 15 of register MONSAVE1 contains a ONE, routine KILLMON is executed. Routine KILLMON clears registers MONSAVE and MONSAVE1 and executes instruction TC TASKOVER. If register MONSAVE1 does not contain a ONE in bit position 15, Task MONREQ requests rescheduling of itself after a 0.5 sec delay. A request is then made to the Job Control to Schedule Job MONDO with priority 30<sub>8</sub> and instruction TC TASKOVER is executed.

#### 17-364. Job MONDO

17-365. Job MONDO is terminated immediately if bit position 15 of register MONSAVE1 contains a ONE. (The kill-monitor bit is on.) If the kill-monitor bit is not on, the content of register DSPLOCK is tested. If register DSPLOCK contains a positive quantity, indicating program section K&D is not available for internal use and monitor activity should be suspended, then program control is transferred to

routine MONBUSY. Routine MONBUSY does not perform the requested display; it turns on indicator KEY RLSE, by entering a ONE into bit position 3 of OUT1 and then executes instruction TC ENDOFJOB.

17-366. If register DSPLOCK contains 00000, indicating program section K&D is available for internal use, the Noun code is transferred to register NVTEMP and program control is transferred to routine NVSUBON. Routine NVSUBMON is a special entrance to routine NVSUB that does not turn on the kill-monitor bit. (Refer to paragraph 17-230.) Since register NVTEMP contains zeros in bit positions 12 through 7, and the Noun code in 6 through 1, routine NVSUB enters the Noun code into register NOUNREG and displays the Noun code in the Noun displays. Thereafter, the Verb code is changed from the monitor Verb to the corresponding display Verb by transferring the content of bit positions 9 through 7 of register MONSAVE to bit positions 3 through 1 of register VERBREG. (Bit positions 15 through 4 of register VERBREG contain zeroes.) Instruction TC PASTEBV is then entered into register ENTRET, the content of bit positions 14 through 1 of register MONSAVE1 is transferred to register NOUNADD, and program control is transferred to routine TESTNN.

17-367. Routine TESTNN decodes the Noun code, obtains the address specified by the Noun, and then transfers control to the corresponding Verb routine. For further details, refer to table 17-11 and to paragraphs 17-265 and 17-266. After the requested display has been completed, instruction TC ENTRET is executed. Since register ENTRET contains instruction TC PASTEBV, routine PASTEBV is executed. Routine PASTEBV transfers the content of bit positions 12 through 7 of registers MONSAVE (the Monitor Verb code) to register NVTEMP and then executes routine NVSUBMON. After this transfer, the content of bit positions 15 through 13 and 6 through 1 of register NVTEMP contain zeroes. Thus, when routine NVSUBMON is executed, the Monitor Verb code is transferred to register VERBREG and displayed in the VERB displays. When program control is transferred back to routine PASTEBV via routine NVSUBEND instruction TC ENDOFJOB is executed.



## 17-368. REQUEST DATA ENTRY INTO SPECIFIED LOCATIONS

17-369. Data is entered into a specified location by routines that:

- a. Request a data entry into the computer.
- b. Respond to the request for a data entry.

17-370. A request for a data entry (either an octal or a decimal data word) is performed whenever routine VBFANDIR (paragraph 17-281) is executed and the Verb code is 21, 22, 23, 24, or 25. These Verb codes cause the execution of routines ALOAD, BLOAD, CLOAD, ABLOAD, or ABCLOAD respectively (attachment 17-9, sheet 1). Observe that the request for a data load has been defined as a function of routines ALOAD, BLOAD, CLOAD, ABLOAD, and ABCLOAD. Such requests may have originated in two ways: The most common way is an operator action using Verbs 21 through 25; the other way is subsection NVSUB using these Verb codes. The distinction made here is not important for understanding the internal behavior of the load routines, but is mentioned to prepare the reader for concepts to be discussed under Proceed and Terminate (paragraphs 17-437 and 17-443).

17-371. Routines ALOAD, BLOAD, or CLOAD transfer program control to routine REQDATX, REQDATY, or REQDATZ respectively. Routines ABLOAD and ABCLOAD change the displayed Verb code from 24 and 25 respectively to 21 and then transfer program control to routine REQDATX. The REQDAT routines perform the following operations:

- a. Clear the corresponding Data Input Buffer Registers (XREG and XREGLP, YREG and YREGLP, or ZREG and ZREGLP).
- b. Clear the sign indicator register (DECBRNCH).
- c. Blank the corresponding display register (R1, R2, or R3).
- d. Enter the proper quantity (00016, 00011, or 00004) into register DSPCOUNT.
- e. Enter the complemented form of the return address into register REQRET (paragraph 17-277).
- f. Initiate the flashing of the VERB and NOUN displays.
- g. Execute instruction TC ENTRET. Register ENTRET contains either instruction TC ENDOFJOB or TC NVSUBEND. Refer to paragraph 17-240 for a discussion of an internally initiated load request.

17-372. Once a request has been made to enter data into the computer, the operator must respond to this request by making one of the following keyboard entries:

- a. Depressing keys + or - (decimal data only), D1, D2, D3, D4, D5, ENTER (enters the requested data)
- b. Depressing keys VERB, 3, 3, ENTER (proceed)
- c. Depressing keys VERB, 3, 4, ENTER (terminate)

17-373. If the load request was internally initiated via NVSUB, the operator must respond with one of the above entries because a non-CHARIN Job is awaiting a response to the request. This Job may be in the dormant state as a result of executing routine ENDIDLE, or else it may be checking the content of register LOADSTAT to determine the state of the load request. The Job is reactivated and register LOAD-STAT is reset only as a result of executing one of the three responses via the keyboard.

17-374. If the load request was keyboard initiated, the operator must execute one of three responses if only to terminate the flashing of the VERB and NOUN displays. Entry (a) of paragraph 17-372, the normal response to a request, causes the continuation of routine ALOAD, BLOAD, CLOAD, ABLOAD, or ABCLOAD. (Refer to figure 17-13). This entry results in either the requesting of another data word (if the request were made by routines ABLOAD or ABCLOAD) or, the scaling of the data already contained in the Data Input Buffer Registers (if the load is completed) and the transferring of the scaled data into the location(s) specified by the Noun. After all the data has been entered into the location(s) specified by the Noun, register LOADSTAT is set to 77777 to indicate the requested data load has been completed; any Job put into the dormant state by routine ENDIDLE while awaiting the completion of the data load is reactivated at location P+3 where P is the location from which the TC ENDIDLE was done. (Refer to paragraph 17-240). This will henceforth be referred to as scaling.

17-375. Entry (b) of paragraph 17-372 causes the execution of routine VBPROC. The initiating group of routines, (paragraph 17-250) is executed whenever key ENTER is depressed because depressing key VERB as a part of entry (b) resets register REQRET to 00000. This entry is taken when the operator chooses not to supply the requested data, but wishes the non-CHARIN Job to continue the execution of the routine that initiated the data load request as best it can without new data.

17-376. Routine VBPROC enters the quantity 00001 into register LOAD-STAT and terminates the flashing of the VERB and NOUN displays. It also reactivates any Job put into the dormant state by routine ENDIDLE at location P+2 where P is the location from which the TC ENDIDLE was done.



17-377. Entry (c) of paragraph 17-372 causes the execution of routine VBTERM. This entry is used if the operator chooses not to recognize the data load request and desires the non-CHARIN Job to terminate the routine that would utilize the requested data. (Note: Only the operator's desire to terminate the routine is communicated to the non-CHARIN Job. Whether or not the routine is terminated depends upon the non-CHARIN Job itself.) Verb 34 can also be used to terminate Monitor activity within the computer. Routine VBTERM enters a ONE into bit position 15 of register MONSAVE1, enters 77776 into register LOADSTAT, terminates the flashing of the VERB and NOUN displays, and reactivates any non-CHARIN Job put into the dormant state by routine ENDIDLE at location P+1. This will henceforth be referred to as terminate.

#### 17-378. Requesting Routines

##### 17-379. Routine ALOAD

17-380. Routine ALOAD, executed when the Verb code is 21, transfers program control to routine REQDATX to request a data word entry into the location specified by the first component of the Noun. During the execution of routine REQDATX, the content of register REQRET is set so that, should the operator choose to enter the requested data, routine ALOAD is continued at a location which causes the following:

- a. The quantity 00000 is transferred to register A.
- b. Program control is transferred to routine PUTCOM where scaling is initiated.

Routine REQDATX is described in paragraph 17-351. Routine REQDATX executes instruction TC ENTRET which transfers program control to either routine ENDOFJOB or NVSUBEND.

##### 17-381. Routine BLOAD

17-382. Routine BLOAD, executed when the Verb code is 22, enters a ONE into bit position 15 of register CLPASS to block the backing-up function of routine CLEAR (paragraph 17-166). (It is not necessary to enter a ONE into bit position 15 of register CLPASS during the execution of routine ALOAD as it is impossible to back up beyond display register R1). Thereafter, program control is transferred to routine REQDATY to request a data word entry that will be stored in the location specified by the second component of the Noun. Routine REQDATY operates as described in paragraph 17-351. The content of register REQRET is set so that, if the operator chooses to enter the requested data, routine BLOAD is continued at a location which causes the following:

- a. The quantity 00001 is transferred to register A.
- b. Program control is transferred to routine PUTCOM where scaling is initiated.

Routine REQDATY executes instruction TC ENTRET which transfers program control to either routine ENDOFJOB or NVSUBEND.

17-383. Routine CLOAD

17-384. Routine CLOAD,executed when the Verb code is 23, enters a ONE into bit position 15 of register CLPASS to block the backing-up function of routine CLEAR. Program control is then transferred to routine REQDATZ to request a data word entry that will be stored in the location specified by the third component of the Noun. Routine REQDATZ operates as described in paragraph 17-351. The content of register REQRET is set so that, if the operator chooses to enter the requested data, routine CLOAD is continued at a location which causes the following:

- a. The quantity 00002 is transferred to register A.
- b. Program control is transferred to routine PUTCOM where scaling is initiated.

Routine REQDATZ executes instruction TC ENTRAT which transfers program control to either routine ENDOFJOB or NVSUBEND.

17-385. Routine ABLOAD

17-386. Routine ABLOAD transfers the quantity 00021 to register VERBREG and executes routine UPDATVB to display the quantity 21 in the VERB displays. Routine UPDATVB operates as described in paragraph 17-233 except that register Q contains the address of instruction TC REQDATX. Thus, after the Verb code 21 is displayed, program control is transferred to routine REQDATX to request a data word entry that will be stored in the location specified by the first component of the Noun. After the request is made, program control is transferred to either routine NVSUBEND or ENDOFJOB. During the execution of routine REQDATX, register REQRET is set so that, if the operator chooses to enter the requested data, routine ABLOAD is continued at a location which causes the following:

- a. The quantity 00022 is transferred to register VERBREG.
- b. Program control is transferred to routine UPDATVB to display the quantity 22 in the VERB displays.

- c. Routine REQDATY requests the entry of a data word that will be stored in the location specified by the second component of the Noun. Program control is then transferred to routine ENDOFJOB. During the execution of routine REQDATY, the content of register REQRET is set so that, if the operator chooses to enter the second requested data word, routine ABLOAD is continued at a location that transfers program control to routine PUTXY. Routine PUTXY transfers the quantity 00000 to register A and then transfers program control to routine PUTCOM.

If the operator enters the Verb codes for Proceed or Terminate in response to a request for a second data word, routine PUTCOM will not be executed. Thus, the first data word contained in the data input buffer registers XREG and XREGLP, will be neither scaled nor entered into the location specified by the c(NOUNADD).

#### 17-387. Routine ABCLOAD

17-388. Routine ABCLOAD operates similarly to routine ABLOAD (paragraph 17-385). However, upon executing routine REQDATY (paragraph 17-386c), the content of register REQRET is set so that, if the operator chooses to enter the second data word into the location specified by the second component of the Noun, routine ABCLOAD will be continued at a location that causes the following:

- a. The quantity 00023 is transferred to register VERBREG.
- b. Routine UPDATVB causes the quantity 23 to be displayed in the VERB displays.
- c. Routine REQDATZ requests that a third data word be entered into the location specified by the third component of the Noun. Program control is then transferred to routine ENDOFJOB.
- d. During the execution of routine REQDATZ the content of register REQRET is set so that, should the operator choose to enter the third data word, routine ABCLOAD will be continued at a location that causes program control to be transferred to routine PUTXYZ. Routine PUTXYZ transfers the quantity 00000 to register A and then transfers program control to routine PUTCOM. Thus, during the execution of routine ABCLOAD, it is possible for as many as 21 CHARIN Jobs to be executed before routine PUTCOM is executed by depressing keys ±D1, D2, D3, D4, D5, ENTER; ±D1', D2', D3', D4', D5', ENTER; ±D1'', D2'', D3'', D4'', D5'', ENTER.



17-389. If the operator depresses keys VERB, 3, 3, ENTER; or VERB, 3, 4, ENTER in response to a request for the second or third data words, routine REQDATZ or routine PUTCOM, respectively, are executed. Thus, the data previously entered is neither scaled nor entered into the locations specified by the Noun.

17-390. Scaling

17-391. Scaling (paragraph 17-374) is executed after all requested data has been entered into the Data Input Buffer Registers. Scaling is initiated by routine PUTCOM which is executed once for each data word entered. Thus, if either routine PUTXY or PUTXYZ was executed prior to routine PUTCOM, then routine PUTCOM will be executed two or three times respectively.

17-392. The data now contained in the Data Input Buffer Registers may be either octal or decimal; if decimal, it is in fractional form. (Refer to routine NUM, paragraph 17-129). Octal data requires no scaling. Thus, if register DECBRNCH contains 00000, indicating octal data, routine PUTCOM obtains the address specified by the Noun, and then transfers the content of XREG, YREG, or ZREG to the specified address. The specific REG register utilized is determined by the content of register A upon execution of this routine. (Register A contains 00000, 00001, or 00002 if routine PUTCOM is being executed to scale the data specified by the first, second, or third component of the Noun respectively.)

17-393. If the content of register DECBRNCH is positive, indicating decimal data, routine PUTCOM not only obtains the address into which the data is to be entered, but also scales the data before storing it. Routine PUTCOM transfers program control to one of five scaling routines (attachment 17-9, sheet 2) depending upon the location into which the data is to be entered. However, before transferring to the scaling routines, the content of registers XREG and XREGLP, YREG and YREGLP, or ZREG and ZREGLP is transferred to register MPAC and MPAC+1. The five scaling routines are described in table 17-27.

17-394. It is important to note that the content of register DECBRNCH is reset to 00000 each time a data word is requested. The content of register DECBRNCH, upon execution of routine PUTCOM, depends upon the presence or absence of an algebraic sign associated with the last data word inserted. Thus, no mixing of octal and decimal data is permitted within a multi-component load.

Table 17-27

## DECIMAL DATA SCALING ROUTINES

Routine	Types of Data Referenced by the Noun	Action Taken
BINROUND	fractional	Routine BINROUND rounds the double precision quantity in registers MPAC and MPAC+1 to a single precision quantity which is then entered into the address specified by the first, second, or third component of the Noun, i. e., c(NOUNADD), c(NOUNADD)+1, or c(NOUNADD)+2. No scaling is done as register MPAC and MPAC+1 contain data in fractional form. Refer to routine NUM, paragraph 17-129.
ARTHINSF	whole	Routine ARTHINSF multiplies the content of registers MPAC and MPAC+1 by the double precision quantity [ 00006; 03240], which is stored at locations SFINTAB1 and SFINTAB2. The product is then multiplied by the quantity $2^{14}$ . The double precision quantity represents the octal equivalent of the decimal value $\frac{10^5}{2^{14}} \cdot \frac{1}{2^{14}} = 6.1035_{10} \cdot \dots \cdot \frac{1}{2^{14}}$ . Thereafter, routine BINROUND rounds the product to a single precision quantity, and the product is stored in the location specified by the c(NOUNADD), c(NOUNADD)+1 or c(NOUNADD)+2.
DEGINSF	angular	Routine DEGINSF multiplies the content of registers MPAC and MPAC+1 by the double precision quantity [ 26161; 30707] which is stored at locations DEGCON1 and DEGCON1+1 respectively. The result of this multiplication is rounded to a double precision quantity and then multiplied by $2^3$ . The quantity [ 26161; 30707] represents the



Table 17-27  
(cont)  
DECIMAL DATA SCALING ROUTINES

Routine	Type of Data Referenced by the Noun	Action Taken
DEGINSF (cont)	angular (cont)	<p>value <math>\frac{10^5}{18000} \cdot \frac{1}{8} = \frac{5.55555}{8} = 0.6944444_{10} = 0.5434343434_8</math>. The expression <math>\frac{10^5}{18000}</math> is used instead of <math>\frac{10^5}{36000}</math> since angles (except gyro angles) are stored in TWO's complement numbers with 15 value bits. Division by 8 is necessary because the computer can only multiply by a fractional number. The result of the above multiplication is stored in TWO's complement form in the location specified by the c(NOUN-ADD), c(NOUNADD)+1, or c(NOUNADD)+2.</p>
OPTDEGIN	optics	<p>Routine OPTDEGIN will not accept a negative optics angle. Thus, if register MPAC contains 77777 or a negative quantity, routine DSP-ALARM is executed to turn on indicator CHECK FAIL and terminate the routine. If register MPAC is a positive quantity or 00000, the content of bit position 13 of register WASOPSET is tested to determine the range of the entered data.</p> <p>If bit position 13 contains a ZERO indicating data is in the <math>180^\circ</math> range (the data is obtained from the scanning telescope to a precision of 0.01 degrees), routine OPTDEGIN multiplies the content of registers MPAC and MPAC+1 by the double precision quantity [26161;30707], and then shifts the product one place to the left. This shift is necessary since the scale factor constant used is the same one used for</p>

Table 17-27  
(cont)  
DECIMAL DATA SCALING ROUTINES

Routine	Type of Data Referenced by The Noun	Action Taken
OPTDEGIN (cont)	optics	<p>360° angular information. Thus, the result of the shift represents the product that would have been obtained had the scale factor represented the octal equivalent of <math>\frac{10^5}{09000} \cdot \frac{1}{8}</math> instead of <math>\frac{10^5}{18000} \cdot \frac{1}{8}</math>.</p> <p>Then the product is rounded to double precision, multiplied by 2<sup>3</sup> (to compensate for the scale factor value), rounded to single precision, and entered in TWO's complement form into the location specified by the c(NOUNADD), c(NOUNADD)+1, or c(NOUNADD)+2.</p> <p>If the data entered is in the 90° range (obtained from the sextant to a precision of 0.001 degrees), a bias of 19.775° must be subtracted before the content of registers MPAC and MPAC+1 is multiplied by the double precision quantity [21616;70771] which is stored at locations DEGCON2 and DEGCON2+1. The quantity [21616;07071], represents the value <math>\frac{10^5}{45000} \cdot \frac{1}{4} = \frac{2.222222}{4} = 0.55555_{10} = 0.434343434_8</math>. Division by four is necessary because the computer can only multiply by a fractional number. Thereafter, the product is rounded to a double precision quantity and control is transferred to routine DEGINSF2. Routine DEGINSF2 multiplies this rounded double precision quantity by 2<sup>2</sup>. The result is rounded to single precision and is stored in TWO's complement form in the location specified by the c(NOUNADD), c(NOUNADD)+1 or c(NOUNADD)+2.</p>

Table 17-27  
(cont)  
DECIMAL DATA SCALING ROUTINES

Routine	Type of Data Referenced by the Noun	Action Taken
DPINSF	gyro or time in seconds	<p>Routine DPINSF multiplies the content of registers MPAC and MPAC+1 by the double precision quantity contained in registers SFTEMP1 and SFTEMP2. These registers contain [00006; 03240] or [00001; 30707] depending upon whether time in seconds or gyro data, respectively, has been entered into the registers. The quantity [00006; 03240] is equivalent to <math>\frac{10^5}{2^{14}} \cdot \frac{1}{2^{14}}</math> which is the same scale factor constant used by routine ARTHINSF. The quantity [00001; 30707] represents the octal equivalent of <math>\frac{10^5}{56250} \cdot \frac{1}{2^{14}} = \frac{1.77777}{2^{14}}</math>. In the case of gyro data or time in seconds, division by <math>2^{14}</math> is necessary as this type of information is stored in double precision. Thus, the product is not multiplied by <math>2^{14}</math> as is done in routine ARTHINSF. After the multiplication, the triple precision product is rounded to a double precision quantity before it is stored in ONE's complement form in two consecutive locations specified by the Noun. The low order bits will be stored in the address specified by the c(NOUNADD)+1, c(NOUNADD)+2, or c(NOUNADD)+3 depending upon whether the first, second, or third component, respectively, of the Noun is being scaled. The high order bits will be stored in the location specified by the c(NOUNADD), c(NOUNADD)+1, or c(NOUNADD)+2, respectively.</p>

17-395. After the data has been entered into the specified address, program control is transferred to routine PUTCOM again (with the content of register A equal to 00001 or 00002) or to routine LOADLV. If routine PUTXY or PUTXYZ called routine PUTCOM, routine LOADLV will not be executed until routine PUTCOM has been executed two or three times respectively.

17-396. Routine LOADLV enters 77754 into register DSPCOUNT to block any clearing action and to indicate an operation key must be depressed next. The routine then enters 77777 into register LOADSTAT to indicate the requested data has been entered. Register CADRSTOR is then tested and if it contains 00000, indicating routine ENIDLE has not put a Job into the dormant state, instruction TC ENDOFJOB is executed. However, if register CADRSTOR contains a positive quantity, indicating routine ENIDLE has put a Job into the dormant state, the Job is reactivated at location P+3 where P is the location from which the TC ENIDLE was done. Thereafter, program section K&D is released for internal use and instruction TC ENDOFJOB is executed.

17-397. Routine PUTCOM

17-398. When routine PUTCOM is executed, register A contains one of the following quantities:

- a. 00000 if data is to be entered into the location specified by the first component of the Noun.
- b. 00001 if data is to be entered into the location specified by the second component of the Noun.
- c. 00002 if data is to be entered into the location specified by the third component of the Noun.

17-399. Routine PUTCOM (attachment 17-9, sheet 2) transfers the content of register A to register DECOUNT, transfers the content of register Q to register DECRET, and obtains the address specified by the Noun if it is mixed (Noun code  $\geq 55$ ). There is no need to obtain the specified address if the Noun is normal (Noun code  $< 55$ ) since routine ENTER entered the address specified by the first component of the Noun into register NOUNADD when the initiating group routines were executed. Thus, program control is transferred to routine PUTCOM1 if the Noun is normal.





17-400. Routine PUTCOM obtains the address specified by the first, second, or third component of a mixed Noun as described in paragraph 17-276(c), except that, before the second indexing operation, the c(DE-COUNT, instead of the c(NOUNTEM) is added to the content of register A.

Mixed Nouns components are not generally stored in consecutive locations; whereas, normal Noun components are always stored in consecutive locations. Once the address of the current component has been obtained, it is then decremented by the content of register DECOUNT and entered into register NOUNADD. The address is decremented before it is entered into register NOUNADD because the routines which accomplish the actual transfer of scaled data to the location specified by the current component assume the noun is normal and hence form the storage address by incrementing c(NOUNADD) by 0, 1, or 2. Since these same routines are used to store scaled data for mixed nouns, which are not usually loaded into consecutive locations and for which the address of each component is obtained individually, it is necessary to compensate for increments within the load routines by preceding them with a decrementing operation that is dependent on the particular component currently being loaded. After the address of the current component has been entered into register NOUNADD, program control is transferred to routine PUTCOM1.

17-401. Routine PUTCOM1 enters the quantity 00000 into register OVFind. (The content of register OVFind has significance only if the data entered is decimal.) Thereafter, the content of register DECOUNT (0, 1, or 2) is indexed with the address of location XREGLP and the content of this indexed location (the low order part of the entered decimal data) is transferred to register MPAC+1 from register REGLP. The content of register DECOUNT is then indexed with the address of location XREG and the content of this indexed location (the high-order part of the entered data) is transferred to register MPAC from register REG). The content of register DECBRNCH is tested and if it is 00000, indicating octal data, program control is then transferred to routine PUTCOM2. If register DECBRNCH contains a positive quantity, indicating decimal data, program control is transferred to PUTDECSF.

17-402. Routine PUTCOM2 is called directly by routine PUTCOM if the data entered is octal. However, routine PUTDECSF is executed before calling routine PUTCOM2 if the data entered is decimal. Routine PUTCOM2 transfers the content of register MPAC, which is either scaled decimal data or octal data, to register A and then executes instruction TC DECRET. Register DECRET contains the address P+1 where P is the address of the location from which TC PUTCOM was done. Address P+1 specifies one of eight different locations, depending upon the particular load routine that called routine PUTCOM. The specified location contains an instruction that initiates one of the actions shown in table 17-28.

TABLE 17-28  
EXITING FROM ROUTINE PUTCOM

Routine That Called Routine PUTCOM	Registers whose data is scaled by Routine PUTCOM	Action Initiated by Executing Instruction TC DECRT	
		Contents which specify location to which scaled data is transferred	Routine to which Program Control is transferred
ALOAD	XREG and XREGLP	c(NOUNADD)	LOADLV 
BLOAD	YREG and YREGLP	c(NOUNADD) +1	LOADLV
CLOAD	ZREG and ZREGLP	c(NOUNADD) +2	LOADLV
PUTXY	XREG and XREGLP	c(NOUNADD)	PUTCOM with c(A) = 00001 
PUTXY	YREG and YREGLP	c(NOUNADD) +1	LOADLV
PUTXYZ	XREG and XREGLP	c(NOUNADD)	PUTCOM with c(A) = 00001
PUTXYZ	YREG and YREGLP	c(NOUNADD) +1	PUTCOM with c(A) = 00002
PUTXYZ	ZREG and ZREGLP	c(NOUNADD) +2	LOADLV
 Refer to paragraph 17-431  Refer to paragraph 17-397			

## 17-403. Routine PUTDECSF

17-404. Routine PUTDECSF, which is executed when routine PUTCOM finds that the information has been specified in decimal form, transfers program control to routine SFCONUM to obtain the scale factor constant code for the component being scaled (paragraph 17-316). After routine SFCONUM has been executed, register A contains one of the following quantities to specify the scale factor constant code:

- a. 00000 if the component references whole data or time in seconds;
- b. 00001 if the component references time in hours;
- c. 00002 if the component references angular and optics data;
- d. 00003 if the component references gyro degrees.

17-405. Routine PUTDECSF then transfers the scale factor constant code from register A to register SFTEMP2. Thereafter, the scale factor constant code is indexed with the address of location SFINTAB1 and the content of this indexed location, which is the high order part of the scaling constant, is transferred to register SFTEMP1. The quantities thus obtained are shown in table 17-29.

TABLE 17-29

## LOAD ROUTINE SCALE FACTORS - HIGH ORDER PART

Location	Content	Interpretation
SFINTAB1	00006	High order part of $\frac{10^5}{2^{14}} \cdot \frac{1}{2^{14}}$
SFINTAB1+1	00000	Hours load not permitted
SFINTAB1+2	00000	Will be obtained in routine OPTDEGIN or DEGIN
SFINTAB1+3	00001	High order part of $\frac{10^5}{56250} \cdot \frac{1}{2^{14}}$



17-406. The content of register SFTEMP2 (the scale factor constant code) is then indexed with the address of location SFINTAB2. The content of this indexed location, which is the low order part of the scaling constant, is transferred to register SFTEMP2. The quantities thus obtained are shown in table 17-30.

TABLE 17-30

## LOAD ROUTINE SCALE FACTORS - LOW ORDER PART

Location	Content	Interpretation
SFINTAB2	03240	Low order part of $\frac{10^5}{2^{14}} \cdot \frac{1}{2^{14}}$
SFINTAB2+1	00000	Hours load not permitted
SFINTAB2+2	00000	Will be obtained in routine OPTDEGIN or routine DEGIN
SFINTAB2+3	30707	Low order part of $\frac{10^5}{52650} \cdot \frac{1}{2^{14}}$

Refer to Table 17-28 for the significance of these scale factors.

17-407. After the scaling constant has been obtained program control is transferred to routine PUTSFNOR or routine SFRUTMIX depending upon whether register MIXBR contains 00001 (normal noun) or 00002 (mixed noun) respectively. Routine PUTSFNOR transfers control to routine SFRUTNOR which obtains the scale factor code. The code is obtained by indexing the Noun code with location NNTYTAB and by transferring the content of bit positions 10 through 6 of this location to bit positions 5 through 1 of register A. (Routine SFRUTMIX is described in paragraph 17-317.) Upon exiting from either routine SFRUTMIX or SFRUTNOR, the scale factor routine code is contained in register A and program control is transferred to routine PUTDCSF2.

17-408. Routine PUTDCSF2 obtains the complete address of the next routine to be executed (a scale factor routine) by indexing the content of register A (the scale factor routine code) with location SFINTABR. The content of this indexed location (table 17-31) is transferred to register A and routine BANKJUMP is executed to transfer program control to the scale factor routine specified.



TABLE 17-31  
SFINTABR TABLE

Location	Content
SFINTABR	Complete address of routine BINROUND
SFINTABR+1	Complete address of routine DEGINSF
SFINTABR+2	Complete address of routine ARTHINSF
SFINTABR+3	Complete address of routine DPINSF
SFINTABR+4	Complete address of routine DSPALARM
SFINTABR+5	Complete address of routine OPTDEGIN

17-409. The scaling routines are described in detail in the following paragraphs. The order of presentation is not as shown in table 17-31 but rather for ease of explanation and referencing.

17-410. Routine DSPALARM. Routine DSPALARM is executed when the operator tries to enter data representing time in hours. This routine enters a ONE into bit position 5 of register OUT1 which turns on indicator CHECK FAIL. Since register ENTRET can only contain instruction TC ENDOFJOB at this point, decrementing the content of register NVQTEM by one has no effect. (The content of register NVQTEM is only utilized during the execution of NVSUBEND.) Thereafter, instruction TC ENTRET is executed and the CHARIN Job scheduled is terminated by depressing key ENTER.

17-411. Routine BINROUND. Routine BINROUND transfers program control to routine 2ROUND. Routine 2ROUND doubles the quantity contained in register MPAC+1. If overflow occurs, it is added to the quantity contained in register MPAC. If the overflow occurs as a result of this addition, the overflow is stored in register OV FIND and program control is transferred back to routine BINROUND. Routine BINROUND tests the content of register OV FIND. If register OV FIND contains 00000, program control is transferred to routine ENDSCALE (paragraph 17-415).

17-412. If register OVFind contains a non-zero quantity, indicating that too large a quantity (either positive or negative) has been entered, program control is transferred to routine DSPALARM to turn on indicator CHECK FAIL. Instruction TC ENTRET is then executed and register ENTRET contains instruction TC ENDOFJOB.

17-413. If the Load request was initiated originally via a call to NVSUB, the operator must keep in mind there is a non-CHARIN Job awaiting his response to the load request. This Job may be in the dormant state as a result of executing routine ENDIDLE, or else it may be checking the content of register LOADSTAT to determine the state of the load request. (Refer to paragraphs 17-243 and 17-244, respectively.) Thus, if indicator CHECK FAIL is illuminated after the operator has depressed key ENTER, he must key in Verb 33 (VBPROC), Verb 34 (VBTERM), or manually reinitiate the entire load sequence by keying in the original Load Verb and Noun. If the original Load Verb were multicomponent, it would have been changed during the loading sequence. (Refer to paragraphs 17-385 and 17-387.) Thus, if indicator CHECK FAIL is turned on during the scaling function, all requested data words must be re-entered. The program will not indicate which of the data words in a multicomponent load caused program control to be transferred to routine DSPALARM to illuminate indicator CHECK FAIL. The operator must check the quantities in display registers R1, R2, or R3 to determine which is in error.

17-414. If the operator is sure that no non-CHARIN Job is awaiting his response to a load request, he is under no obligation to take further action. However, it is likely that he will want to reenter the Load Verb and Noun and then continue to reenter the data.

17-415. Routine ENDSCALE, which is executed after all necessary scaling has been done, transfers the complete address of routine PUTCOM2 to register A and then executes routine BANKJUMP. Routine PUTCOM2 transfers the scaled data to be stored from register MPAC to register A and then transfers program control back to the routine that called routine PUTCOM. Refer to paragraph 17-402.

17-416. Routine ARTHINSF. Routine ARTHINSF transfers program control to routine DMP to multiply the double precision data contained in registers MPAC and MPAC+1 by the double precision scale factor constant contained in registers SFTEMP1 and SFTEMP2. Upon exiting from routine DMP, the product is stored in registers MPAC, MPAC+1, and MPAC+2, with high order bits of the product in register MPAC. Then routine ARTHINSF multiplies the content of register MPAC, MPAC+1, and MPAC+2 by the decimal quantity  $2^{14}$ . This

multiplication is implemented within routine ARTHINSF by transferring the content of register MPAC+2 to register MPAC+1. Thereafter, the b(MPAC+1) is transferred to register MPAC and the b(MPAC) is transferred to register A. If register A contains any quantity except 00000, program control is transferred to routine DSPALARM because the data loaded is too large. (The action taken as a result of executing this routine has been previously discussed in paragraphs 17-412 through 17-414.) If register A contains 00000, program control is transferred to routine BINROUND which was described in paragraph 17-411.

17-417. Routine DEGINSF. Routine DEGINSF has three entrances: one at location DEGINSF, one at location DEGINSF+2 which is utilized by routine OPTDEGIN if the data is obtained via the telescope, and one at location DEGINSF2 which is utilized by routine OPTDEGIN if the data is obtained via the sextant. Routine DEGINSF transfers program control to routine DMP to multiply the data contained in registers MPAC and MPAC+1 by the scale factor constant contained in locations DEGCON1 and DEGCON1+1. Locations DEGCON1 and DEGCON1+1 contain 26161 and 30707 respectively. These quantities represent the decimal quantity  $\frac{10^5}{18000} \cdot \frac{1}{8}$ . (Division by 8 is necessary because the computer can only multiply by a fractional number.) Upon exiting from routine DMP, the product is stored in registers MPAC, MPAC+1, and MPAC+2.

17-418. Routine DEGINSF then proceeds to round the product by testing the content of register MPAC+1. (The content of register MPAC+2 is ignored.) If register MPAC+1 contains a positive quantity, the quantity 02000 is added to the content of register MPAC+1. Similarly, if register MPAC+1 contains a negative quantity, the quantity 75777 is added to the content of register MPAC+1. If register MPAC+1 contains 00000 or 77777, the content of register MPAC+1 remains untouched. If overflow or underflow occurs as a result of the addition, then the quantity 00001 (for overflow) or 77776 (for underflow) is added to the content of register MPAC. If this results in either overflow or underflow, then the quantity 00001 or 77776 is entered into register OVFINd.

17-419. Program control is then transferred, via routine SWCALL, to address SL1+3 in routine SL1 to shift the content of registers MPAC and MPAC+1 one place to the left. In reality, the shift is carried out by addition, the equivalent of multiplying by 2. If overflow or underflow occurs as a result of this shift, the quantity 00001 or 77776, respectively, is added to the quantity contained in register OVFINd. Then, program control is transferred back to routine DEGINSF at address DEGINSF2 via routine SWRETURN.

17-420. Routine DEGINSF2 is also called by subroutine OPTDEGIN2 in routine OPTDEGIN. This entrance is used when optics data obtained via the sextant is being scaled. Routine DEGINSF2 multiplies the content of registers MPAC and MPAC+1 by 4. This multiplication is implemented by transferring program control twice (via routine SWCALL) to routine SL1 at address SL1+3 where the content of register MPAC+1 is shifted left one position each time. If overflow or underflow occurs, the quantity 00001 or 77776 is added to the content of register MPAC. Thereafter, the content of register MPAC is shifted one place to the left. If overflow or underflow occurs, the quantity 00001 or 77776 is added to the content of register OVFINd. After program control is transferred back to routine DEGINSF2 via routine SWRETURN, the content of register OVFINd is tested. If register OVFINd does not contain 00000, program control is transferred to routine DSPALARM which was previously discussed in paragraphs 17-412 through 17-414.

17-421. If register OVFINd contains 00000, routine DEGINSF again transfers program control (via routine SWCALL) to location SL1+3 in routine SL1 to shift the content of registers MPAC and MPAC+1 one additional place to the left. If overflow or underflow occurs, the quantity 00001 or 77776, respectively, is entered into register OVFINd. Thereafter, program control is transferred back to routine DEGINSF2 (via routine SWRETURN) where the content of register MPAC is tested. If register MPAC contains a negative quantity, the content of register MPAC is incremented by one since angular and optics information is stored in TWO's complement form. If register MPAC contains 77777, the quantity 00000 is entered into register MPAC for the same reason. Thereafter, program control is transferred to routine SIGNFIX.

17-422. Routine SIGNFIX tests the content of register OVFINd. If register OVFINd contains 00000, indicating that no overflow or underflow occurred during the shifting, program control is transferred to routine ENDSCALE which was discussed in paragraph 17-415. If register OVFINd contains 00001, indicating overflow occurred during the shifting, program control is transferred to routine SGNT01 where a ONE is entered into bit position 15 of register MPAC before program control is transferred to routine ENDSCALE. If register OVFINd contains 77776 indicating underflow, routine SIGNFIX tests the contents of register MPAC and performs the following operations:

- a. If register MPAC contains a negative quantity, a ZERO is entered into bit position 15 of register MPAC and program control is transferred to routine ENDSCALE. Routine ENDSCALE, transfers program control to routine PUTCOM2 where the content of register MPAC is trans-

- ferred to register A and program control is transferred to the routine that called routine PUTCOM.
- b. If register MPAC contains 77777, the quantity 37777 is entered into register MPAC and program control is transferred to routine ENDSCALE.
  - c. If register MPAC contains 00000 the quantity 40000 is entered into register MPAC and program control is transferred to routine ENDSCALE.

17-423. Routine OPTDEGIN. Routine OPTDEGIN tests the content of register MPAC. If register MPAC contains a negative quantity, indicating negative data has been keyed in, program control is transferred to routine DSPALARM (paragraphs 17-412 through 17-414). If register MPAC contains a positive quantity, the content of bit position 13 of register WASOPSET (the sextant/telescope indicator) is tested. When this bit position contains a ONE, indicating the information was obtained via the sextant, program control is transferred to routine OPDEGIN2. When bit position 13 contains a ZERO, indicating the data was obtained via the telescope, program control is transferred to routine DMP to multiply the content of registers MPAC and MPAC+1 by the double precision quantity 26161; 30707 contained in locations DEGCON1 and DEGCON1.

This quantity represents the decimal value  $\frac{10^5}{18000} \cdot \frac{1}{2^3}$ .

17-424. Program control is then transferred, via routine SWCALL, to address SL1+3 in routine SL1 to shift the product one place to the left. This shift is necessary since optics (telescope) angles in the  $180^\circ$  range use the same scale factor as angular information in the  $360^\circ$  range. Thus, shifting the product yields the result that would have been obtained had the scale factor constant represented the value of the appropriate ratio  $\frac{10^5}{09000} \cdot \frac{1}{8}$  instead of  $\frac{10^5}{18000} \cdot \frac{1}{8}$ .

17-425. Program control is then transferred to routine DEGINSF at address DEGINSF+2 to shift the product three additional places to the left and to store the result in the location specified by the first, second, or third component of the Noun. The operation of routine DEGINSF commencing at address DEGINSF+2 is described in paragraphs 17-418 through 17-422.

17-426. Routine OPDEGIN2, executed when optics (sextant) information in the  $90^\circ$  range is entered, subtracts a bias of 19.775 degrees from the data contained in register MPAC. Since the information contained in register MPAC is in fractional form, the octal quantity 71527,



which represents the decimal quantity  $-\frac{19775}{10^5}$ , is added to the content of register MPAC. The c(MPAC+1) is not decremented. Thereafter, program control is transferred to routine DMP to multiply the content of registers MPAC and MPAC+1 by the double precision quantity 21616; 07071 contained in locations DEGCON2 and DEGCON2+1. This quantity represents the decimal ratio  $\frac{10^5}{45000} \cdot \frac{1}{4}$ . Division by four is necessary because the computer is only able to multiply by a fractional number. Upon exiting from routine DMP, the product is stored in registers MPAC, MPAC+1, and MPAC+2 with the high order part stored in register MPAC. Thereafter, the scaled product is rounded by adding the quantity 04000 to the content of register MPAC+1, ignoring the c(MPAC+2). If overflow occurs, it is added to the content of register MPAC. If overflow occurs as a result of this addition, the quantity 00001 is entered into register OVFLND. Program control is then transferred to routine DEGINSF2.

17-427. In routine DEGINSF2, the content of registers MPAC and MPAC+1 is shifted two places to the left (multiplying the scaled data by four to compensate for the  $\frac{1}{4}$  in the decimal ratio  $\frac{10^5}{45000} \cdot \frac{1}{4}$ ) and the result of this shift is transferred from register MPAC to the location specified by the first, second, or third component of the Noun. Routine DEGINSF2 was described in paragraphs 17-420 through 17-423.

17-428. Routine DPINSF. Routine DPINSF transfers program control to routine DMP to multiply the data contained in registers MPAC and MPAC+1 by the double precision scale factor constant contained in registers SFTEMP1 and SFTEMP2. Registers SFTEMP1 and SFTEMP2 contain 00006; 03240 or 00001; 30707 (refer to paragraphs 17-404 through 17-406) depending upon whether DPINSF is executed to scale time data in seconds or gyro data. Upon exiting from routine DMP, the product is contained in registers MPAC, MPAC+1 and MPAC+2 with the high order part of the product in register MPAC. The product is rounded by doubling the content of register MPAC+2 and storing the result back into register MPAC+2. If overflow or underflow occurs, the quantity 00001 or 77776 is added to the content of register MPAC+1. Should this addition produce overflow or underflow, the quantity 00001 or 77776, respectively, is added to the content of register MPAC. Overflow or underflow will not occur as a result of this second addition.

17-429. Routine DPINSF then stores the content of register MPAC+1 into the second of the two consecutive locations specified by the first, second, or third component of the Noun. (The content of register MPAC

is transferred to the first consecutive location specified when program control is transferred to routine ENDSCALE). Routine DPINSF obtains the second of the two specified locations by transferring the quantity 00000 to register A if the Noun is normal or by transferring the content of register DECOUNT if the Noun is mixed and then transferring program control to routine DPINCOM. (Register DECOUNT contains 00000, 00001, or 00002 when the first, second, or third component, respectively, is being scaled and stored.)

17-430. Routine DPINCOM adds the content of register A to the content of register NOUNADD. This sum, the address of the destination of the high order part of the scaled data, is then stored temporarily in register Q. Register NOUNADD contains the first of the two consecutive addresses specified by the Noun only if the Noun is normal. A normal Noun must be single component if it specifies double-precision data. If the Noun is mixed, register NOUNADD contains the address specified by the first component, the address specified by the second component decremented by one, or the address specified by the third component decremented by two. (Refer to paragraph 17-400.) Then the content of register MPAC+1 (the low order part of the scaled data to be stored) is transferred to the location obtained by incrementing the high order destination address contained in register Q by one. Thereafter, program control is transferred to routine ENDSCALE (paragraph 17-415), where the content of register MPAC is transferred to register A and program control is transferred back to the routine that called routine PUTCOM. The calling routine itself must store the high order part of the scaled data. Refer to table 17-32 for examples of how data is entered into the AGC.

#### 17-431. Routine LOADLV

17-432. Routine LOADLV (attachment 17-9, sheet 1) is executed only after all requested data has been entered into the location(s) specified by the Noun. This routine enters the quantity 00000 into register DECBRNCH. In addition, routine LOADLV enters the quantity 77777 into register LOADSTAT to indicate all requested data has been entered and the quantity 77754 is entered into register DSPCOUNT to block further entry to routine NUM, to block any clearing action, and to indicate an operation key must be depressed. Thereafter, routine BANK-JUMP is executed to switch program control from routine LOADLV in bank 21 to routine RECALTST in bank 24.

17-433. Routine RECALTST, also called by routines VBTERM and VBPROC, tests the content of register CADRSTOR. If the register contains a zero, indicating no Job has been put in the dormant state by routine ENDIDLE, instruction TC ENDOFJOB is executed.

TABLE 17-32  
DATA ENTRY

Keys De-pressed	Content of the X, Y or Z Data Input Registers upon depressing key ENTER	Content of Specified Location								
		Octal	Decimal							
			Fractional information (-.99994 to +.99994)	Whole information (-16383 to +16383)	Time in Hours	Time in Seconds (-999.99 to +999.99)	Angular information in degrees (+359.99 to -359.99)	Optics information in degrees (000.00 to +179.00)	Optics information in degrees (000.00 to 89.999)	Gyro information in degrees (-9.999 to +9.9999)
04000	04000; 00000	04000								
+02048	00517; 21326		00520	04000	⚠	00000; 04000	03510	07220	63313	00000; 01124
-02048	77260; 56451		77257	73777	⚠	77777; 73777	74270	⚠	⚠	77777; 76653
+99999	37777; 32604		⚠	⚠	⚠	00006; 03237	⚠	⚠	⚠	00001; 30706
-99999	40000; 45173		⚠	⚠	⚠	77771; 74540	⚠	⚠ ⚠	⚠ ⚠	77776; 47071

⚠ A time load in hours is not permitted and indicator CHECK FAIL is turned on.

⚠ Negative optics angles are not permitted and indicator CHECK FAIL is turned on.

⚠ Data load is too large and indicator CHECK FAIL is turned on.



17-434. If register CADDRSTOR contains a positive quantity, indicating a Job has been put in the dormant state by routine ENDIDLE (paragraph 17-244), routine RECALTST does the following:

- a. Executes instruction INHINT.
- b. Transfers the content of register CADDRSTOR (the reactivating address of the Job put into the dormant state by routine ENDIDLE) to register A and executes routine JOBWAKE to reactivate the Job put into the dormant state by routine ENDIDLE.
- c. Enters the quantity 00000 into register CADDRSTOR to indicate no Job is now in the dormant state.
- d. Tests the content of register LOADSTAT. At this point, register LOADSTAT contains 77777, 00001, or 00002 indicating routine LOADLV, VBTERM, or VBPROC respectively called routine RECALTST. Depending upon the routine that called RECALTST, the quantity 77775, 77777, or 77776 is transferred to register A and program control is transferred to routine RECAL2.

17-435. Routine RECAL2 adds the  $c(A)$  to the  $c(IWLOC_J)$ . (Refer to paragraph 12-25.) Register  $IWLOC_J$  contains the 12-bit awakening address (in complement form) of the Job just reactivated. Performing the addition with the  $c(A)$  negative has the effect of incrementing the awakening address by 2, 0, or 1, respectively. Thus, the awakening address is set to  $P+3$ ,  $P+1$ , or  $P+2$  (where  $P$  is the address from which the TC ENDIDLE was done) depending upon whether routine LOADLV, VBTERM, or VBPROC called routine RECALTST. Instruction RELINT is then given and program control is transferred to routine RELDSP1.

17-436. Routine RELDSP1 executes instruction INHINT and then transfers program control to routine RELDSP2 where indicator KEY RLSE is turned off by entering a ZERO into bit position 3 of register OUT1. Then, the quantity 00000 is entered into register DSPLOCK which releases program section K&D for internal use. Since the Job just reactivated has already "grabbed" program section K&D, releasing program section K&D for internal use is equivalent to releasing it for the Job just reactivated. Thereafter, instruction RELINT is given and instruction TC ENDOFJOB is executed which terminates Job CHARIN.

17-437. Proceed

17-438. Verb proceed is executed whenever an operator depresses keys VERB, 3, 3, ENTER. This Verb is primarily used in response to a load

request when the operator chooses not to supply the requested data but wishes the non-CHARIN Job to continue the execution of the routine that initiated the data load request as best it can without new data.

17-439. A secondary use of Verb Proceed is in response to an internally initiated display, coupled with a request to initiate the flashing of the VERB and NOUN displays. (An example of this may be seen in routine DSPLOAD, described in Issue 14 in subsection System Test.) It is necessary that the display be coupled with a request to initiate the flashing of the VERB and NOUN displays as this is the only way an operator knows that he has to respond. This secondary use allows the operator to key in Verb 33 (Proceed) if he feels that the data displayed is acceptable; or to key in Verb 34 (Terminate) if he feels that the displayed results are not acceptable. In some situations he can also load new data by keying in a Load Verb and Noun combination and then the data. This capability depends upon the particular internal routine that requested the display and will be discussed when the routine is described.

17-440. There is one additional situation in which an operator might want to use Verb 33 (Proceed). This situation occurs when an operator has executed a data load request, decides not to load the data and wishes to turn off the flashing of the VERB and NOUN displays.

17-441. Routine VBPROC

17-442. Depressing keys VERB, 3, 3, ENTER causes execution of routine VBPROC. This routine enters the quantity 00001 into register LOADSTAT to indicate the operator has keyed in VERB 33. Thereafter, routine VBPROC releases program section K&D for internal use by executing routine RELDSP. Routine VBPROC then terminates the flashing of the VERB and NOUN displays and transfers program control to routine RECALTST to reactivate any Job put into the dormant state by routine ENDIDLE at address P+2 where P is the address from which the TC ENDIDLE was done. The action initiated by transferring program control to address P+2 depends upon the specific routine that called routine ENDIDLE. Routine RECALTST is described in paragraphs 17-433 through 17-435.

17-443. Terminate

17-444. Verb terminate is executed when the operator depresses keys VERB, 3, 4, ENTER. An operator may take this keyboard action for any of the following reasons:

- a. To terminate Monitor activity.

- b. To respond to a data load request by indicating he does not wish to load data and he wishes that the non-CHARIN Job terminate the routine that initiated the data load.
- c. To respond to a data display coupled with the flashing of the VERB and NOUN displays by indicating the displayed data is unacceptable and the non-CHARIN Job should terminate the routine that initiated the data load. (Refer to paragraph 17-439.) This Verb is also used to terminate a routine when the data displayed is acceptable; for example, the two Jobs LGNTEST1 and PIPTEST1 (requested during the execution of Verbs 70 and 71 respectively) must be concluded with a Verb 34 (Terminate).
- d. To terminate the flashing of the VERB and NOUN displays. This situation would arise when the operator initiated the data load and then decides not to complete the load.

#### 17-445. Routine VBTERM

17-446. Routine VBTERM carries out the Terminate actions by transferring program control to routine KILMONON. Routine KILMONON enters a ONE into bit position 15 of register MONSAVE1. Thus, if or when Task MONREQ is executed, monitor activity will be terminated. (Refer to paragraph 17-362.) This is the only action within routine VBTERM related to terminating monitor activity; all other actions within this routine are concerned with the other function of the Terminate VERB. Thereafter, routine VBTERM enters the quantity 77776 into register LOADSTAT to indicate the terminate Verb has been keyed in. Program control is then transferred to routine RELDSP to release program section K&D for internal use. Then, the flashing of the VERB and NOUN displays is terminated by executing routine DSPIN1 to enter a ZERO into bit position 11 of register DSPTAB+11 and to signal T4RUPT of this change in the DSPTAB registers. Program control is then transferred to routine RECALTST to reactivate any Job put into the dormant state by routine ENDIDLE at address P+1 where P is the address from which the TC ENDIDLE was done. The action taken after the transfer depends upon the specific routine that called routine ENDIDLE. Routine RECALTST is described in paragraphs 17-433 through 17-435.

#### 17-447. REQUEST EXECUTION OF A JOB OR TASK

17-448. An operator may request the execution of a Job or a Task by utilizing Verb 20 (VBRQEXEC) or Verb 10 (VBRQWAIT), respec-

tively. If an operator wishes to request the execution of a Job, he must key in the priority associated with the Job and also the starting address of the Job. If an operator wishes to request the execution of a Task, he must key in the time delay after which the Task is to be executed and also the starting address of the Task.

17-449. Since routine VBRQEXEC expects to find the priority of the Job being requested in register DSPTEM1, the operator must depress keys VERB, 2, 1, NOUN, 2, 6, ENTER to preload register DSPTEM1 with the priority. When the VERB and NOUN displays flash, the operator depresses keys D1, D2, D3, D4, D5, ENTER. The quantity D1, D2, D3, D4, D5 contains the desired priority in bit positions 14 through 10 and contains a ONE in bit position 1 if a work area is needed for this Job or a ZERO if no work area is needed.

17-450. Since routine VBRQEXEC also expects to find the complete starting address of the Job in register NOUNADD, the operator uses a Specify Machine Address Noun (Noun Code 01-07) with Verb 20. Thus, after the operator preloads register DSPTEM1, he then depresses keys VERB, 2, 0, NOUN, 0, 1; 0, 2; ...; 0.7. When the VERB and NOUN displays flash, the operator depresses keys D1', D2', D3', D4', D5', ENTER where D1', D2', D3', D4', D5'; represent the complete starting address of the Job. Upon entering the complete Job Address, routine VBRQEXEC is executed.

17-451. Thus, if the operator wants to request the execution of a Job, he must enter four commands into the computer by the following keyboard actions:

- |   |   |                                 |
|---|---|---------------------------------|
| a. Depressing keys VERB, 2, 1,          | } | Load the<br>priority            |
| NOUN, 2, 6, ENTER                       |   |                                 |
| b. Depressing keys D1, D2, D3,          | } | Load the<br>starting<br>address |
| D4, D5, ENTER                           |   |                                 |
| c. Depressing keys VERB, 2, 0,          | } | Load the<br>starting<br>address |
| NOUN, 0, 1; 0, 2, ...; or<br>0.7, ENTER |   |                                 |
| d. Depressing keys D1', D2', D3',       | } |                                 |
| D4', D5', ENTER                         |   |                                 |

After key ENTER is depressed the fourth time, routine VBRQWAIT is executed.

17-452. Since routine VBRQWAIT expects to find the delay associated with the task contained in register DSPTEM1, the operator must first depress keys VERB, 2, 1, NOUN, 2, 6, ENTER to preload register

DSPTM1. When, as a result of this keyboard action, the VERB and NOUN displays flash, the operator depresses keys D1, D2, D3, D4, D5, ENTER where D1 through D5 represent the desired time delay before the Task is to be executed. For example, assume the operator wants the Task to be executed after 60 seconds (6000 10-msec intervals) have elapsed, he would depress keys 1, 3, 5, 6, 0 which causes the octal quantity 13560 to be displayed in R1. If the operator chooses to execute a decimal load, he would depress keys +, 0, 6, 0, 0, 0 which causes the decimal quantity +06000 to appear in R1. However, in both cases the quantity 13560 would be entered into register DSPTM1 during the execution of routine NUM. Only after the desired time delay has been entered into register DSPTM1, may Verb 10 be used successfully.

17-453. Since routine VBRQWAIT also expects to find the complete starting address of the Task contained in register NOUNADD, the operator must use a Specify Machine Address Noun (Noun code 01-07) with Verb 10. Thus, after the operator preloads register DSPTM1 with the time delay, he may then depress keys VERB, 1, 0, NOUN, 0, 1; 0, 2; ...; 0, 7, ENTER. This time, when the VERB and NOUN displays flash, the operator depresses keys D1', D2', D3', D4', D5', ENTER where D1', D2', D3', D4', D5' represent the complete starting address of the Task. The starting address is displayed in R3 as it is entered and routine VBRQWAIT is executed when key ENTER is depressed.

17-454. Thus, if the operator wants to request the execution of a Task, he must enter four commands into the computer by the following keyboard actions:

- |   |   |                                 |
|---|---|---------------------------------|
| a. Depressing keys VERB, 2, 1,<br>NOUN, 2, 6, ENTER                 | } | Load the<br>time delay          |
| b. Depressing keys D1, D2, D3,<br>D4, D5, ENTER                     |   |                                 |
| c. Depressing keys VERB, 1, 0,<br>NOUN, 0, 1; 0, 2...; 07,<br>ENTER | } | Load the<br>starting<br>address |
| d. Depressing keys D1', D2', D3',<br>D4', D5', ENTER                |   |                                 |

After key ENTER is depressed the fourth time, routine VBRQWAIT is executed.

## 17-455. Routine VBRQEXEC

17-456. Routine VBRQEXEC (figure 17-14) simulates one of the following calling sequences depending upon whether or not a work area is required by the Job to be scheduled:

## a. Work area not required

L	CAF	JOBPR
L+1	TC	NOVAC
L+2	CADR	JOBADR
L+3		Control returns here after execution of routine NOVAC

## b. Work area required

L	CAF	JOBPR
L+1	TC	FINDVAC
L+2	CADR	JOBADR
L+3		Control returns here after execution of routine FINDVAC

These calling sequences are described in detail in paragraphs 12-21 and 12-22.

17-457. Routine VBRQEXEC enters instruction TC ENDOFJOB into register PROGREG+2 which is the equivalent of location L+3. Then, it executes routine RELDSP (paragraph 17-189) to release program section K&D for internal use by entering 00000 into register DSPLOCK, turning off indicator KEYRLSE, and reactivating any Job put in the dormant state by routine NVSUBSY1. Thereafter, the content of bit position 1 of register DSPTM1 is tested (paragraph 17-438). If bit position 1 contains a ONE, indicating a work area is needed, instruction TC FINDVAC is entered into register PROGREG which is equivalent to location L+1. If bit position one contains a ZERO, instruction TC NOVAC is entered into register PROGREG. Then, the content of register NOUNADD (the Job address) is transferred to register PROGREG+1 which is the equivalent of location L+2, the priority in bit positions 14 through 10 of register DSPTM1 is transferred to register A, and instruction TC PROGREG is executed. Thus, the simulation is complete as shown:

L	CAF	JOBPR
L+1	TC	NOVAC or TC FINDVAC



L+2	CADR	JOBADR
L+3		Control returns here after execution of routine NOVAC or FINDVAC

where

c(A)	= Priority of the Job
c(PROGREG)	= TC NOVAC or TC FINDVAC
c(PROGREG+1)	= Complete Address of Job
c(PROGREG+2)	= TC ENDOFJOB

17-458. After routine NOVAC (paragraphs 12-26 through 12-31) or routine FINDVAC (paragraphs 12-33 through 12-35) has requested the execution of the specified Job, program control is transferred back to location PROGREG+2 which contains instruction TC ENDOFJOB. This terminates the CHARIN Job that began with the fourth ENTER (paragraph 17-451).

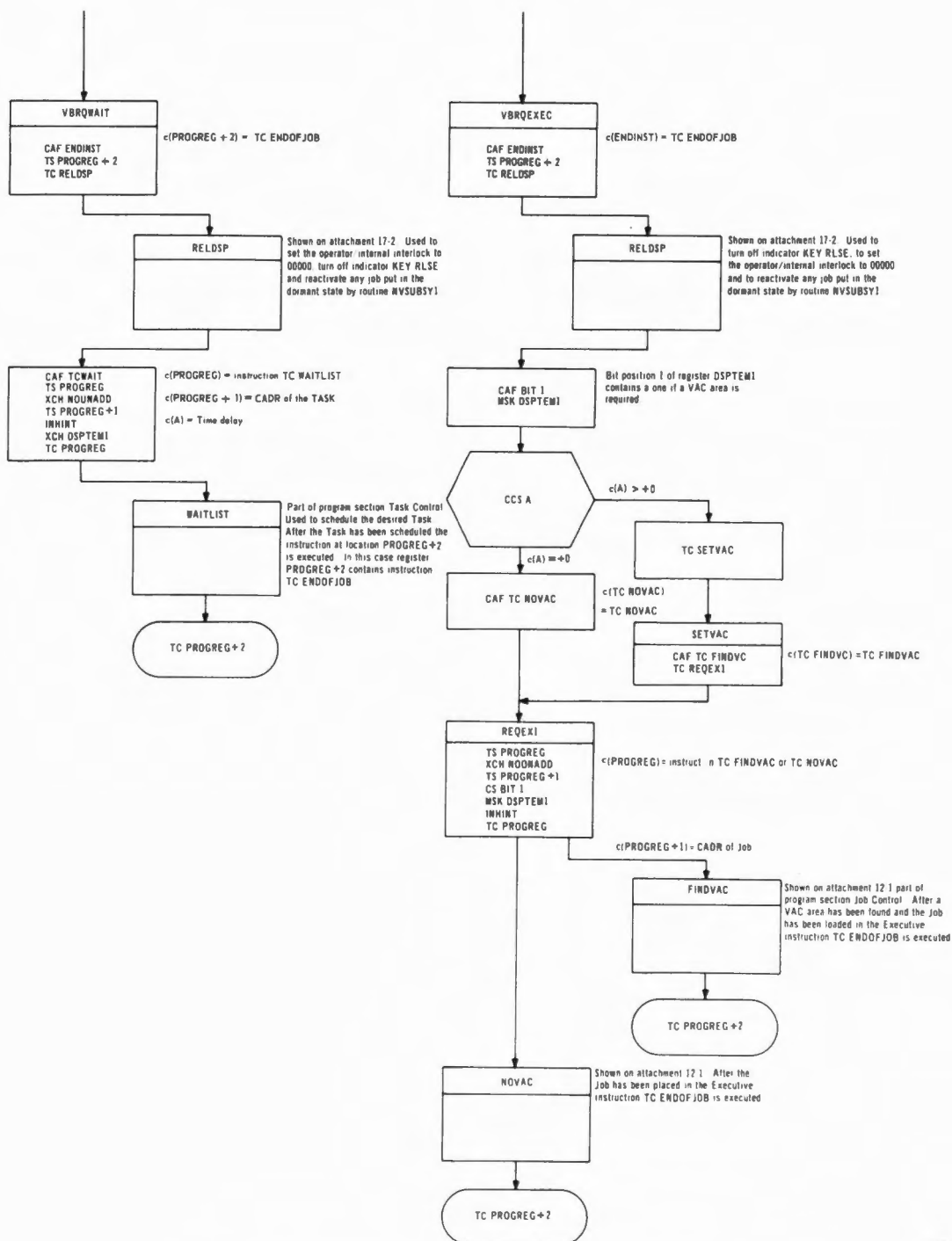
17-459. Routine VBRQWAIT

17-460. Routine VBRQWAIT (figure 17-14) simulates the following calling sequence:

L	CAF	DELTAT
L+1	TC	WAITLIST
L+2	CADR	TASKADR
		Control returns here after execution of routine WAITLIST

This calling sequence is described in detail in paragraph 12-87. It is important to note the desired time delay must be contained in register A upon executing instruction TC WAITLIST and the complete address of the Task must be available in location L+2 where L+1 is the location from which the TC WAITLIST is done.

17-461. Routine VBRQWAIT enters instruction TC ENDOFJOB into register PROGREG+2 which is the equivalent of location L+3. It then executes routine RELDSP (paragraph 17-189) to release program section K&D for internal use by entering 00000 into register DSPLOCK, turning off indicator KEY RLSE, and reactivating any Job put in the dormant state by routine NVSUBSY1. Instruction TC WAITLIST is entered into register PROGREG (the equivalent of location L+1) and the content of register NOUNADD which is the Task Address (paragraph 17-452) is transferred to register PROGREG+1 (the equivalent of location L+2).



1198

Figure 17-14. Routines VBRQWAIT and VBRXEC



Before instruction TC PROGREG is executed, the content of register DSPTEM1, which is the time delay associated with the Task (paragraph 17-443), is transferred to register A and simulation is complete.

L	CAF	DELTAT
L+1	TC	WAITLIST
L+2	CADR	TASKADR
L+3		Control returns here after execution of routine WAITLIST

where

c(A)	= Time Interval
c(PROGREG)	= TC WAITLIST
c(PROGREG+1)	= Complete Address of the Task
c(PROGREG+2)	= TC ENDOFJOB

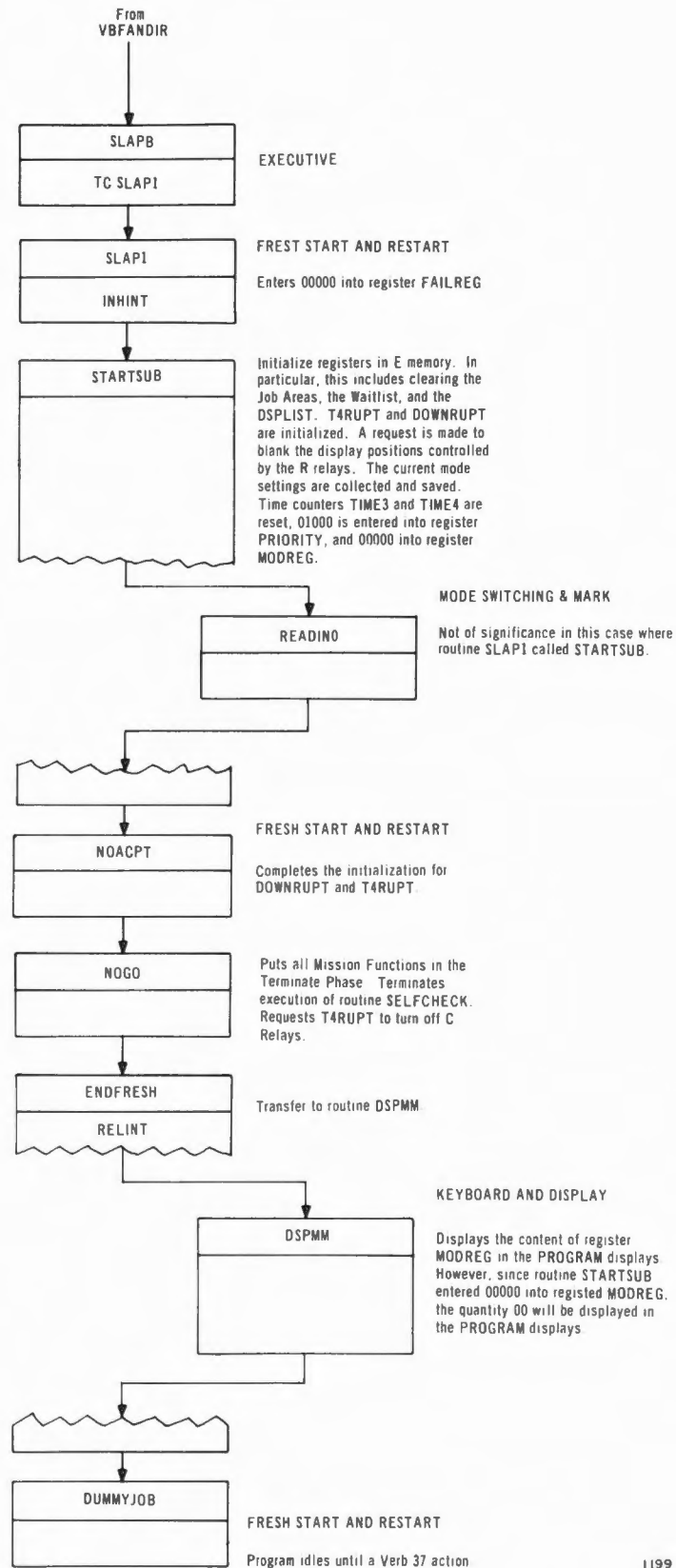
17-462. After routine WAITLIST has scheduled the desired Task (paragraphs 12-84 through 12-99), control is transferred back to location PROGREG+2 into which instruction TC ENDOFJOB has been entered. This terminates the CHARIN Job that began with the fourth ENTER.

#### 17-463. RE-INITIALIZE THE SYSTEM

##### 17-464. Routine SLAPB

17-465. Sometimes it is necessary to stop computer activity, initialize the system, and start anew, as in the case of repeated alarms or in the case where it becomes necessary to abruptly change Mission Functions. Verb 36, Fresh Start, is provided for this purpose (figure 17-15). Routine SLAP1, a part of program section Fresh Start and Restart is implemented to reinitialize the system. This routine is executed via SLAPB in the Executive whenever the operator depresses keys VERB, 3, 6, ENTER. Routine SLAP1 is the only other routine besides routine ERROR that enters the quantity 00000 into register FAILREG. Since routine SLAP1 is not a part of program section K&D, it will be described only in functional terms; a detailed description will be provided in a forthcoming issue on program section Fresh Start & Restart.

17-466. Before functionally describing routine SLAP1, it is necessary to introduce the concept of a Mission Function and the Phase of a Mission Function. (Note: These are not to be taken as definitions, but rather as descriptions of their characteristics).



1199

Figure 17-15. Verb 36 (Fresh Start)

Fundamentally, a Mission Function is a Job; however, not all Jobs are Mission Functions. A Mission Function (PRELAUNCH, for example) differs from other Jobs (CHARIN, for example) in that the routines which comprise the Mission Function are subdivided into groups of routines, according to the function performed by the routines. A number, called the Phase Code, is associated with each group of routines within the Job that constitutes the Mission Function. (Inactivity, the dormant state of a Mission Function, is also referred to as a Phase.) For example, Phase 11 of Mission Function Orbital Integration consists of routines that are executed to compute the position and velocity of the spacecraft. Loosely speaking, Phases can be considered as intermediate stages in the execution of a Mission Function. If an error occurs during the execution of a Mission Function, it can be said to have occurred during a specific Phase of that Mission Function.

17-467. If an error is such that execution of the Mission Function should not or cannot proceed, the Phase which the Mission Function was in at the time the error occurred affords a restarting point, thus eliminating the need to restart the Mission Function at the beginning. Mission Functions are processed by program section Progress Control; this processing includes reference to the Phase tables which are registers in E memory containing Phase information relative to all Mission Functions.

17-468. As discussed in paragraph 17-465, Fresh Start, executed by routine SLAP1, is provided for the situation in which an error occurs and restarting (via a GOJAM) a Mission Function at the beginning of the current Phase results in repeated alarms. If the error is in the hardware, function Fresh Start will not correct the error. However, if the error is the result of an abort or alarm condition, there is a chance that reinitializing the system could correct the error.

17-469. In general, reinitializing the system may be thought of as destroying the contents of Erasable Memory (i. e., stored data, requests for the execution of Jobs, Tasks scheduled for execution, or Mission Functions in progress). Routine SLAP1 terminates the present operations by transferring program control to routine DUMMYJOB. (During the execution of routine SLAPB, register PRIORITY is set to 01000.) Since all Mission Functions are put into the terminate Phase, a Verb 36 (Fresh Start) keyboard entry must be followed by Verb 37 (Change Major Mode) or the computer will continue to idle indefinitely. Although Verb 37 is called Change Major Mode, the name is misleading since, under these circumstances, the function of Verb 37 is to initially request the execution of a Mission Function at a specified

phase. Verb 37 is discussed in paragraph 17-474.

#### 17-470. Routine DSPMM

17-471. Routine DSPMM, utilized by routine SLAPB (figure 17-15), is shown in detail in (figure 17-16) as it is a part of program section K&D. Routine DSPMM displays the contents of register MODREG in the DSKY PROGRAM displays. The operation of routine DSPMM is similar to that of routines UPDATVB and UPDATNN, previously described in paragraphs 17-233 through 17-236. The main difference is that register DSPCOUNT is set to 00025 (instead of 23 or 21) and the contents of register MODREG (instead of register VERBREG or NOUNREG) are transferred to register A before program control is transferred to routine DSP2BIT.

17-472. Routine DSP2BIT obtains the character code for the content of bit positions 6 through 4 and 3 through 1 of register A (in this case, the bits correspond to requested Mission Functions contained in register MODREG) and then executes routine DSPIN to place the character codes in the proper positions of a DSPTAB register. (For further details pertaining to routine DSP2BIT, refer to paragraph 17-234.) After the character codes have been entered into the proper positions of a DSPTAB register, the quantity 77754 is entered into register DSPCOUNT to indicate the next key to be depressed must be an operation key (not a sign or digit key) and to block any clearing action. Thereafter, program control is transferred back to the routine that called DSPMM. When the Verb is 36, routine DSPMM is called by routine ENDFRESH immediately prior to executing routine DUMMYJOB. Under these conditions, the quantity 0, 0 will be displayed in the PROGRAM displays on the DSKY's, indicating no Mission Function is active or requested.

#### 17-473. CHANGE MAJOR MODE

17-474. Verb 37, Change Major Mode, provides for the initiation of a Mission Function at a specified Phase. If the execution of the Mission Function has already been initiated, Verb 37 is used to request a change in the Phase of the Mission Function.

17-475. These requests are implemented whenever the operator depresses keys VERB, 3, 7, ENTER; D1, D2, ENTER. Digit D1 is a code corresponding to the specified Mission Function; digit D2 is the code corresponding to the particular Phase of that Mission Function. By convention, requests for the initiation of a Mission Function at a Specified Phase are the exclusive province of the operator.

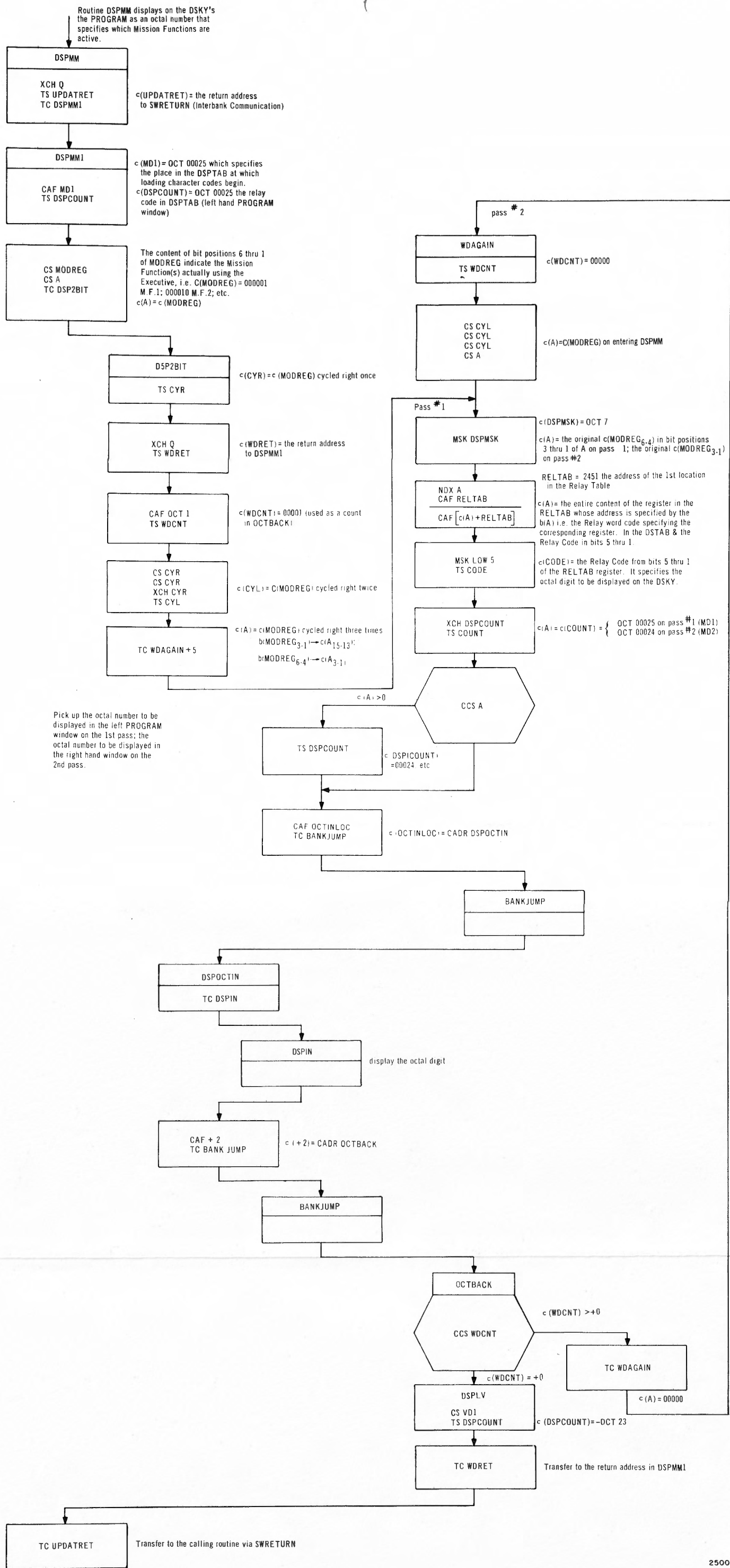


Figure 17-16. Routine DSPMM

A request to change the Phase of a Mission Function already being executed can be initiated either by operator action or by programmed action.

17-476. Since there is only one digit, D1, which can be used to specify a Mission Function, no more than eight different Mission Functions can exist within an AGC program as presently designed. In SUNRISE45 there are but two Mission Functions: the first is PRELAUNCH ALIGNMENT represented by D1 = 0; the second is ORBITAL INTEGRATION REPRESENTED BY D1 = 1. (A third function, System Test, resembles a Mission Function in that the program displays do indicate its operation, although there is no cell in the Phase table assigned to this function.) Similarly, since there is only one digit, D2, which can be used to specify a Phase, it is possible to specify only eight different Phases of a Mission Function by operator action. Since it is possible for a Mission Function to be in one of  $32_{10}$  Phases (0 through  $37_8$ ), the eight Phases that can be indicated by a keyboard action are referred to as Manual Phases while the remaining  $24_{10}$  are referred to as Machine Phases.

17-477. The initial request for the execution of a Mission Function at a specified Phase may be thought of as consisting of two actions:

- a. Entering the specified Phase code, D2, in the proper cell (defined by D1) of the Phase Tables.
- b. Requesting the execution of the Mission Function Job via a call to routine FINDVAC or NOVAC.

The starting address of the Mission Function Job is constant with respect to the specified Phase. Thus, the Mission Function Job checks the corresponding cell within the Phase Tables and commences the execution of the appropriate routines. For example, the starting address of Mission Function PRELAUNCH ALIGNMENT is STARTPL2. (PRELAUNCH is assigned a priority of 24.) Routine STARTPL2 transfers program control to Progress Control to check the Phase Tables. Upon exiting from Progress Control, the specified Phase is contained in register MPAC. Routine STARTPL2 then begins executing the routine specified by the requested Phase.

17-478. A request to change the phase of a Mission Function previously requested and still being executed may be thought of as entering the code D2 into the proper cell (defined by D1) of the Phase Tables. There is no need to request the scheduling of the Mission Function Job corresponding to digit D1 as it has already been scheduled. (If it had not been scheduled, then Verb 37, Change Major Mode, operates as described in paragraph 17-466.) Although the Phase Tables will be



modified, the Mission Function Job specified by D1 will not alter its execution until the Mission Function Job transfers program control to Progress Control to check the Phase Tables. Upon exiting from Progress Control, the Phase (D2) specified by the operator will be contained in register MPAC. The routine within the Mission Function Job, from which program control was transferred to Progress Control, will then transfer control to the appropriate functional group of routines indicated by the specified Phase. Within the functional group of routines, the Manual Phase (in the appropriate cell) will be replaced by the corresponding Machine Phase, indicating the manual request has been recognized and processed.

#### 17-479. Routine MMCHANG

17-480. Routine MMCHANG (attachment 17-10) is executed when the operator depresses keys VERB, 3, 7, ENTER. This routine transfers program control to routine REQMM to request that a two-digit code, D1, D2, where D1 represents the Mission Function and D2 represents the Phase of the Mission Function, be entered into the computer.

17-481. Routine REQMM complements the return address in register Q and enters it into register REQRET for use when the operator depresses keys D1, D2, ENTER. Thereafter, routine REQMM enters the quantity 00021 into register DSPCOUNT and the quantity 00000 into register NOUNREG. Program control is then transferred to routine 2BLANK (paragraphs 17-117 through 17-119) to blank the NOUN displays. Then, the flashing of the VERB and NOUN displays is initiated (refer to paragraph 17-269 part d), and instruction TC ENTRET is executed. Register ENTRET contains instruction TC ENDOFJOB which terminates the CHARIN Job. The exit will never be to NVSUBEND, because internal requests for phase changes are made via routine PHASCHNG in program section PROGRESS CONTROL and not through NVSUB calls with Verb 37.

17-482. When the operator depresses keys D1 and D2, the appropriate digits are displayed in display positions ND1 and ND2 respectively. Upon depressing key ENTER, program control is transferred to the continuing group of routines in subsection ENTER since register REQRET contains a negative quantity. As discussed in paragraph 17-277 the continuing group of routines terminate the flashing of the VERB and NOUN displays, recomplement the contents of register REQRET, and then execute instruction TC REQRET. Register REQRET contains address MMCHANG+1.

17-483. The actions initiated by transferring program control to



MMCHANG+1 are as follows:

- a. The quantity 00000 is entered into register NOUNREG and the b(NOUNREG), the entered code, is transferred to register MPAC.
- b. The quantity 00021 is entered into register DSPCOUNT. (Routine NUM would have decremented the previous content of register DSPCOUNT so it is necessary to reset register DSPCOUNT.)
- c. Program control is transferred to routine 2BLANK (paragraphs 17-117 through 17-119) to blank the NOUN displays.
- d. The content of register MPAC is transferred to register A and program control is transferred via routines POSTJUMP (part of Interbank Communication) and MODROUTB (part of the Executive) to routine MODROUT which is part of program section Progress Control.

17-484. Routine MODROUT, which is not a part of program section K&D, is shown in figure 17-16. Routine MODROUT, one of several entry points to program section Progress Control, decodes the c(A) and stores the requested phase into the appropriate cell of the phase table. In addition, should this be a request for a Mission Function not currently active, an Executive request is made for the Mission Function Job, the c(MODREG) is updated to reflect the newly requested Mission Function, and c(MODREG) is displayed in the DSKY PROGRAM displays. The possible displays for Sunrise45 are shown in table 17-33.

TABLE 17-33  
PROGRAM DISPLAYS

Mission Function Active or Requested	c(MODREG)	Program Displays
None	00000	00
Prelaunch Alignment	00001	01
Orbital Integration	00002	02
Prelaunch and Orbital	00003	03
System Test	00004	04
Orbital and System Test	00006	06