

**UPDATING GENERIC PROGRAM
 PERIPHERAL DATA STORAGE PROCESSOR (PDSP)
 2-WIRE NO. 1 ELECTRONIC SWITCHING SYSTEM**

CONTENTS	PAGE
1. GENERAL	1
2. INTRODUCTION	1
3. PROCEDURE	1
4. OVERWRITE	3

revision, arrows ordinarily used to indicate changes have been omitted.

1. GENERAL

1.01 This section provides the procedures required for updating the generic program for the Peripheral Data Storage Processor (PDSP) in the 2-Wire No. 1 Electronic Switching System (ESS) office.

1.02 This section is reissued to include the 2A generic. Since this reissue is a general

2. INTRODUCTION

2.01 The generic update will change the old generic to a new generic supplied by Western Electric. This may also affect the data base. If the data base format changes, the complex reallocation procedure must be followed. A concise complex reallocation procedure is shown in Fig. 1. For a detailed procedure, refer to Section 231-144-303.

3. PROCEDURE

3.01 The complete procedure for a generic update is as follows. For a concise procedure, refer to Fig. 2.

STEP	PROCEDURE
1	<p>Compare the data base in main store with data base on tape by entering the following at the maintenance terminal:</p> <p style="margin-left: 20px;">CMPR:DB;DBFILE:ERROR n!</p> <p><i>Note:</i> The n is the number of errors in a range from 1 to 10 that can occur before the comparison will end. If error is found, use local record to determine the correct data entries; then rectify, using the same RC and verification procedure at the maintenance terminal.</p>
2	<p>Enter the following:</p> <p style="margin-left: 20px;">UPD:DBFILE!</p> <p>Make sure that the DB is updated onto the cartridge.</p>

NOTICE

Not for use or disclosure outside the
 Bell System except under written agreement

STEP

PROCEDURE

-
- 3 At the system status panel (SSP), lock control unit (CU) off line by depressing the LOCK key. See Fig. 3.
- 4 At the maintenance terminal enter:
RMV:CU!

This will remove the CU from service, preventing updates when the LOCK key is released.
- 5 Place new generic cartridge into the tape data controller associated with the off-line CU and restore that TDC by entering:
RST:DEVICE;TDC x!

x = TDC with new generic tape (this is just to make sure the tape is available).
- 6 Copy the data base from the off-line CU to the new generic tape with changes to take into account the amount of generic growth by entering:
UPD:DBFILE;OFL:ADD z!

z = Amount of generic growth in decimal words.

Note 1: Address will shift with generic growth.

Note 2: Write-protect table will change to match generic shift.
- 7 At CC control panel (Fig. 4), place both processors into MANUAL state by depressing the MANUAL key.
- 8 Page in cartridge utilities by entering:
ALW:TAPEUTIL!
- 9 Physically remove cartridge in the TDC associated with the on-line CU from its transporter.
- 10 Load off-line memory with new generic and translations by entering:
LOD:OMAS;BOOT:FULL!
- 11 At SSP, release the LOCK key.
- 12 Initialize all tasks and update original on-line memory to that of off-line memory by entering:
SW:INIT:STABLE!

Note: This step will cause a system reinitialization.
- 13 At off-line CC panel, release MANUAL key.

STEP	PROCEDURE
14	Let system soak for 2 to 5 minutes.
15	At the SSP, depress the LOCK key.
16	Remove the CU from service by entering: RMV:CU
17	Place new generic cartridge into the TDC associated with the off-line CU and restore that TDC by entering: RST:DEVICE:TDC x! x = The TDC associated with the off-line CU (this is just to make sure the tape is available).
18	Place the data base onto the second new generic tape by entering: UPD:DBFILE;OFL!
19	At SSP, release LOCK key.
20	Restore the CU by entering: RST:CU;UCL!
21	The procedure for updating the backup cartridges can now be used. Four backup cartridges should be maintained. See BSP 231-144-303.

4. OVERWRITE

4.01 An overwrite is required to make a patch into those portions of the software system that are not changeable through any recent change procedure, usually the resident or nonresident generic.

4.02 The overwrite procedure used to make a patch is best illustrated by an example. In the following example, the system inputs and responses are identified.

STEP	INPUT/RESPONSE	EXPLANATION
1	Input: ALW:OW! Response: PF	Requests activation of the overwrite program.
2	Input: IN:GENID:3AP1! Response: OK	Identify the generic and issue to which the OW is to be applied.

STEP	INPUT/RESPONSE	EXPLANATION
3	Input: IN:ISSID:2A! Response: OK	
4	Input: INIT:OWFILE! Note: Perform this input one time only.	The overwrite file on cartridge tape will be reformatted. This message should only be used as a last resort if the overwrite file is unusable. It is assumed that the issue generic and issue identifiers have been entered per Steps 2 and 3. This message effectively wipes out all records of previous overwrites to the resident and nonresident programs.
5	Input: UPD:OWFILE!	The header information in the first block of the overwrite file on the cartridge tape is updated by this message. It is assumed that the new value for them has been entered prior to this request via the IN:GENID and IN:ISSID input messages.
6	Input: IN:OW 113:TTY, NO CHECK Response: WT PF	Overwrite 113 is about to be entered via the TTY. The "NO CHECK" removes the requirement for a checksum value. In Step 8, the entry will equal zero.
7	Response: IN OW COMPL	The header has been formatted in the overwrite buffer. It has been verified that the tape contains Issue 2.1 of the EF-1 generic and that it does not contain an overwrite 113. If the check fails, the response would be IN OW STOPPED.
		The standard input is IN:OWDATA:C, SEQ, ADDR, OLD DATA, NEW DATA!
8	Input: IN:OWDATA:0,5,1652,0,3254! Response: OK	The location at absolute address 1652 in segment 5 is being changed to 3254. The OK indicates that 1652 is an equipped address. If it were not, an NG response would be given. The data is entered into the overwrite buffer. If the "NO CHECK" is not used in Step 6, the "C" entry will be a given number supplied with the overwrite data. This is desirable to implement as a means of verifying the entry of the other digits in the message.
9	Input: IN:OWDATA:0,0,3456,0,1100! Response: OK	The data is entered into the overwrite buffer.
10	Input: IN:OWDATA:0,0,3457,0,1100! Response: OK	The data is entered into the overwrite buffer.

STEP	INPUT/RESPONSE	EXPLANATION
11	Input: VFY:OW:OLD! Response: PF	Retrieve the current contents of the locations identified in the buffer and compare it to the old data entered via the IN:OWDATA messages.
12	Response: VFYOW COMPL	All the data is compared. If a mismatch is detected, a VFY OW ERR message is generated to identify the error followed by the VFY OW STOPPED response.
13	Input: OP:OW;TAPE! Response: PF	Appendix the overwrite in the buffer to the end of the overwrite file on tape. The overwrite status at this point is inactive. Consequently, if a system initialization results in a program reload at this point, this overwrite will not be used.
14	Response: OP OW COMPL	The tape write was successful. The failure response is OP OW STOPPED.
15	Input: IN:OW 113; TAPE! Response: PF	This step is functionally superfluous, but is included so that the data tested is actually the data on the tape. The overwrite buffer is loaded with the contents of overwrite file 113 off-cartridge tape.
16	Response: IN OW COMPL	The tape read was successful. The failure response is IN OW STOPPED.
17	(LOCK system) Lock is a hardware function that prevents a switch from one CU to another CU. Depress the LOCK key on the SSP.	This prevents a control unit switch while the off-line store is being loaded. A switch would almost certainly be fatal because the store, containing half an overwrite, will be inconsistent.
18	Input: LOD:OW:NEW! Response: PF	Request that the new data be used to overwrite the off-line store (resident) or the off-line tape (nonresident).
19	Response: LOD OW COMPL	The overwrite operation was successful. The failure response is LOAD OW STOPPED.
20	(remove LOCK)	
21	Input: VFY:OW:NEW! Response: PF	Verify that the new contents of store matches the new data in the overwrite buffer.
22	Input: SW:CU! Response: OK	Switch to the side with the overwritten data.
23	(test data)	The test should exercise the overwritten data. In particular, the execution of an overwritten

SECTION 231-144-304

STEP	INPUT/RESPONSE	EXPLANATION
		nonresident program should be requested. If the test fails, a switch to the original side should be requested and the following LOD OW message should be qualified with OLD instead of NEW to remove the overwrite.
24	(LOCK system)	Test passed.
25	Input: LCD:OW:NEW! Response: PF	Overwrite the second store or tape unit.
26	Response: LOD OW COMPL	The overwrite operation was successful. The failure response is LOD OW STOPPED.
27	Response: VFY OW COMPL	The store load was successful. An unsuccessful load returns VFY OW ERR 0, loc.addr, store loc. content, expected content, VFY ON STOPPED.
28	(remove LOCK)	The new data is now active in the system, but a program reload initialization will still result in its elimination from store. The data still exists on the cartridge tape in the inactive state.
29	Input: ACT:OW n! Response: PF	Mark the overwrite active in the overwrite file so that subsequent program reload initializations will include this overwrite.
30	Response: ACT OW COMPL	Activation was successful. The failure response is ACT OW STOPPED.
31	Input: UPD:OW! Response: PF	Update the checksum file on tape to reflect overwrite number 113. The checksum file is updated to reflect the new overwrite using the old and new data contained in the overwrite buffer. The checksum file is used to speed up some system error recovery operations.
32	Response: UPD OW COMP	The update was successful. The failure response is UPD OW STOPPED.
33	Input: INH:OW! Response: OK	Deactivate the overwrite program.

4.03 This completes the enter overwrite procedure. Other procedures are similar in nature. To determine the content of an overwrite, perform an IN:OW n;TAPE followed by an OP:OW;TTY. To remove an overwrite, do an RMV:OW n and then follow with the enter overwrite procedure except that the LOD:OW message would be qualified with OLD instead of NEW.

4.04 The only restrictions on overwrites, other than those implicit in the procedures, are the number of overwrites and the maximum size of an overwrite. The size is limited by the overwrite buffer. The buffer must be capable of holding the header (six words) and the data blocks. The data blocks will vary in size depending on the number of consecutive words being overwritten. Overwriting an isolated word will create a block of six words. For two consecutive words, the size will be nine words. In the general case, the block length is $3 + 3n$ where n is the number of

consecutive words being overwritten. The worst case is created by overwriting isolated words. In this situation, a buffer length of 606 words would be required to enter a 100-word overwrite.

4.05 Changes larger than the maximum size will need to be split into two overwrites. In the procedure, this means that everything doubles. For example, just prior to testing, one would need to do an IN:OW n;TAPE and LOD:OW of part one of the change, followed immediately (before the switch) by an IN:OW n+1;TAPE and LOD:OW of part two.

4.06 The maximum number of overwrites is fixed by the size of the overwrite file. It is not a constant because more small overwrites can be accommodated than large ones. The sum of the lengths of all the overwrites must be less than the length of the overwrite file.

SECTION 231-144-304

ON-LINE PROCESSOR
(INITIALLY)

OFF-LINE PROCESSOR
(INITIALLY)

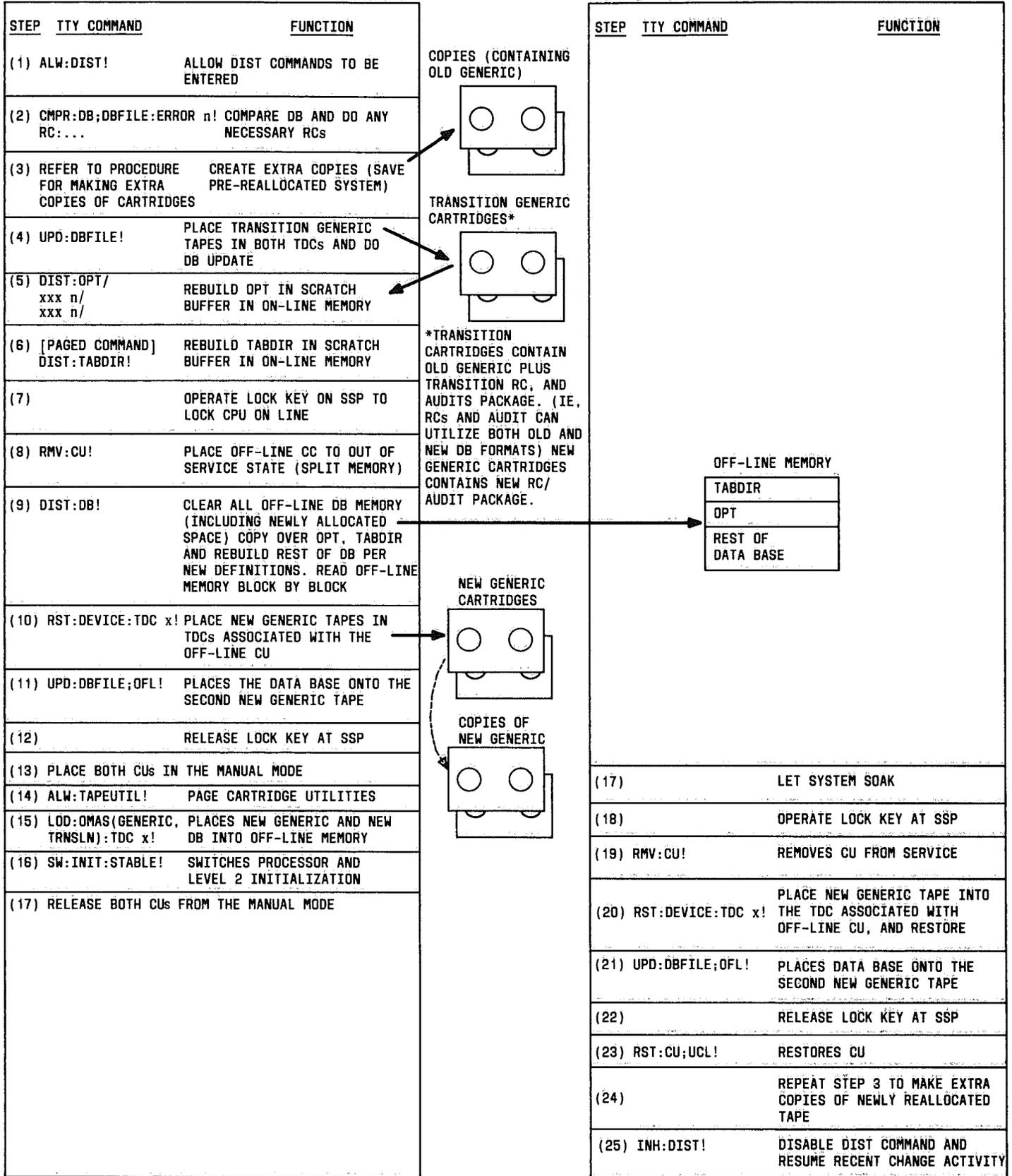


Fig. 1—Complex Reallocation

STEP	PROCEDURE																								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">OLD GENERIC</td> <td style="width: 50%; text-align: center;">NEW GENERIC</td> </tr> <tr> <td style="text-align: center;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">//////////////////DATA BASE//////////////////</td> </tr> <tr> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">//////////////////</td> </tr> <tr> <td style="text-align: center;">***G***</td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </table> </td> <td style="text-align: center;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="3"></td> </tr> <tr> <td style="width: 50%;"></td> <td style="width: 10%; text-align: center;">//</td> <td style="width: 40%;"></td> </tr> <tr> <td style="text-align: center;">***G***</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table> </td> </tr> </table>	OLD GENERIC	NEW GENERIC	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">//////////////////DATA BASE//////////////////</td> </tr> <tr> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">//////////////////</td> </tr> <tr> <td style="text-align: center;">***G***</td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </table>	//////////////////DATA BASE//////////////////			//////////////////	***G***				<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="3"></td> </tr> <tr> <td style="width: 50%;"></td> <td style="width: 10%; text-align: center;">//</td> <td style="width: 40%;"></td> </tr> <tr> <td style="text-align: center;">***G***</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>					//		***G***					
OLD GENERIC	NEW GENERIC																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">//////////////////DATA BASE//////////////////</td> </tr> <tr> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">//////////////////</td> </tr> <tr> <td style="text-align: center;">***G***</td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </table>	//////////////////DATA BASE//////////////////			//////////////////	***G***				<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="3"></td> </tr> <tr> <td style="width: 50%;"></td> <td style="width: 10%; text-align: center;">//</td> <td style="width: 40%;"></td> </tr> <tr> <td style="text-align: center;">***G***</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>					//		***G***									
//////////////////DATA BASE//////////////////																									
	//////////////////																								
G																									
	//																								
G																									
1	Enter: CMPR:DB;DBFILE:ERROR n!																								
2	UPD:DBFILE!																								
3	Depress LOCK at SSP.																								
4	Enter: RMV:CU																								
5	Insert new generic cartridge into tape controller.																								
6	RST:DEVICE;TDC x!x=off-line CU.																								
7	Physically remove cartridge in TDC associated with the off-line CU from its transport.																								
8	UPD:DBFILE;OFL:ADD z! z= Amount of generic growth in decimal words.																								
9	At CC control panel: depress the Manual key on both processors.																								
10	Enter: ALW:TAPEUTIL! LOD:OMAS;BOOT:FULL!																								
11	Release lock key at SSP.																								
12	Enter: SW:INIT:STABLE! (This causes a system initialization).																								
13	Release the MANUAL key on the off-line processor.																								
14	Let system soak for 2 to 5 minutes.																								
15	Depress the LOCK key.																								
16	Enter: RMV:CU!																								
17	Place new generic into TDC associated with off-line CU.																								
18	Enter: RST:DEVICE:TDC x! x= TDC associated with off-line CU.																								
19	Enter: UPD:DBFILE; OFL!																								
20	Release the LOCK Key.																								
21	Enter: RST:CU;UCL! This restores CU.																								
22	Update 4 backup cartridges. See BSP 231-144-303.																								

Fig. 2—Generic Update Concise Procedure

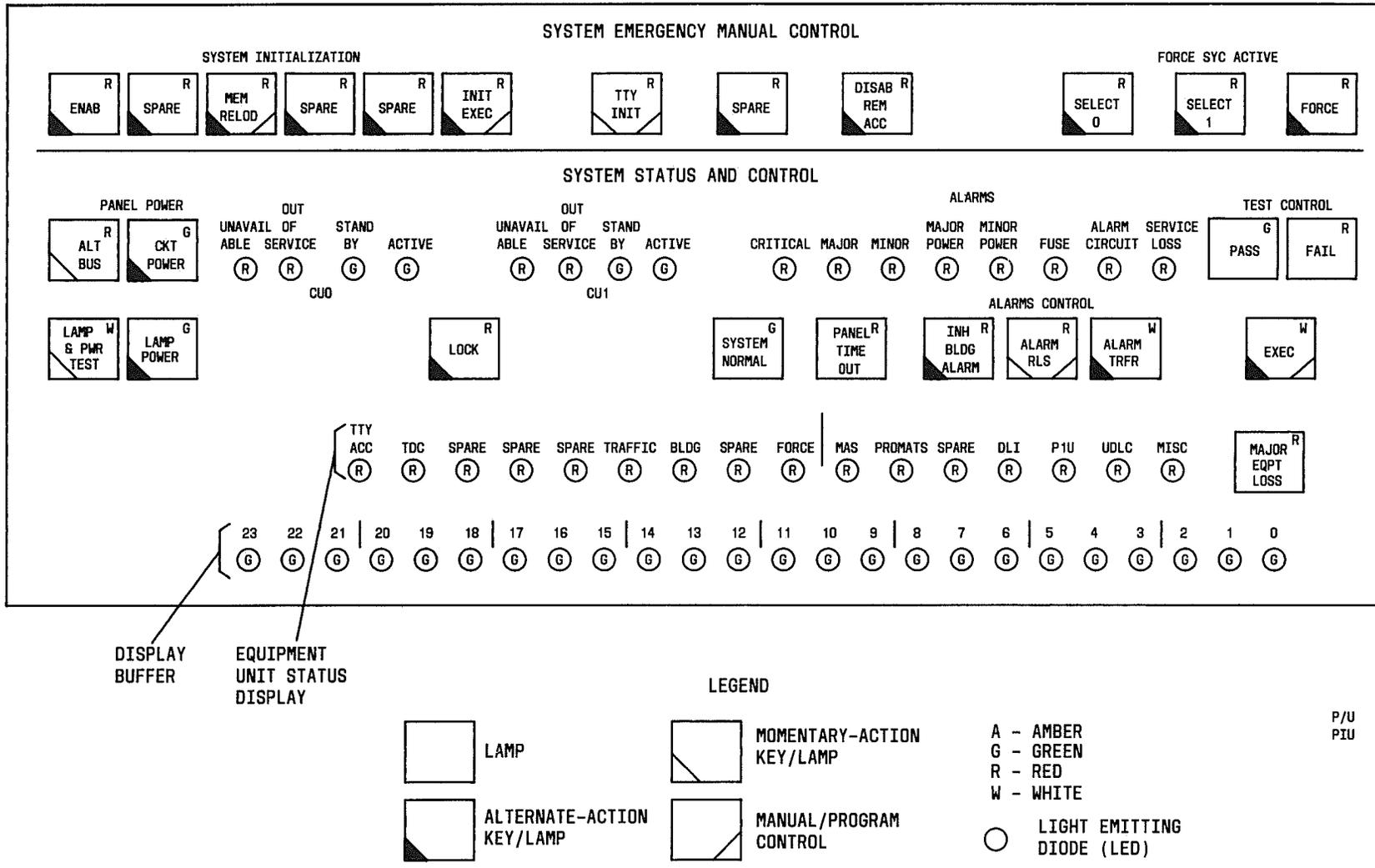


Fig. 3—System Status Panel Keys, Lamps, and LEDs



