IBM 7090/7094 IBSYS – IOEX

Programming Systems Analysis Guide

This manual was prepared by the Programming Systems Department to provide detailed information on the internal logic of the IBM 7090/7094 Basic Monitor. It is intended for technical personnel who are responsible for diagnosing the system operation, or for adapting the programming system to special usage. The charts and detailed descriptions herein are based on Version 2 of the system.
Preface

Certain knowledge is a prerequisite for the full utilization of this manual. The reader should have a basic knowledge of the 7090 FAP language. Familiarity with the information contained in the following manuals is necessary to understand the material contained in this manual:

*IBM 7090/7094 Operating Systems*, Basic Monitor (IBSYS), Form C28-6248.
*IBM 709/7090 Programming Systems: FORTRAN Assembly Program (FAP)*, Form C28-6235.
*IBM 7090 Data Processing System*, Form A22-6528.
<table>
<thead>
<tr>
<th>Description</th>
<th>Page</th>
<th>Chart</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Monitor Supervisor — IBSUP</td>
<td>7</td>
<td>AA</td>
<td>8</td>
</tr>
<tr>
<td>Trap Supervisor — IOEX</td>
<td>9</td>
<td>AB, AC</td>
<td>10, 11</td>
</tr>
<tr>
<td>Storage Map — IBSUP, IOEX, DUMP</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chart Descriptions — IBSUP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cold Start and Initialization Routine</td>
<td>14</td>
<td>BA, BB</td>
<td>36, 37</td>
</tr>
<tr>
<td>IBSYS — $IBSYS Card Routine</td>
<td>16</td>
<td>BC</td>
<td>38</td>
</tr>
<tr>
<td>IBRES — $RESTORE Card Subroutine</td>
<td>17</td>
<td>BD</td>
<td>39</td>
</tr>
<tr>
<td>Control Card Interpretation Routines —</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBCDN/IBTPIN</td>
<td>18</td>
<td>BE</td>
<td>40</td>
</tr>
<tr>
<td>IATC/IBDTC—$ATTACH/$DETACH Card Routines</td>
<td>19</td>
<td>BF</td>
<td>41</td>
</tr>
<tr>
<td>IBATC Routine</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBDTC Routine</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBXCC — $EXECUTE Card Routine</td>
<td>19</td>
<td>BG</td>
<td>42</td>
</tr>
<tr>
<td>Non-Data Select Routines</td>
<td>20</td>
<td>BH</td>
<td>43</td>
</tr>
<tr>
<td>IBREW — $REWIND Card Routine</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBWEF — $ENDFILE Card Routine</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBRUN — $REMOVE Card Routine</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miscellaneous Card Routines and Error Print-Out</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routines</td>
<td>21</td>
<td>BJ</td>
<td>44</td>
</tr>
<tr>
<td>IBSTP — $STOP Card Routine</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBDAT — $DATE Card Routine</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBCCC — $CARDS Card Routine</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBPAC — $PAUSE Card Routine</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBTCC — $TAPE Card Routine</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBLS — $LIST Card Routine</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBUNL — $UNLIST Card Routine</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miscellaneous Error Routines</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBSWC — $SWITCH Card Routine</td>
<td>22</td>
<td>BK</td>
<td>45</td>
</tr>
<tr>
<td>IBUNIT — $UNITS Card Routine</td>
<td>22</td>
<td>BL</td>
<td>46</td>
</tr>
<tr>
<td>IBREL/IBASC — $RELEASE/$AS Card Routines</td>
<td>23</td>
<td>BM</td>
<td>47</td>
</tr>
<tr>
<td>IBREL Routine</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBASC Routine</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBEDT — $IBEDT Card Routine</td>
<td>24</td>
<td>BN</td>
<td>48</td>
</tr>
<tr>
<td>Restore Nucleus, System Interrupt, and Dump</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Caller Routines</td>
<td>26</td>
<td>BP</td>
<td>49</td>
</tr>
<tr>
<td>CALIB1 — Restore Nucleus Routine</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSRPT — System Interrupt Routine</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUMP — Dump Caller Routine</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALIB3/SYSLDL — System Loader</td>
<td>27</td>
<td>BR</td>
<td>50</td>
</tr>
<tr>
<td>Disk Loader Waiting Loop and Disk Channel Routines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALPOW — Disk Loader Waiting Loop</td>
<td>28</td>
<td>BS</td>
<td>51</td>
</tr>
<tr>
<td>CALSY — Load System Record</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALSY — Write Checkpoint Record</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALSY — Load SLTABL</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALSY — Read Checkpoint Record</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Return and Print Tape Positions Routines</td>
<td>29</td>
<td>BT</td>
<td>52</td>
</tr>
<tr>
<td>IBRET — System Return Routine</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPPS — Print Tape Positions Routine</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skip Job Routines</td>
<td>30</td>
<td>BU</td>
<td>53</td>
</tr>
<tr>
<td>IBNJB Routine</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBNJJ2 Routine</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Locate Desired System and Validate Channel and</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit Designation</td>
<td>31</td>
<td>BV</td>
<td>54</td>
</tr>
<tr>
<td>IFIND — Locate Desired System Routine</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBVAL — Validate Channel and Unit Designation</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routine</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routine Description</td>
<td>Chart</td>
<td>Chart Page</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>INITUS — Initialize System Tape Usage Routine</td>
<td>33</td>
<td>BW 55</td>
<td></td>
</tr>
<tr>
<td>SYUCU — Symbolic Unit Conversion Routine</td>
<td>33</td>
<td>BX 56</td>
<td></td>
</tr>
<tr>
<td>RSYUR — Remove Unit from Availability Chain</td>
<td>35</td>
<td>BY 57</td>
<td></td>
</tr>
<tr>
<td>Routine and Sample Availability Chain</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RYSUR Routine</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample Availability Chain</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Chart Descriptions — IOEX</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTV/(ACTIV — Activate Channel Routine</td>
<td>58</td>
<td>CA, CB 75, 76</td>
<td></td>
</tr>
<tr>
<td>SAVE/XTRAP — Trap Supervisor</td>
<td>59</td>
<td>CC, CD, CE 77, 78, 79</td>
<td></td>
</tr>
<tr>
<td>BG1 — Initiate Next Channel Operation Routine</td>
<td>61</td>
<td>CF, CG 80, 81</td>
<td></td>
</tr>
<tr>
<td>Read Redundancy Recovery Routine</td>
<td>62</td>
<td>CH 82</td>
<td></td>
</tr>
<tr>
<td>Write Redundancy Recovery Routine</td>
<td>63</td>
<td>CJ 83</td>
<td></td>
</tr>
<tr>
<td>Redundancy Subroutines</td>
<td>64</td>
<td>CK 84</td>
<td></td>
</tr>
<tr>
<td>RED50 — Increment Redundancy Count Subroutine</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RED10 — Print Redundancy Error Message</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subroutine</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NRTST — Noise Record Test Subroutine</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDUN — Permit Write Redundancy Trap Subroutine</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NDSEL — Non-Data Select Routine</td>
<td>64</td>
<td>CL 85</td>
<td></td>
</tr>
<tr>
<td>Select Routine for Non-Data Operations</td>
<td>65</td>
<td>CM 86</td>
<td></td>
</tr>
<tr>
<td>(Except NDWEF)</td>
<td>67</td>
<td>CN 87</td>
<td></td>
</tr>
<tr>
<td>Select Routine for NDWEF</td>
<td>67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk Recovery Routines</td>
<td>67</td>
<td>CP, CR 88, 89</td>
<td></td>
</tr>
<tr>
<td>Issue Seek Routines</td>
<td>69</td>
<td>CS 90</td>
<td></td>
</tr>
<tr>
<td>ISSK1 — Issue Seek One Routine</td>
<td>69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISSEK — Issue Multiple Seeks Routine</td>
<td>69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machine Pause and Stop Routines</td>
<td>70</td>
<td>CT 91</td>
<td></td>
</tr>
<tr>
<td>PAUSE/PAWS Routines</td>
<td>70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP Routine</td>
<td>70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST — File Validity Test Subroutine</td>
<td>71</td>
<td>CU 92</td>
<td></td>
</tr>
<tr>
<td>Perform SNS and Form Disk Order Subroutines</td>
<td>71</td>
<td>CV 93</td>
<td></td>
</tr>
<tr>
<td>DATNS — Performs SNS Routine</td>
<td>71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDAMT — Form Disk Order Routine</td>
<td>71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-Line Print, Punch, and CVPRT Routines</td>
<td>72</td>
<td>CW 94</td>
<td></td>
</tr>
<tr>
<td>(PROUT and (PUNCH Routines</td>
<td>72</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(CVPRT Routine</td>
<td>74</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Appendix A — On-Line Messages</strong></td>
<td></td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>During IBSUP Operation</td>
<td></td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>During IOEX Operation</td>
<td></td>
<td>97</td>
<td></td>
</tr>
<tr>
<td><strong>Appendix B — Assembly Parameters</strong></td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Machine and System Definitions</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Unit Definitions</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Intermediate Definitions</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td><strong>Appendix C — Channel Information and Operation Tables</strong></td>
<td></td>
<td>102</td>
<td></td>
</tr>
<tr>
<td><strong>Appendix D — Key Switches and Storage Locations</strong></td>
<td></td>
<td>103</td>
<td></td>
</tr>
<tr>
<td><strong>Appendix E — Abbreviations</strong></td>
<td></td>
<td>107</td>
<td></td>
</tr>
<tr>
<td><strong>Appendix F — Glossary</strong></td>
<td></td>
<td>108</td>
<td></td>
</tr>
<tr>
<td><strong>Appendix G — Autochart Reference Tables</strong></td>
<td></td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>Table A — Labels</td>
<td></td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>Table B — Entry Connection References</td>
<td></td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>Table C — Subroutine Usage</td>
<td></td>
<td>113</td>
<td></td>
</tr>
<tr>
<td><strong>Appendix H — Sample Autochart Symbols</strong></td>
<td></td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>Chart</td>
<td>Description</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>AA</td>
<td>Basic Monitor, Overall Flow</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>AB</td>
<td>IOEX Main Routines, Overall Flow</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>IOEX Auxiliary Routines, Overall Flow</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Figure 1</td>
<td>Storage Map: IBUP, IOEX, DUMP</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Figure 2</td>
<td>Format of an Entry in SYSNAM Table</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>BA</td>
<td>Cold Start and Initialization, Sheet 1</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>BB</td>
<td>Cold Start and Initialization, Sheet 2</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>IBSYS Routine for $IBSYS Card</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>BD</td>
<td>IBRES Routine for $RESTORE Card</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>BE</td>
<td>Control Card Interpretation Routine, IBCDN/IBTPIN</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>BF</td>
<td>IBATC Routine $ATTACH Card</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBDTC Routine for $DETACH Card</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BG</td>
<td>IBXCC Routine for $EXECUTE Card</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>BH</td>
<td>IBREW Routine for $REWIND Card</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBWEF Routine for $SENDFILE Card</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBRUN Routine for $UNLOAD Card</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BJ</td>
<td>Miscellaneous Card Routines and Error Routines:</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBSTP/$STOP IBSWJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBDAT/$DATE IBATI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBCC/$CARDS IBAS1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBPAC/$PAUSE IBERR</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBTCC/$TAPE IBLIS/$LIST</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBUNL/$UNLIST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BK</td>
<td>IBSWC Routine for $SWITCH Card</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>BL</td>
<td>IBUNT Routine for $UNITS Card</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>BM</td>
<td>IBREL Routine for $RELEASE Card</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBASC Routine for $AS Card</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BN</td>
<td>IBDT Routine for $IBDTC Card</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RGTP0 Routine for general tape Positioning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BP</td>
<td>CALIB1 Routine for Restoring Nucleus</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSRPT Routine for Processing Interrupt Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DUMP Routine for Providing Core Dump</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BR</td>
<td>SYSLDR (CALIB3) Routine for Loading Records from System Tape or Disk</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>BS</td>
<td>Disk Loader, Wait Loop, and Channel Routines:</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CALPOW, CAL8Y, CAL5Y, and CAL6Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BT</td>
<td>IBRET Routine for System Return</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TPPS Routine for Printing Tape Positions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BU</td>
<td>IBNJ1 and IBNJ2 Routines for Processing Skip Job Request</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>BV</td>
<td>IFIND Routine for Locating and Positioning System Tape</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBVAL Routine for Validating and Locating a UCB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BW</td>
<td>INITUS Routine for Initializing System Tape Usage</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>BX</td>
<td>SYUCU Routine for Conversion of a Symbolic Unit to a Unit Address</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>BY</td>
<td>RSYUR Routine for Removing a Unit from Availability Chain Sample Chaining Layout</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Figure 3</td>
<td>Recovery Actions Taken for Specific Disk Errors</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td>ACTV Routine, Sheet 1</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>CB</td>
<td>ACTV Routine, Sheet 2</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>SAVE/XTRAP Routines, Sheet 1</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>CD</td>
<td>SAVE/XTRAP Routines, Sheet 2</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>CE</td>
<td>SAVE/XTRAP Routines, Sheet 3</td>
<td>79</td>
<td></td>
</tr>
</tbody>
</table>
IBSYS is a basic supervisory program used to coordinate a sequence of jobs which might involve distinct and independent operating systems. This coordination is achieved through the use of a common input-output trap supervisor (IOSX), and a series of Unit Control Blocks (UCBs) which contain the statistics and control information for all the I/O units in use by the system.

IBSUP is that portion of the Basic Monitor which is in control on initial loading, and between each system that is run. IBSUP consists of a group of small routines, each performing one specific function as directed by one control card that has been read in. Chart AA shows the overall flow within IBSUP. Unless a system stop or a transfer to another system is desired, control returns to read in another card and perform the function requested by this card.

The control cards are divided into four categories:

1. Operational: $EXECUTE, $IBSYS, $PAUSE, $CARDS, $TAPE, $RESTORE, and $STOP.

2. Unit Assignment, Physical Attachment: $ATTACH, $DETACH.
   Logical Assignment: $AS, $RELEASE, and $SWITCH.

3. Tape Manipulation: $SENDFILE, $REWIND, and $REMOVE.

4. Miscellaneous: $DATE, $*, $UNITS, $UNLIST, $LIST, $IBEDT, and $ID.

Operational control cards perform functions concerning the overall operation of the IBSYS system while IBSUP is in control. They indicate the location of the input control cards, provide a temporary or permanent stop, and cause restoration of the IBSYS system or read-in of another system.

Unit assignment control cards indicate which units are physically available, and which logical function is to be handled by a particular physical unit.

The tape manipulation control cards perform non-data select operations on logical units, and maintain the proper tape position indicators in the individual UCB for the unit receiving the operation.

The miscellaneous control cards perform a variety of accounting type functions, and can call in an editor to perform required system tape maintenance.

For a detailed description of the exact function of each card, refer to the IBM Reference Manual, IBM 7090/7094 Operating Systems, Basic Monitor (IBSYS), Form C28-6248.
Chart AA. Basic Monitor, Overall Flow
The IOEX portion of the Basic Monitor, shown on Charts A8 and A9, consists of a trap supervisor for data channel operations, and a series of subroutines to perform specific functions that might be required by a user. The subroutines initiate or give priority to movement of data, perform non-data select operations, convert data from one form to another, print or punch data on-line, and provide temporary or permanent halts to the program. The trap supervisor processes all data channel traps received, and maintains operation on any channel as long as activity is requested for any unit on that channel.

Communication between the user's program and the trap supervisor is through the use of Unit Control Blocks (UCB's) associated with each physical I/O unit in the system. The user indicates that an operation is requested for a unit by placing the address of a select routine in word 2 of the UCB. Later, in the select routine, the user will indicate to the trap supervisor that this operation is complete by clearing word 2 of the UCB. A complete description of each portion of the UCB with its associated meaning may be found in the IBM Reference Manual, IBM 7090/7094 Operating Systems, Basic Monitor (IBSYS), Form C28-6248.

A user may initiate activity on a channel through the use of the ACTIV subroutine. If the request does not call for priority (word 1 of the calling sequence has a prefix of MZE) and the channel is already active, no action is necessary, and control returns immediately to the user's program. If the channel is dormant, however, the user's select routine will be entered to activate the channel before returning to the user's program. If the user requests priority by using a prefix of MZE in the first word of the calling sequence, there are two possible courses of action for IOEX: (1) If the routine is entered at non-trap time, control remains in the routine, if necessary, until the desired select routine has been entered. (2) If the routine is entered at trap time, the unit is given priority by placing the location of the proper UCB in the channel priority cell, and the return to the user's program is made immediately.

If a user wishes to perform some non-data select operation on a unit, he uses the NDSEL subroutine. This initiates the operation with priority through the use of the ACTIV subroutine with a prefix of MZE in the calling sequence.

The trap supervisor first saves all machine registers and machine status conditions so that they may be restored upon exit from the routine. The channel and unit causing the trap are determined, and the I/O operation just performed is checked for errors. If errors are found, recovery is attempted in a manner prescribed by the type of error and assembly configuration. When a successful operation has been performed, the next waiting I/O operation will be initiated by the user's SELECT routine. First, a test is made to determine if one should be started on the same unit that just completed the operation. If none is required on this unit, each UCB subsequent to this one is checked to determine if a request is waiting for that unit. This test is made through the entire block of UCB's for this channel by returning to the start of the UCB's for the channel and continuing the test. In this manner, any UCB prior to the one for the unit that has just completed an operation will also be checked. The first UCB found with an address of a select routine in word 2 will have the requested operation started before leaving the trap supervisor. If no UCB requests an operation, the channel is allowed to become dormant, and will require an entry to ACTIV to start it in operation again. Before returning to the point interrupted by the data channel trap, the machine registers and status conditions previously saved are restored.

The remaining subroutines in IOEX perform specific conversions of data, on-line printing or punching of Hollerith data, or machine pauses and stops. These are explained in detail, along with their usage, in the reference manual, Form C28-6248.
Chart AB. IOEX Main routines, Overall Flow
Chart AC. IOEX Auxiliary Routines, Overall Flow
The storage map, Figure 1, shows the relative location of the various routines in the Basic Monitor and IOEX. The actual memory locations shown are not necessarily correct for all compiled systems. Minor changes in assembly parameters, which might be made by a specific user, may change many of these locations to a slight degree. The system origin SYSORG and the following three tables within the nucleus should remain the same: table of one word entries, SYSTRA to SYSTWT; system units table, SYSL81 to SYSUT4; and the IOEX communications table, CACTIX to UBEDZ.

The first column shows the contents of storage required for IOEX and the nucleus. This load is in storage during the execution of any system operating under the Basic Monitor system. The lower half of this column is an exploded view of the contents of the nucleus, and shows the locations of its cells when the system is operating. The lines that extend to the right indicate the location of the nucleus in IBSUP prior to being relocated to its operating position.

The second column shows the contents of storage when the IBSUP portion of Basic Monitor is in control. It also shows the original location of the nucleus as it is loaded from tape or disk.

The third column shows the approximate limits of the system dump program, and the short block of locations used to store the machine cells and status conditions prior to entry to the dump program.
Figure 1. Storage Map: IBSUP, IOEX, and Dump
Chart Descriptions — IBSUP

Cold Start and Initialization Routine
This routine is used for the initial loading, or whenever a RESTORE control card is read in, or a direct transfer is made to IBRES (Block BA01). Entry is at location COLD (Block BA01) for an initial load or at DRCOLD (Block BA05) for restoration.

On initial loading, a check is made for proper read-in of the program, preliminary housekeeping is performed, and the input switch is set for the proper input unit. This unit is made available if it is assigned and attached. The original value of SYSLBI and the disk limits are stored for future reference in CLDLB1 and CLDLB1 + 1.

For both initial loading and restoration, a set of Unit Control Blocks (UCB's) is generated. One block of four words is generated for each existing card machine, tape drive, disk access arm, and Hypertape drive.

The SYSUAI table is set up so that each entry will have as its address the address of the UCB for the unit assigned to that entry. After any use of the routine, the exit is to IBSYS (Block BA01).

Block BA01: Location 0 is cleared to provide a stop if a redundancy is encountered while reading in the Basic Monitor. Location COLD is also cleared to indicate that cold start has been executed.

Block BA02: All channels are tested for tape redundancies. If one is found, control transfers to location 0 and halts.

Blocks BA03 and BA04: If there are no redundancies, the nucleus is moved from the temporary area where it was initially loaded to the normal working area. The value in the entry keys is placed in location IBKEYS. All sense switches are tested and their settings stored as individual bits in location IBKSW.

Block BA05: Location SYSRORC + 101 is tested to determine if the system is loaded on disk. If this location is zero, indicating that the system is not on disk, control transfers to COLD at Block BA07.

Block BA06: If the system is on disk, SYSLBI and its associated disk limit location are set up for disk operation.

Block BA06.5: The value of SYSLBI and the disk limits (in case the original system was on disk) are stored for future reference in CLDLB1 and CLDLB1 + 1.

Block BA07: All UCB's are preset as follows: words 1 and 3 are set to MZE; words 2 and 4 are set to PZE. If the routine is being used by IBRES, word 3 is not altered, since this would destroy the file and record count.

Block BA08: M2 is set to 3, which is the maximum number of card machine UCB's possible for one channel.

Blocks BA09 and BA10: The MQ is loaded with a word containing indicators for all the card machines that are on all channels. The sense indicators are loaded from a table (AAAT) that contains indicators for the attached units and their model types.

Block BA11: The address for the first card machine on this channel is picked up for use in generating word 1 of the UCB. The sign of the MQ is tested to determine if this card machine exists. If it does not exist, no UCB is generated for this unit. The sense indicators are tested to determine if the unit is attached. If it is not attached, a 1 bit is or'ded to the accumulator in position 1 to indicate that the unit is not attached. The constructed Unit Control Word (UCW) is stored in word 1 of the UCB, and M4 is stepped to point to the start of the next UCB. The unit address is stepped for the next card machine on this channel, and the model and attachment bits are shifted for the next test. M2 is tested to determine if all card machines have been checked for this channel. If there are more to be checked, the entire operation within this block is repeated.

Block BA12: When all UCB's for card machines have been generated for this channel, the model and attachment bits are stored temporarily to allow the tape, disk, and Hypertape UCB's to be generated.

Block BA13: The number of tapes on this channel is extracted from TNCHR and placed in R1. The first tape unit address is placed in the accumulator. R1 is tested for zero to determine if there are any tapes on this channel. If there are none, control transfers to GXXDI (Block BA14).

Subroutine GXXUN is used to generate word 1 for one UCB, and R1 is again tested to determine if there are any more tape units remaining. If there are, control returns to subroutine GXXUN and the process is repeated until the UCB's for all existing tape units have been generated.

Block BA14: The sense indicators are loaded with the model type and attachment bits for disk units on
interface 1, access 0. \texttt{IR1} is set to the number of modules for interface 1, access 0. The address for the first module on interface 1, access 0 is picked up from location \texttt{DFW1}.

\texttt{IR1} is tested to determine if there are any modules with this access arm. If there are none, control transfers to \texttt{GXX05} (Block \texttt{BA15}). Subroutine \texttt{GNSXK} is used to generate a \texttt{UCB} for disk. This routine generates information for word 1 and word 3 of the \texttt{UCB}.

\texttt{IR1} is tested to determine if there are any more modules on this access arm; if there are, control returns to subroutine \texttt{GNSXK}.

\texttt{Block BA16:} The method used in block \texttt{BA14} is repeated for interface 2, access 0.

\texttt{Block BA18:} The sense indicators are loaded with the attachment bits for the Hypertape units on interface 1. \texttt{IR1} is loaded with the number of units on interface 1. The unit address for the first Hypertape unit on interface 1 is taken from location \texttt{HTW1}.

\texttt{IR1} is tested to determine if there are any Hypertape units on interface 1. If there are none, control exits immediately to \texttt{GXX12} (Block \texttt{BA19}). Subroutine \texttt{GXXU} is used to generate word 1 for one \texttt{UCB}. \texttt{IR1} is tested to determine if there are any more tape units on interface 1. If there are, control returns to subroutine \texttt{GXXU}.

\texttt{Block BA19:} The method used in block \texttt{BA18} is repeated for the Hypertape units on interface 2.

\texttt{Block BA20:} All unit addresses are incremented by 1000 for the next channel. The starting location for the availability chain is incremented by 1. Location \texttt{CHN} is incremented by 1 and its value left in \texttt{IR1}.

\texttt{Block BA21:} \texttt{IR1} is tested to determine if all channels have been processed. If they have not, control returns to \texttt{BDE} at block \texttt{BA08}.

\texttt{Block BA22:} When all channels have been processed, the set density instructions in \texttt{IOEX} are changed to \texttt{NOP's} to prevent any density changes until the units are referred to by \texttt{IBSYS}.

\texttt{Block BA23:} \texttt{IR2} is set to the complement of the location of the first unit in the system units table (\texttt{SYSUN}). The contents of \texttt{IR2} become a pointer to this location. \texttt{IR1} is set to the number of entries in the \texttt{SYSUN} table.

\texttt{Block BA24:} One entry in the \texttt{SYSUN} table is picked up. It is presently in the form: \texttt{PRE CHAN, UNIT. IR2} is saved.

\texttt{Block BA25:} Subroutine \texttt{IBVAL} is used to validate the channel and unit picked up in block \texttt{BA24}. On exit from \texttt{IBVAL}, the accumulator will be zero if the channel and unit are not valid. If they are valid, the accumulator will contain the location of the \texttt{UCB} for that unit. The sense indicators will contain the contents of the first word of that \texttt{UCB}, and \texttt{IR2} will be set to the complement of the accumulator address to point to the \texttt{UCB}.

\texttt{Blocks BA26 and BA27:} Position 1 of the sense indicators is tested to determine if the unit is attached. If it is, control transfers to \texttt{IVEF3} (Block \texttt{BA28}). If not, the accumulator is cleared.

\texttt{Block BA28:} \texttt{IR2} is restored and the address of the \texttt{UCB} is placed in the \texttt{SYSUN} table.

\texttt{Blocks BA29 and BA30:} Positions 5 and 12 of the sense indicators are tested to determine if the unit is disk. If it is, control transfers to Block \texttt{BA31}. If the unit is not disk, subroutine \texttt{BSYUR} removes this unit from the availability chain.

\texttt{Block BA31:} \texttt{IR1} is restored to point to the present \texttt{SYSUN}. \texttt{IR1} is saved because it is used for subroutine linkage in Block \texttt{BA22}.

\texttt{Blocks BA32 and BA33:} Subroutine \texttt{INITUS} is used to initialize system tape usage. \texttt{IR1} is restored and \texttt{IR2} is decremented to point to the next entry in the \texttt{SYSUN} table.

\texttt{Blocks BA34 and BB01:} \texttt{IR1} is tested to determine if all units have been processed. If they have not all been processed, control returns to \texttt{IVEF1} (Block \texttt{BA21}). When all units have been processed, the set density instructions in \texttt{IOEX} are restored.

\texttt{Blocks BB02 and BB03:} The \texttt{SYNSAM} table is moved from the area where it was loaded to the normal working area. All entries in this relocated table are rearranged to fit the format that will be used by \texttt{IBSYS}. (See Figure 2.)

\texttt{Block BB04:} If cold-start has been entered from \texttt{IBRES}, control now returns to \texttt{IBSYS}; if not, control passes to Block \texttt{BB05}.

\texttt{Block BB05:} The input switch is cleared to indicate that control is to come from cards. Sense switch 1 is tested; if it is down, control goes directly to Block \texttt{BB06}. If sense switch 1 is up, the input switch is set to indicate that the input is to come from tape, and control passes to Block \texttt{BB06}.

\texttt{Block BB06:} The designation for \texttt{SYSCMD} is picked up. The input switch is then tested to determine whether card or tape is being used for input. If card input is

Original format as loaded with \texttt{IBSUP}:

\begin{verbatim}
BCI 1, SYNSAM
PZE TFILES, INDEX, NFILES
\end{verbatim}

Altered format after COLD START:

\begin{verbatim}
BCI 1, SYNSAM
PZE INDEX, NFILES
INDEX 1 to 4 to indicate SYSL01 to SYSL04
NFILES Number of files \texttt{IBSUP} must skip before scatter loading the first record of the next file.
TFILES Total number of files in the system named.
\end{verbatim}
indicated (input switch is zero), control goes immediately to Block BB07. If tape is indicated (input switch is not zero), the designation for SYSIN1 is picked up and control passes to Block BB07.

**Block BB07**: The input switch is tested again; if tape is specified, control goes immediately to Block BB08. If card input is specified, the interrupt switch is set, and control passes to Block BB08.

**Block BB08**: The accumulator contains either the location of the VCB for SYSIN or SYSIN1, depending upon the setting of the interrupt switch. This location is placed into m2 in complement form to act as a pointer to the VCB. m2 is tested to determine if there is any unit assigned. If not, control transfers to Block BB10. If any unit is assigned, the sense indicators are loaded with the contents of the VCB and tested to determine if the unit is attached. If the unit is not attached, control transfers to Block BB10.

**Block BB09**: Because the unit is assigned and attached, the sign position of the sense indicators is reset to 0 to indicate that the unit is available. The sense indicators are replaced into word 1 of the VCB. The sign position of word 3 of the VCB is cleared to indicate a positive tape position, and control transfers to INJ (Block BU01).

**Blocks BB10 to BB12**: Sense switch 1 is tested to determine if the initial input was from cards or tape. If the switch is up, control transfers to INNOIN (Block BC15) for an on-line print-out, indicating that no assignment was made and that clarification is needed in the card reader. If sense switch 1 is down, subroutine PROUT is used to print an on-line message to indicate a complete failure. Subroutine STOP is used for a dead-stop halt from which there is no recovery.

**Blocks BB13 to BB17**: This is one entry point to a closed subroutine to generate a VCB. The initial address for a disk unit is placed in the decrement of word 1 of the VCB. The access and module are extracted from this address and expanded to BCD characters in the form "0123456789"). All BCD zeros are converted to octal 12. The BCD designation for "0123456789" is formed and stored in word 3 of the VCB. The original address which was stored in word 1 is placed in the accumulator and control passes to CYYUNX (Block BB18) to complete the generation of word 1.

**Blocks BB18 to BB23**: This is the second entry point to this closed subroutine. It is used whenever a VCB for a Hypertape unit is required and only word 1 must be generated. m2 is saved since it is also required for the subroutine exit during the operation of the subroutine. The sense indicators at this time contain the model and attachment bits from AAMT. If the unit is not attached, position 1 of the accumulator is set to 1. If the unit type is model IV, position 2 of the accumulator is set to 1.

The new word is stored in word 1 of the VCB. The tape address is placed into m2 and incremented by 1. The sense indicators are tested again to determine if the unit is attached. If it is, its address is entered into the availability chain. The model and attachment bits are shifted left one place for the next unit to be tested, and the next unit address is placed into the accumulator decrement. m2 is restored and the subroutine exits to the calling program via 1, 2.

**IBSYS — $IBSYS Card Routine**

IBSYS receives control from a cold-start operation and at the completion of all major control card routines except $ IBSTP ($STOP) and IBXEX ($EXECUTE).

The system input unit is initialized and word 2 of the VCB is set to the proper select routine for either card or tape input, as required. Subroutine AACTV in I0EX is used to start an I/O operation for this unit and the program waits for a trap. When a trap occurs, the select routine posts this operation as complete and sets up the trap-wait location to exit to the proper control card interpretation routine.

**Blocks BC01 to BC04**: m2 is set for card input, and the input switch is checked to determine which input unit is desired. If the input switch is set, m2 is reset for tape input. m4 is set to point to the VCB for the required input unit.

**Block BC05**: m4 is checked to determine if there is an input unit. If there is no input unit, control transfers to INNOIN (Block BC13).

**Blocks BC06 to BC08**: If there is an input unit, subroutine INTLOT is used to initialize system tape usage. Location IHWAT is set to transfer to itself in order to wait for the trap. m1 is set to point to word 1 of the VCB.

**Blocks BC09 to BC12**: Word 2 of the VCB is tested for zero to determine if the unit is dormant. If it is not, the program loops until it becomes dormant. When the unit becomes dormant, the decrement of word 2 of the VCB is set to the address of the tape select routine (IBFO). The address of the unit is tested to determine if it is a card machine or a tape unit. If the address is for a card machine, the decrement of word 2 is reset to the address of the card select routine (IBON). If the unit was not a card machine, control transfers directly to AACTV.

**Blocks BC13 and BC14**: Subroutine AACTV in I0EX is used to start an I/O operation on this unit, and the program waits for a trap.

**Block BC15**: This block is entered if no input unit had been specified. The MQ is loaded with the BCD
designation for SYSIN1. Control transfers to IFNDS for a print-out which states that there has been no assignment made for this unit and clarification is needed in the card reader.

**Block BC16**: IBN0 is the start of the select routine provided by IBNISYS for ACTV (in IOEX) to use in reading control cards on-line. There is also a similar routine, IBOF (Block BC35), provided for reading control cards from the off-line input tape.

The sign of the accumulator is used to tell the routine if it is being used to start an I/O operation (sign is plus), or to post the operation as complete following a channel trap (sign is minus). The sign of the accumulator is tested; if it is plus, control transfers to IBPSL (Block BC30).

**Block BC17**: This block is entered when posting completion of an I/O operation (sign of accumulator is minus). XTRAP (in IOEX) will have set the necessary bits in the indicators for the test to be made by the select routine. Position 1 of the sense indicators is tested to determine if an end-of-file was encountered on the last operation. If there was no end-of-file, control transfers to IBNCN at Block BC33.

**Block BC18**: Following an end-of-file indication, the interrupt switch (RUPSW) is tested. If RUPSW is not zero, control transfers to Block BC24.

**Blocks BC19 to BC22**: If the interrupt switch is zero, IR1 is saved and subroutine PROUT is used to print an on-line message that more control cards are needed. Subroutine PAUSE is used to provide a pause to allow the operator to reply by adding the required cards before continuing. When START is depressed, IR4 is restored and the program returns to ACTV.

**Block BC23**: This block is entered from Block BC17 when there has been no end-of-file condition in the card reader. IR2 is set to IBC0EN to indicate the routine required for interpreting the card information previously read in. Control then transfers to Block BC38.

**Block BC24**: This block is entered from Block BC18 if an end-of-file from the card reader occurs during an interrupt operation. IR2 is set to indicate that this routine is to be used instead of one of the control card interpretation routines. (IBRET causes return of control to the current system when IBWAT is next encountered.) Control then transfers to Block BC25.

**Block BC25**: This is the entry point to the select routine provided for ACTV to use in reading control cards from the off-line input tape.

The sign of the accumulator is tested; if it is plus, control transfers to IBPSL at Block BC30.

**Block BC26**: When posting is required (accumulator is minus), IR2 is set to IBTPIN to indicate the routine required to interpret the card information read in from tape.

**Block BC27**: Position 1 of the sense indicators is tested to determine if there had been an end-of-file while reading from tape. If there had been an end-of-file, control returns to ACTV.

**Blocks BC28 and 29**: The contents of IR2 are stored in IBWAT to set it for the proper exit after ACTV returns control to IBNISYS. Word 2 of the UCB is cleared, indicating that activity on this unit has ceased. Control is returned to ACTV.

**Blocks BC30 to BC33**: Entry is made here from each select routine when the sign of the accumulator was plus, indicating that a posting operation was not desired. IR2 is set to LAUCB to act as a pointer to the UCB. The IOAT command is picked up from word 2 of the UCB and placed in the ICNX table for use in reading a card image. The unit address is picked up from word 1 of the UCB, shifted right 18 places, and stored in the IN instruction for use in selecting the I/O unit required. The IN instruction is executed; then the instruction in the ICNX table is executed indirectly to start the I/O operation. Control is returned to ACTV.

---

**IBRES — $RESTORE Card Subroutine**

IBRES receives control whenever a user desires to restore the Basic Monitor to the configuration specified by the original assembly parameters. The user indicates this by placing a $RESTORE card in his deck of monitor control cards. If the user had previously changed the unit assignment of one or more units, or detached one or more units, their original assignment would be restored when this card is processed. A frequent use of this procedure would be between stacked jobs on one input tape. After each job, control would be returned to IBNISYS through use of a $IBNISYS card (or a similar card that is unique to the previous system being run), and this would be followed by a $RESTORE card. The $RESTORE card does not have to follow immediately, but should be placed before the next job that is to be run. This insures that any changes made in the system by one user will be nullified so that the next user will be working with the original system at the start of his job. The next user may then make changes or not, as he requires for his job. The nucleus is restored, and part of cold-start is used to reinitialize the system.

**Blocks BD01 to BD03**: The original value of SYSLB1 is picked up from CLELBI where it had been stored by the cold-start routine. Subroutine IBVAL is used to check the validity of this unit designation which was originally in the form FZE CHANx, UNIT. If the unit is valid, the location of the UCB for this unit is formed and this address is stored in SYSLB1. The original disk limits are picked up from CLELBI+1 and restored to
the proper disk limit location in case SYSLBL1 was on disk.

**Blocks BD04 and BD05:** IR2 is set to point to SYSLBL1 to be used by subroutine INITUS, which initializes the system tape (or disk).

**Blocks BD06 to BD08:** SYSLD0 is set to load in the system from the proper unit, and the exits from CAL1BL1 are set to return to IBRS1 instead of SYSTRA−2. Subroutine CAL1BL1 is used to restore IBSUP to core. It will use CAL0B1 (SYSLD0) if the system is on disk, and the return to IBRS1, (Block BD09) will then be from SYSLD0 instead of from CAL1BL1.

**Blocks BD09 to BD11:** Location BCLD+3 is set to NOP to prevent cold-start from destroying the tape position (word 3 of the vacu) during execution of that routine. The interrupt switch status is saved in its position in upper core along with the status of sense switch 1. The setting of the input switch is saved in location IBSNP (non-relocated . . . i.e., upper core).

**Blocks BD12 to BD14:** The nucleus, except for the cold-start location is moved to lower core. Location COLD is cleared to indicate that cold start has been executed. Location COLDX is set to transfer to IBSYS at that point in the cold-start routine. Control transfers to BRCOLD at Block BA05.

---

**Control Card Interpretation Routines — IBCDIN/IBTPIN**

This routine is entered from IBWAT after XTRAP has posted completion of a control card read-in. It has two entry points, IBCDIN and IBTPIN. IBCDIN is used if control cards are being read on-line, and IBTPIN is used if they are being read off-line. The setting of the list switch determines if the cards will be printed on-line, unless the card is a STOP, PAUSE, SLIST or an illegal card. STOP, PAUSE and SLIST cards are always printed. In the case of an illegal card, the card is always printed out along with a message indicating an illegal card; the list switch is then cleared, causing all future cards to be listed until a SUNLIST card is read.

Comment cards (\$) are printed if the list switch is on (zero) and control returns to IBSYS for another card. AID cards cause a transfer of control to the installation accounting routine, if there is one present, and eventual return to IBSYS. All other legal card types cause a transfer to a specific routine as determined by a table look-up for the matching card type. A subsequent transfer, using the index value obtained from the table look-up, is executed to select the desired routine.

**Block BE01:** This is the entry when reading control cards on-line. Subroutine MAC0D is used to convert the 12-bit Hollerith card image starting at IBMAG, into a 6-bit BCD image starting at ICD (also referred to as CARD).

**Block BE02:** Location IBNCT is tested for zero to check for a skip-job request. If IBNCT is zero, control transfers to IBNP (Block BD13) to allow processing of the control card.

**Block BE02.5:** When a skip-job request is not indicated, the list switch is tested to determine if the control cards are to be printed. This switch is normally clear (off) unless a SUNLIST card has been previously read in. If it is found to be set (on), control transfers to Block BD04 to bypass the print subroutine.

**Block BE03:** When the list switch is off, subroutine PRTOUT is used to print the control card on-line.

**Block BE04:** The first word of the card image is picked up for use in comparing for a legal operation code.

**Block BE05:** IRQ is set to the length of the control operation table and a comparison is made with one entry in this table. If a comparison is found, IRQ is used as an index into a table of transfer instructions that will transfer control to the routine needed to process that control card. If no comparison is found, the next entry is checked by using IRQ to step through the table. If IRQ goes to 1 and no comparison is found, the exit from this table look-up routine is to Block BE06 for further checking of the card type.

**Block BE06:** Initially the first two characters of the card are checked to determine if this is a comment card; i.e., \$*. If the card is a comment card, control transfers to IBSYS, Block BCO1.

**Block BE07:** If the card type was not found in the control operation table and the card is not a comments card, the first three characters are tested to determine if a request is being made for the installation ID routine, i.e., AID card. If the columns match, control transfers to SYS2DR, Block BEO10.

**Block BE08:** At this point, since there has been no comparison with any op code in the table, nor a comparison with a comment card nor a system ID request card, the card is deemed illegal. The list switch is checked to determine if the card has been printed. If it has been printed, control transfers to Block BEO11.

**Block BE09:** Because the illegal card has not been printed, the list switch is turned off and control transfers to IBTPIN (Block BE02). This action allows the illegal card to be printed during another pass through the valid card search routine. This time when Block BE08 is encountered, control transfers to Block BEO11.

**Block BE10:** This routine is entered from Block BE07 upon detection of a AID card, and is a calling sequence for the installation accounting routine. It consists of a
TSX SYSDIR, 4 followed by PZE ICARD, 12. The installation accounting routine returns control to the location following the calling sequence routine via a TRA 2, 4. Control then returns to ISSYS for another card.

Blocks BE11 and BE12: This routine is entered after detection and printout of an illegal card. Subroutine PROUT is used to print a message indicating an illegal card, and subroutine PAWS is used to provide an operator pause before returning to ISSYS.

IBATC/IBDTC — $ATTACH/$DETACH Card Routines

IBATC sets up a specific UCB to indicate that the unit is attached, and also inserts the unit into the availability chain for the proper channel unless the unit specified is a card machine. The routine checks for the validity of the unit specified, sets the proper model bit in the UCB, and stores the location of the UCB in LATUN. (LATUN is used by IBASC to assign this unit to a specific system unit.)

IBDTC sets up a specific UCB to indicate that the unit is not attached and removes the unit from the availability chain of the associated channel. The routine checks for the validity of the unit specified, and removes any reference in the SYSUIN table that points to this unit.

IBATC Routine

Blocks BF01 to BF03: This is the entry point for the IBATC routine. Subroutine SYUCU converts the symbolic designation on the card to a channel-unit designation in the form: PZE CHAN,..., UNIT. An illegal designation causes the subroutine to return to 1, 4 and control transfers to IBAT1 (Block BJ12). If the unit is legal, the subroutine returns to 2, 4. Subroutine IBVAL is then used to check the validity of the unit, and form the location of the UCB for that unit in the accumulator. The contents of the accumulator (21-35) are stored in LATUN. If the subroutine finds that the unit is not valid, the return will be made with the accumulator cleared.

Block BF04: The accumulator is checked to see if the unit is valid. If it is not valid, the accumulator is cleared and control transfers to IBAT1 (Block BJ12).

Blocks BF05 to BF07: For a valid unit, the model type is extracted from location LATUN and placed in the sense indicators. The sign position of the sense indicators is set to 1 and position 1 is set to 0 to indicate that the unit is attached and in the availability chain. The contents of the sense indicators are stored in UCB word 1.

Block BF08: Positions 10 and 11 of the sense indicators are checked to determine if the unit is a card machine. If a card machine is indicated, control transfers to ISSYS to read in another control card.

Blocks BF09 to BF12: The channel number is extracted from the sense indicators, and placed in M1 in complement form. Subroutine RYSVN is used to remove the unit from the availability chain of the indicated channel if it is already entered in the chain. This is a precautionary measure to eliminate redundant references because the next step is to insert the unit at the beginning of the same chain. Control then transfers to ISSYS.

IBDTC Routine

Blocks BF13 and BF14: This is the entry point for the IBDTC routine. Subroutine SYUCU is used to convert the symbolic designation on the card to the channel-unit designation in the form: PZE CHAN,..., UNIT. An illegal designation causes the subroutine to return to 1, 4 and control transfers to IBAT1 (Block BJ12). A legal designation causes the subroutine to return to 2, 4. Subroutine IBVAL is used to check the validity of the unit specified and to form the location of the UCB for that unit in the accumulator.

Block BF15: The accumulator is checked to determine if the unit specified is valid. If the unit is not valid, control transfers to IBAT1 (Block BJ12).

Blocks BF16 to BF19: For a valid unit the address of the UCB is stored in LUCAC to be used in removing any references to this unit from the SYSUIN table. The sign position and position 1 of the sense indicators are set to 1 to indicate that the unit is not available and not attached. The sense indicators are then stored in word 1 of the UCB. Subroutine RYSVN is used to remove the unit from the availability chain of the proper channel.

Blocks BF20 to BF24: IB4 is set to the number of entries in the SYSUIN table. Each entry of the SYSUIN table is compared to LUCAC to determine if there had been any reference to this unit. Any entry in the table that is equal to LUCAC is deleted. Control transfers to ISSYS to read in another control card after all the entries in the system units table have been checked.

IBXCC — $EXECUTE Card Routine

IBXCC is the routine used to set up the proper system loader to read in a desired operating system and transfer control to it when it is read into storage. If SYSLB1 is on tape, the tape is positioned properly and CALIB2 is given control to load the system. Control is transferred to it if it is found and loaded without error. If SYSLB1 is on disk, the system name is compared with entries in the SYSNAM table. When an equal compare is found, the location of the system on disk is taken from this table.
Control is given to SYSLDR (CALIB3) to read in the system and transfer control to it if it is loaded without error.

**Blocks BC01 to BC03**: IM2 is set to point to SYSLB1, and the cold-start switch (COLD) is tested to determine if cold-start has been executed. If COLD is not zero, indicating that cold-start has not been executed, control transfers to Block BC04. If COLD is zero, subroutine INITUS is used to initialize system tape usage on SYSLB1 before control is passed to Block BC04.

**Blocks BC04 and BC05**: IM2 is set to point to SYSUT4 as an indication to subroutine INITUS, which is then used to initialize system tape usage for this tape.

**Block BC06**: SYSORG+101 is tested to determine if SYSLB1 is on disk. If disk operation is indicated, control transfers to DFIND at Block BG11.

**Block BG07**: When SYSLB1 is not on disk, subroutine IFIND is used to position the tape in front of the first record for the desired system. Upon return from IFIND, IM2 is set to point to the SYSUNI entry that contains the location of the UCB for the desired unit.

**Blocks BG08 to BG10**: Block BC08 is entered from Block BG07 and also from the IBEDT routine (Block BN09). IM2 and the decrement of SYSLDR are set to point to the location of the UCB. The complement of the channel number minus 1 is placed in IM1. The I/O commands in the loader (CALIB2) are set to the desired values; control then transfers to CALIB2.

**Block BG11**: The system name (disk operation) is extracted from the card image at ICARD+2 and ICARD+3 and positioned in the accumulator for comparison with entries in the DSYSSM table.

**Block BG12**: The DSYSSM table is searched for a comparison with the desired system name. The table contains two words for each system entry. The first word of each pair contains the system name in BCD. The second word is in the format: PZE 'DOBC. When a comparison is found between the desired system name and a name in the table, control transfers to Block BG13. If the system is not included in this table, control transfers to IFIND (Block BV11).

**Blocks BG13 to BG16**: The system name that has been used for comparison is stored in SYSCUR as the current system under consideration. The second word of the two-word pair is picked up from the DSYSSM table (IM1 remained set to the relative location of the desired word when a valid comparison was found), and the decrement portion of this word (DOBC) is placed in the decrement of SYSSPOS. The contents of IM4 are stored in the address of SYSSPOS. The decrement of SYSLDR is set to the location of SYSLB1 and control transfers to SYSLDR with the decrement of the TSS instruction set to 2. This indicates that the location denoted by the decrement of SYSSPOS is to be loaded.

---

**Non-Data Select Routines**

Chart IM1 shows the flow of the non-data routines used to rewind, unload or write an end-of-file on a desired system tape. If an illegal system unit name was designated, an on-line message is printed to give the operator an opportunity to correct the card or skip to the next job. If a tape is rewound, the density is set as indicated by the sign of the SYSUNI entry for that system unit.

**IBREW — $REWIND Card Routine**

**Block BH01**: This is the entry point for the $REWIND routine (IBREW). IM1 is set to 3 to indicate a rewind operation and control transfers to Block BH04.

**IBWEF — $SENDFILE Card Routine**

**Block BH02**: This is the entry point for the $SENDFILE routine (IBWEF). IM1 is set to 7 to indicate a write end-of-file operation and control transfers to Block BH04.

**IBRUN — $REMOVE Card Routine**

**Block BH03**: This is the entry point for the $REMOVE routine (IBRUN). IM1 is set to 4 to indicate an unload operation and control passes to Block BH04.

**Block BH04**: IM1 is stored in IBRN3 as part of the NDSER calling sequence. The system unit name is picked up from the card image (locations ICARD+2 and ICARD+3) and positioned in the accumulator for comparison with the SYSUNI table.

**Block BH05**: The system unit name in the accumulator is compared to the entries in the SYSUNI table until a comparison is found. If no comparison can be found, control transfers to IBAI (Block BJ13).

**Block BH06**: When the system unit name is in the SYSUNI table, theSYSUNI entry for the unit is tested to determine if the unit is assigned. If the unit is not assigned (SYSUNI entry is zero), control transfers to IBSYS (Block BC01).

**Blocks BH07 and BH08**: If the unit is assigned, the sign of word 1 of the UCB is reset to 0 to indicate that the unit is available. Subroutine NDSER is used to perform the desired function as specified in its calling sequence. This calling sequence has been previously set up at Block BH04.

**Block BH09**: IM1 is set to the complement of the location of the UCB to act as a pointer to this location. Word 3 of the UCB is picked up to find the present position of the system tape.

**Block BH10**: The tape position in the accumulator is checked to determine if the tape is unloaded (accumulator minus), or not at load point (accumulator non-zero). If either condition is present, control transfers to IBSYS (Block BC01).
Blocks BH11 and BH12: At this point (accumulator zero) the tape must be at load point, but not unloaded. This would be the result of a rewind operation. The density mode is taken from the SYSUNI entry (sign position), and placed in the NSSEL calling sequence. Subroutine NSSEL sets the density as required for this tape. Control then transfers to IBSYS (Block B001).

Miscellaneous Card Routines and Error Print-Out Routines

Chart 12 shows the general error print routine, and also seven of the shorter main routines used in monitor operation.

IBSTP — $STOP Card Routine

Blocks BJ01 to BJ02: This is the entry point for the IBSTP ($STOP) routine. It is used whenever there are no more jobs to be processed on the input tape and operation is to be terminated on the monitor system. The list switch, LSTSW, is tested to determine if the card has been printed. If LSTSW is set, indicating that control cards are not being printed, subroutine PROUT is used to print this control card. In either case, subroutine PROUT is used to print the on-line message "END OF JOB." Subroutine STOP will cause the on-line print-out "CANNOT PROCEED," and then provide a final stop. This stop is a permanent one, because depressing START will have no effect.

IBDAT — $DATE Card Routine

Blocks BJ03 and BJ04: This is the entry point for the IBDAT ($DATE) routine. The date is picked up from ICARD+2 and ICARD+3, positioned in the MQ and stored in SYSDATE. (SYSDATE is defined and referred to as SYSDAT.) Control returns to IBSYS (Block B001).

IBCCC — $CARDS Card Routine

Blocks BJ05 and BJ06: This is the entry point for the IBCCC ($CARDS) routine. The input switch, IBSENP, is cleared to set the card read-in routine to read from the on-line card reader. The interrupt switch is set and control returns to IBSYS (Block B001).

IBPAC — $PAUSE Card Routine

Blocks BJ07 to BJ07.5: This is the entry point for the IBPAC ($PAUSE) routine. The list switch, LSTSW, is tested to determine if the card has been printed. If the card has not been printed, subroutine PROUT is used to print the card on-line. In either case, subroutine PAUSE will provide the print-out "OPERATOR ACTION PAUSE," and then a halt to allow the operator to perform some manual intervention. When START is depressed, there is a second on-line print-out, "ACTION COMPLETED," and control returns to IBSYS.

IBTCC — $TAPE Card Routine

Block BJ08: This is the entry point for the IBTCC ($TAPE) routine. The input switch is set to non-zero to indicate to the control card read-in routine that the cards are to be read in from tape. Control returns to IBSYS.

IBLIS — $LIST Card Routine

Blocks BJ09 to BJ09.5: This is the entry point for the IBLIS ($LIST) routine. The list switch, LSTSW, is tested to determine if this card has been printed. If this card has not been printed, subroutine PROUT is used to print the card on-line. In either case, LSTSW is cleared (set to OFF) to indicate that control cards are to be printed on-line. This switch is normally off upon initial loading, and must be turned on by the use of the SUNLIST card. Control returns to IBSYS.

IBUNL — $UNLIST Card Routine

Block BJ10: This is the entry point for the IBUNL ($UNLIST) routine. The list switch, LSTSW, is set to non-zero (set to ON) to indicate that control cards are not to be printed on-line. Control returns to IBSYS.

Miscellaneous Error Routines

Block BJ11: This is the entry point for the error print-out routine used to print the following message: "MIXED HARDWARE SWITCH INVOLVING SYSLBI IS ILLEGAL." Control transfers to IBERR (Block BJ14) after loading the MQ with the location of the calling sequence command used to locate and print this message.

Block BJ12: This is the entry point for the error print-out routine used to print the following message: "ILLEGAL UNIT SPECIFIED." Control transfers to IBERR (Block BJ14) after loading the MQ with the location of the calling sequence command used to locate and print this message.

Block BJ13: This is the entry point for the error print-out routine used to print the following message: "ILLEGAL SYSUNI DEFINITION." Control transfers to IBERR (Block BJ14) after loading the MQ with the location of the calling sequence command used to locate and print this message.

Block BJ14: This is the entry point for the general error print routine. It is used by the following routines: IBSW1, IBAT1, IBAT1, IFIND, and IBEDT. Upon entry to this routine, the MQ must contain the location of the calling sequence command to be used by subroutine PROUT in printing the message. The list switch, LSTSW, is tested. If it is non-zero (ON), control transfers to Block BJ19 to turn the switch off prior to the transfer to IBTPIN.
Blocks BJ15 to BJ18: When the list switch is initially off, the MQ is stored in the PROUT calling sequence. Subroutine PROUT is used to print the desired message on-line with the following message on the next line: "PROVIDE CLARIFICATION AND RESTART IN THE CARD READER." Subroutine PAUSE is used to provide a temporary stop for the operator to perform the desired manual intervention. Upon depressing START, the input switch, IBSP, is cleared to simulate a SCARDS card. Control returns to IBSYS.

Block BJ19: The list switch, LSTSW, is cleared (turned off), and control transfers to IBTPIN to print the control card on-line. IBTPIN will exit to the desired control card routine again, repeating the checks made before and return to the error print-out routine a second time. On the second pass through these routines, at Block BJ14, the program will find the list switch, LSTSW, off, and control will not be transferred to Block BJ19. The duplicate passage through these routines is required to force the print-out of the control card in error when the SUNLIST card has been used.

IBSWC — $SWITCH Card Routine

IBSWC is used to switch the unit references for two system units. A check is made to determine if both the designated units are valid and also to determine if an illegal switch is being attempted. An illegal switch would be an attempt to switch a tape unit with disk or disk with tape when either unit is SYS1. Either error condition will cause a pause to allow the operator to correct the card or skip to the next job.

Blocks BK01 to BK04: Each SYSUNI designated on the card is picked up from the card and compared with the SYSUNI table. If both units were not found in the table, control would transfer to IBAS (Block BJ04) for a print-out indicating that an illegal SYSUNI was specified.

Block BK05: If both units are found in the SYSUNI table, they are compared to SYS1. This comparison or check is made in the following manner. 1R2 and 1R4 were used when checking to see if both units were valid. Each 1R is now tested to determine if the unit was found in the table at a point higher than SYS1. If neither unit specified was SYS1, control transfers to IBSW2 (Block BK10).

Blocks BK06 to BK09: Word 1 of the UCB's for both specified unit is checked to determine if a request is being made to switch SYS1 with another unit that is not of the same type. If a mixed switch is requested, control transfers to IBSW1 (Block BJ11) for an error print-out indicating this condition.

Blocks BK10 and BK11: When the unit types are the same, the entries in the disk limit table are switched using the index values contained in 1R2 and 1R4. The same values are also used to switch the UCB addresses in the SYSUNI table.

Blocks BK14 and BK15: The address of the SYSUNI entry for the first unit is computed and set in 1R2. Subroutine INITUS is used to initialize the system tape usage for that unit.

Blocks BK14 and BK15: The address of the SYSUNI entry for the second unit is computed and set in 1R2. Subroutine INITUS is used to initialize the system tape usage for that unit. Control then returns to IBSYS.

IBUNT — $UNITS Card Routine

This routine prints all the SYSUNI names with their unit assignments and assigned densities. It also lists all attached units not assigned or reserved, and all inter-system reserve units. It is used primarily to verify unit assignments after making specialized changes for a specific system run. The routine contains a series of loops to examine each entry in the SYSUNI table, and then each UCB.

Block BL01: This is the entry point for the routine. Subroutine PROUT is used to print the heading on-line.

Block BL02: 1R2 is set to count the entries in the SYSUNI table, and the first entry in this table is picked up and placed into 1R1 in complement form.

Block BL03: 1R1 is tested to determine if a unit has been assigned to this SYSUNI function. If so, control transfers to Block BL05.

Blocks BL04 and BL05: If no unit has been assigned to the referenced SYSUNI function, a word of BCD blanks is stored in temporary message storage. The BCD designation "NONE" is picked up and control transfers to URC3 (Block BL10).

Blocks BL05.5 and BL06: If a unit is assigned to the SYSUNI function, the sense indicators are loaded with word 1 of the UCB and are tested to determine if the unit is a disk unit. If it is not a disk unit, control transfers to IBMP4 (Block BL07). If the unit is a disk unit, DORC and DEND are picked up from the disk limits table and converted to BCD. These limits are stored in the print image, and the BCD designation for "THRU" is picked up before transferring control to IBMP5 (Block BL08).

Block BL07: If the referenced unit is tape, the old density designation is blanked out in the print image and the density of this unit is tested. After the test, either the image "HIDEN" or blanks are loaded into the MQ.

Blocks BL08 and BL09: The MQ is stored in the print image. This places the configuration for blanks, "HIDEN,"
or “THRU” in the image as required for a particular tape or disk unit. Word 1 of the UCB is picked up and subroutine SYNCCV is used to convert this unit address to its symbolic form.

**Blocks BL10 and BL11:** Entry is made to this block with either the symbolic designation of the unit or the BCD designation for “NONE” in the accumulator. The designation is stored in the temporary message storage area, and subroutine PROUT is used to print the completed message.

**Block BL12:** m2 is tested to determine if all entries in the SYSUNI table have been printed. If all entries have not been printed, control returns to IBUNI (Block BL02).

**Blocks BL13 and BL14:** When all entries in the SYSUNI table have been printed, subroutine PROUT is used to print a heading for the attached units not assigned or reserved. m2 is set to the number of UCB’s and MAPSW is cleared. MAPSW is used to determine if there have been any units printed in this table. It is set whenever a unit is printed.

**Block BL15:** The sense indicators are loaded with the unit status bits from word 1 of the UCB.

**Blocks BL16 and BL17:** The sense indicators are tested to determine which units are not assigned to a SYSUNI function and are attached but not reserved. If a unit does not meet these conditions, control transfers to IBMPT (Block BL28).

**Blocks BL18 to BL21:** When a unit is attached but not assigned or reserved, the unit address is loaded into the MQ from word 1 of the UCB. Subroutine SYNCCV is used to convert this address to a symbolic designation for the unit. This symbolic designation is then stored in the print image, and MAPSW is set to indicate that there has been data printed under this heading. Subroutine PROUT is used to print this symbolic designation.

**Block BL22:** This block is entered under the following conditions: (1) after a unit is printed, (2) when a unit is assigned to a SYSUNI function, (3) when a unit is reserved and not attached. m2 is tested to determine if all UCB’s have been processed. If m2 equals 4 or less, all units have been processed. If they have not, control returns to IBMPT (Block BL15) to pick up the next unit (UCB word 1).

**Blocks BL23 and BL24:** When all units have been processed, MAPSW is tested to determine if there has been any data printed under the heading. If MAPSW is not zero, indicating that data has been printed, control transfers to IBMPT to clear MAPSW. If no data has been printed (MAPSW=0), subroutine PROUT is used to print the word “NONE.”

**Blocks BL25 and BL26:** MAPSW is cleared (if set) prior to the print-out of the next heading. The contents of MAPSW are then used to determine if any data is printed under the heading. Subroutine PROUT is used to print the intersystem reserve heading.

**Block BL27:** m2 is again loaded with the number of UCB’s and the sense indicators are loaded with the unit status bits from word 1 of a UCB.

**Block BL28:** Position 3 of the sense indicators is tested to determine if the unit is an intersystem reserve unit. If the unit is not, control transfers to IBMPT (Block BL33).

**Blocks BL29 to BL32:** When a unit is an intersystem reserve unit, the unit address is loaded into the MQ from word 1 of the UCB. Subroutine SYNCCV is used to convert this to a symbolic designation which is stored in the print image. MAPSW is set to indicate that there has been printing under this heading, and subroutine PROUT is used to print the image on-line.

**Block BL33:** m2 is tested to determine if all UCB’s have been processed. If m2 is equal to or less than 4, all units have been processed. If they have not, control returns to IBMPT (Block BL27) to pick up word 1 of the UCB for the next unit.

**Blocks BL34 and BL35:** When all units have been processed, MAPSW is tested to determine if there has been a print-out under this heading. If one or more units have been printed, control transfers to IBYS. If no units have been printed, subroutine PROUT is used to print the word “NONE,” and control returns to IBYS.

**IBREL/IBASC — $RELEASE/$AS Routines**

Chart SM shows two routines which differ only in the entry point and one operation. IBASC is used to assign the last attached unit specified in location LATUN to a specific system unit function. IBREL is used to release the specified system unit.

In IBREL, the unit previously assigned to that system unit function is entered into the availability chain for that channel. Because LATUN is cleared, IBREL does not assign a new unit to the SYSUNI function.

IBASC must also remove the newly assigned unit from the availability chain for the associated channel.

In both routines, a check is made for the validity of the unit specified on the card. An invalid unit will cause an error message print-out.

**IBREL Routine**

**Block BM01:** This is the entry point for the IBREL routine. Location LATUN is cleared and control passes into the IBASC routine. LATUN is the location where the designation for the last unit attached would normally be stored. When the IBASC routine is entered with this location set to zero, it causes the SYSUNI entry for the
specified function to be cleared to indicate that there is no unit assigned to that function.

**IBASC Routine**

*Block BM02:* This is the entry point for the **IBASC** routine. It is also entered from **IBREL** with location **LATUN** cleared. The system name is picked up from **ICARD+2** and **ICARD+3**, and positioned in the accumulator for comparison with the **SYNAM** table.

*Block BM03:* The **SYNAM** table is searched for an entry that will compare with the one specified on the control card. If none can be found, control will transfer to **IBAS1** (Block B13) to print an on-line message indicating an illegal **SYUNIX** function.

*Block BM04 to BM06:* If the system name is found in the **SYNAM** table, the previous assignment for this function is picked up from the **SYUNIX** table and stored in **IBASX** for later use in **IBASI** (Blocks BM24 through BM26.) The unit specified in **LATUN** is either the address of the **UCB** following a **REATTACH** card, or zero if the card presently being processed is a **RELEASE** card. This unit is placed in the **SYUNIX** table, and the sense indicators are loaded indirectly from this **SYUNIX** entry. This sets the contents of **UCB** word 1 into the sense indicators.

*Blocks BM07 and BM08:* Positions 4 and 11 of the sense indicators are tested to determine if the unit is disk. If the unit is disk, control transfers to Block BM09; if not, subroutine **BSYUN** is issued to remove the unit from the availability chain for the channel.

*Block BM09:* **ICARD+3** is loaded into the **MQ**. The density specification is extracted from the contents of the **MQ** and loaded into the accumulator.

*Blocks BM10 to BM11.5:* The density specification is compared to the **BCD** designation "L." If the specification is greater than or equal to "L," low density is indicated and the accumulator prefix is set to plus. If the specification is less than "L," high density is indicated and the accumulator prefix is set to minus.

*Block BM12:* The prefix of the accumulator is stored in the **SYUNIX** table. This will cause the **SYUNIX** entry to be minus for a high density unit and plus for a low density unit.

*Blocks BM13 and BM14:* **MB** is set to point to the **SYUNIX** entry, and subroutine **INITUS** is used to initialize the tape usage for this unit.

*Blocks BM15 and BM16:* **MB** is set to the relative location of the desired unit within the **SYUNIX** table. The sense indicators are loaded indirectly from this **SYUNIX** location to give the contents of word 1 of the desired **UCB**.

*Block BM17:* Positions 4 and 11 are again tested to determine if the unit under consideration is disk. If it is not, control transfers to **IBAS3** (Block BM24). ”

*Blocks BM18 to BM23:* When the unit is disk, the origin and number of cylinders are loaded into the **MQ** from the **BCD** designation on the control card. Location **IBASX** is cleared for use as a temporary storage cell in building up the binary equivalent of the origin and number of cylinders. The origin is converted to binary, multiplied by 40, and stored in the address of the disk limit table entry for that unit. The number of cylinders is converted to binary and multiplied by 40. The value of the last cylinder is computed and stored in the decrement of the disk limit table entry for that unit.

*Blocks BM24 to BM26:* When control is given to **IBAS3**, the contents of **IBASX** are tested for zero to determine if there had been any previous unit assigned to the system unit function. If no unit had been assigned (**IBASX** equals zero), control transfers to **IBSYS**. If a unit has been assigned, the address portion of **IBASX** is compared to all the addresses in the **SYUNIX** table to determine if the unit is still in use by any other system unit function. If the unit is still in use, control transfers to **IBSYS**. If the unit is not in use, word 1 of the **UCB** is tested to determine if it is a card machine. If the unit is a card machine, control transfers to **IBSYS**. At this point, if the unit is not a card machine, provisions are made to enter it at the top of the availability chain for the correct channel.

*Blocks BM27 to BM30:* The no-repositioning flag is reset and the sign is set to indicate that the unit is not assigned. The word is replaced in the **UCB**. The channel number is extracted from the unit address and placed in **MB** in complement form. Subroutine **BSYUN** is used to remove the unit from the availability chain. This is done to eliminate any chance of the unit appearing in the availability chain twice. The unit is placed at the beginning of the availability chain and control transfers to **IBSYS**.

**IBEDT — $IBEDT Card Routine**

The **IBEDT** routine (Chart BN) is used to call in and execute the system editor from **SYSLB1**. If the system is on disk, an **EXECUTE EDITOR** card is simulated. If the system is not on disk, the tape is positioned to the **EDITOR** and the **EDITOR** is read in. Control then transfers into the **IBXCC** routine to cause execution of the **EDITOR**. This chart also shows the select routine used by **IBEDT**.

*Block BN01:* This is the entry point for this routine. **SYSORG+101** is tested to determine if the system is on disk. If the system is not on disk, control transfers to **EDTS** (Block BN03).

*Block BN02:* If **SYSLB1** is on disk, the **BCD** designation for the **EDITOR** is placed into the card image to simulate
an EXECUTE EDITOR card and control is transferred to
IBMCC (Block BEG) for execution.

Block BN03: If SYSLST is on tape, H23 is loaded with
the complement of the location of SYSLST. This becomes an
indirect reference to the UCB for the desired system
tape. The desired tape position counter, RCT3, is inc-
cremented by 1 to indicate to routine RCTPO to set-up for
reading the next file.

Blocks BN04 to BN06: Subroutine RCTPO is used to
position the system tape to the next file on the tape.
H11 is set to the complement of the location of the UCB,
and the select word (EDIT) is stored in word 2 of the
UCB. This word contains the location of the select rou-
tine (RCT2) in the decrement and the location of the I/O
command in the address.

Blocks BN07 and BN08: Subroutine ACTIV is used to
read the first record of the file. Word 2 of the UCB is
tested to determine when the record has been read in.
Control remains in a loop until word 2 of the UCB is
cleared. When the word is finally cleared by the trap
supervisor in IOEX, control will be allowed to leave the
loop and continue to Block BN09.

Block BN09: The second word of the record read-in
is compared with "IBEDT." If there is an equal compare
(EDITOR found), control transfers to IBMCC (Block BEG).

Block BN10: If the second word of the record does
not equal "IBEDT," it is then compared to "*EOT." If
there is still an unequal comparison, control transfers
back to IBMEDT to check the next file for the EDITOR.

Block BN11: At this point the EDITOR has not been
found up to the end of the system tape and therefore
must be missing from the tape. The EDITOR name is
deposited into an error message and control transfers to
IBERR (Block BN14) for an error print-out.

Blocks BN12 and BN13: This is the entry point for
the general tape positioning routine. It is entered with
H13 set to the complement of a location which contains
the address of the UCB for the desired unit. The decre-
ment of the accumulator contains the file position num-
ber to be compared to the file position contained in the
UCB. H14 is saved, and the desired tape position is stored
in RCT3. Subroutine INITUS is used to initialize system
tape usage for this tape.

Blocks BN14 and BN15: Location RCT3 is tested to
determine if the desired tape position is zero. If it is,
subroutine NSXEL is used to rewind the tape and con-
control returns to RCT3 to exit from the routine.

Block BN16: When the desired tape position is not
zero, H11 is set to point to the location of the UCB.

Block BN17: The desired tape position in RCT3 is
compared to word 3 of the UCB, which is the actual tape
position at this time. If the tape positions are the same,
control transfers to RCT2 (Block BN23) to exit from the
routine. If the actual tape position is less, control trans-
fers to RCT4 (Block BN19).

Block BN18: Because the tape is neither at the de-
sired location nor at a point before the desired loca-
tion, it is beyond the desired location. Subroutine NSXEL
is used to backspace the tape one file, and control
transfers back to Block BN17 for another comparison.

Block BN19: If the actual tape position is less than
the desired position, word 2 of the UCB is tested to
determine if the unit is dormant. This is done to make
sure that the tape has completed any movement due to
a previous attempt to use the routine to skip forward.
The program loops within routine RCT4 until word 2 of
the UCB is set to zero by IOE. After the trap supervisor
clears word 2 of the UCB, it regains control and returns
to the testing loop. At this time, control passes to
Block BN20.

Blocks BN20 and BN21: Location RCT5 is set to pro-
vide a waiting loop when subroutine ACTIV is being used.
The select word, RCT6, is stored in word 2 of the UCB
and control is transferred to ACTIV in IOE.

Block BN22: RCT5 contains a transfer to itself to act
as a waiting loop to allow ACTIV to move the tape ahead
one record. The select routine will change the contents
of RCT5 to a transfer to RCT1 (Block BN16). When the
trap supervisor gains control at the end of the tape
movement, it returns to RCT5. Control is then trans-
ferred to RCT1 (Block BN16) to compare this tape posi-
tion with the desired tape position again.

Block BN23: After the tape is rewound (desired
position equals zero) or after the desired tape position
is found, H14 is restored and control exits to the calling
program via 1, 4.

Blocks BN24 and BN25: This is the select routine to
be used by RCTPO in forward spacing a tape one file.
H13 is set to point to the UCB for the desired unit. The
accumulator is tested to determine if IOEX is using the
routine to post the completion of an I/O operation or to
initiate one. If it is posting, the accumulator is minus,
and control is transferred to RCT5 (Block BN29).

Blocks BN26 to BN28: If an I/O operation is to be
initiated (accumulator plus), RCTX, 1 is set to zero to
indicate no redundancy checking. The location of the
I/O command is placed into RCTR, 1. The binary select
address is generated from word 1 of the UCB and stored
in the RNS instruction. The RNS and RCTR, 1 instructions
are executed. Control exits to the calling program (IOE)
via 1, 4.

Blocks BN29 to BN32: If a posting operation is indi-
cated, the tape position is picked up from word 3 of
the UCB. This position is compared with the desired
tape position in RCT3. If the actual position is less
than the desired position, control transfers back to IOE,
which will initiate another I/O operation to skip the next record on the tape. If the actual position is equal to or greater than the desired position, word 2 of the UCS is cleared to indicate that activity has ceased on this unit at this time. The waiting cell in RCPO is changed to a transfer to RCII. Control exits to IOEX via 1, 4.

Restore Nucleus, System Interrupt, and Dump Caller Routines

Chart BP shows the routine used to call MBOX into memory (CALLB). It also includes the two routines that utilize this routine in performing their functions, the system interrupt routine (SYSNRT or INTRP), and the system dump routine (SYSMSP or DUMP).

CALLB will load one record from VSLL1 if VSLL1 is not disk. If VSLL1 is disk, VSLL1B will be used to load one record. In either of these cases, the record must be in the self-loading format with the desired I/O command at the beginning of each block of data within the record. Ten attempts will be made to read the record properly. If these are all unsuccessful, the program transfers to location 0, which will have a halt-transfer to the restart point in case the operator desires to try again. The record will be checked as it is being read in (as well as at the end of the read-in), and the channel will be immediately stopped if a redundancy is encountered. This eliminates passing a long tape record across the read head when the error is found near the beginning of the record. The tape position is updated, and the program tests the dump switch, DMPSW, to determine whether to transfer control to IBSYS or return to read in the next record (SYSMP) and eventually transfer control to it.

SYSNRT checks sense switch 1; if the switch is down, it sets the interrupt switch and transfers control to CALLB to return control to IBSYS after reloading INSUP.

SYSMSP saves the present contents of core, all key registers, and status indications. The dump switch, DMPSW, is set and control is given to CALLB to load in IBSYS and eventually the system dump routine.

CALLB — Restore Nucleus Routine

Block BP01: Word 1 of the UCS for the system unit is tested to determine if the system unit is disk. If the system unit is disk, control transfers to Block BP19.

Blocks BP02 and BP03: If VSLL1 is not disk, M2 is set to point to the location of UCS. The necessary I/O commands for tape unit A1 are generated and stored in their respective locations in the routine. The system tape is then rewound and the tape position indicator in word 3 of the UCS is set to zero.

Block BP04: The channel is tested to determine if it is busy. If it is busy, the program will loop here until the channel completes the present operation.

Blocks BP05 and BP06: When the channel becomes "not busy," traps are disabled and location zero is set for the restart entry. M4 is set to 10 to allow this number of rereads after a redundant read occurs. The operator also has the option of using this routine as a restart entry point after 10 consecutive, unsuccessful attempts to read a record. An additional 10 rereads are initiated by depressing START. This operation may be repeated more than once.

Blocks BP07 to BP10: The channel is started in operation and tested for redundancies. If any are found during the read operation, control transfers immediately to Block BP16 to stop the channel. This test for redundancies is followed by a test for channel in operation. As long as the channel is in operation, and no redundancies occur, control will remain alternately with these two instructions. When the channel finally completes the operation, it is again tested for redundancies.

If redundancies are encountered after the operation is completed, control transfers to Block BP17 for possible retries at reading the record.

Blocks BP11 and BP12: When the channel operation is completed without encountering redundancies, the tape position in word 3 is incremented by 1. The dump switch, DMPSW, is tested to determine if this routine is being used by the dump routine. If it is, control transfers back to CALLB (Block BP04) to read in the dump program.

Blocks BP13 to BP15: When the dump routine is not required, the enable switch, ENSW, is tested to determine if trapping is to be enabled. If ENSW is nonzero, control transfers directly to SYSTRE. If ENSW is zero, traps are enabled before control is transferred to SYSTRA.

Block BP16: Block BP16 will be entered if a redundancy is encountered while reading in the record. The channel will be stopped immediately to eliminate wasted time in passing the rest of the record over the read head.

Blocks BP17 and BP18: Any time a redundancy is encountered, the tape will be backspaced. M4 is tested to determine if the reread has been tried 10 times. If 10 rereads have not been tried, control returns to CALL (Block BP07) for another attempt. After 10 attempts, control will transfer to location 0 and halt. The operator may now decide to attempt 10 more rereads by depressing START.

Block BP19: If VSLL1 is on disk, the decrement of SYSTCH is set to the disk origin as obtained from the
disk limits table. Control is then given to SYSLDR to load in the system from disk.

**SYSRPT — System Interrupt Routine**

*Block BP21:* Entry is made to this block by using the communication cell SYSRPT. If sense switch 1 is up, control transfers to Block BP25 to clear INTSW and exit to the calling program.

*Blocks BP22 to BP24:* If sense switch 1 is down, INTSW is tested to determine if this interrupt is to be permitted. If INTSW is non-zero, control returns to the calling program at 1, 4. If INTSW is zero, indicating that the interrupt is to be permitted, BPFSW is set. IBSUP is cleared to indicate that the input unit is to be the card reader. Control then transfers to CALIB1 to re-load IBSUP, and take the next control card from the card reader. The operator should then have cards that indicate the next course of action for the system in the on-line card reader.

*Block BP25:* INTSW is cleared to indicate that future requests for interrupts may be honored. Control returns to the calling program at 1, 4.

**DUMP — Dump Caller Routine**

*Blocks BP26 to BP29:* The dump routine first starts to put out a save record on the save tape. (This is initialized to SYSTU.) This save record consists of location 0 through the end of the storage cells for the machine registers which are to be printed out by SYSDFMP.

Location 0 is cleared and traps are disabled. Transfer trap status is saved, and transfer trapping mode is disabled if it was in effect. At DMPRO, the channel is checked to see if the save record has been written. Until it has been completely written, the program loops at this point.

*Blocks BP30 to BP32:* After the save record has been written, BMPF is set to indicate that a dump is requested. (This will cause CALIB1 to load in the dump routine before transferring control to SYSTRA.) When the SYSDMP record is read in, SYSTRA will be set to transfer to the dump program. SYSDMP will be inserted in SYSCUR to indicate that the dump is the current system, and ENBLSW will be set to non-zero to indicate that trapping is not to be enabled before entering the dump program. The key registers and status conditions are saved within the dump program. Control transfers to CALIB1 to re-load IBSUP and SYSDMP, and transfer control to SYSDMP.

**CALIB3/SYSLDR — System Loader**

CALIB3 is entered from the SYSLDR entry in the communications table in the nucleus. The value of the decrement of the TSX instruction used to transfer to SYSLDR determines which one of the three options will be used in loading from the system library unit. If the decrement is equal to 0, the next record in sequence will be loaded. If the decrement is 1, the desired SYSSEC name is taken from the calling sequence, (see inset on Chart BP). SLTABL is scanned to locate the track address and the system record is loaded. If the decrement is 2, the location previously placed in SYSPOS is loaded.

If this routine is entered for a system that is not disk, the next record in sequence is loaded, using the CALIB2 entry on Chart BP. If the dump switch is on when an entry is made into this routine, two records will be loaded. When a dump is requested, the dump caller will properly position the system tape or set up the disk commands to read two records; therefore, the dump routine should be used to provide a dump.

*Blocks BR01 and BR02:* The unit address is picked up from the first word of the UCB and tested to determine if the system unit is disk. If the system unit is not disk, control transfers to CALIB2, which will load in one record from the system tape and transfer control to it. If the system is on disk, location 0 is cleared to provide an error halt. The data channel is reset and the designation SYSSEC is picked up from the calling sequence and saved in case it is required later.

*Block BR03:* The value of "N" from the calling sequence is placed into IBIH, where it is tested to determine which of the three options has been requested. If N equals 0, control transfers to CALSY (Block BR19) to load the next record. If N equals 2, control transfers to CALSY (Block BR20) to load the location indicated in SYSPOS.

*Blocks BR04 and BR05:* If N equals 1, the designated SYSSEC is loaded. Location 0 is loaded with the address for a restart at this point in the program. The channel is reset and started with the channel program at CALSY. This will seek the spare surfaces on disk and write a checkpoint record. Subroutine CALPWS tests for an I/O Check, Sequence Check, Unusual End or Adapter Check during the operation of the channel. If none occurs, control will pass to CALLSY. If any of these conditions occur, control will return to CALLSY to start the channel for another attempt at writing the checkpoint record.

*Blocks BR06 and BR07:* Location 0 is reset to the new restart location. The channel is reset and started with the program at CALSY. This will read in the SLTABL, which contains the table for comparison with SYSSEC and the appropriate track addresses for each entry in this table. Subroutine CALPWS will determine if the read-in was valid before allowing the search of SLTABL.

*Blocks BR08 to BR10:* The SLTABL is scanned for the entry denoted as SYSSEC, and the track address for this
record is then obtained from this table. Location 0 is reset for restart purposes. The channel is reset and started with the program at CALSY. This will restore core to its previous status using the checkpoint record written earlier. Subroutine CALPOW will allow control to pass to DECVD when the record has been read in successfully.

**Blocks BR11 to BR13:** Subroutine DECVD is used to convert the track address found in SLBL to BCD. Subroutine FDAMT generates the required disk orders to load in this record, and these orders are stored in the proper locations.

**Blocks BR14 and BR15:** Location 0 is reset for restart purposes and the channel is reset and started with the program at CALSY. This will seek the desired record, read in its bootstrap for the scatter load routine and transfer control to this scatter load routine. Meanwhile, subroutine CALPOW will determine if the record has been read in correctly and will initiate additional attempts if required.

**Block BR16:** When a complete record has been read in, SYSTCH will have been loaded with a bit in position 3. This is tested to determine if the record is complete, or if it has bridged two cylinders and requires an additional read for completion. SYSTCH will have the track address for the next record in either case. If an additional read is required, control will transfer to Block BR22 to set up the address of the next record desired.

**Blocks BR17 and BR18:** When a complete record has been read in, the dump switch is tested to determine if a dump is requested. If it is, control transfers to Block BR21. If a dump is not requested, R2 is set to point to the location of the UCB for SYSLB, and control transfers to SYSRA-2 (Block BR14).

**Block BR19:** The sense indicators are loaded with the contents of SYSTCH which has the track address for the next record to be loaded (N equals 0).

**Block BR20:** The system position in SYSPOS is loaded into the sense indicators to indicate the next record to be loaded (N equals 2).

**Block BR21:** When a dump is requested or when N equals 0 or 2, bit 3 of SYSTCH is reset to 0 to make certain that only one record will be loaded (unless that record changes the value of this bit in SYSTCH).

**Block BR22:** The contents of the sense indicators are placed into the accumulator. This will be the address of the next record desired in binary form. This could be the result of a dump request requiring a second read, a system that bridges two cylinders, or a request for the record denoted by SYSPOS or SYSTCH. Control passes to Block BR11 for conversion of this address to BCD and subsequent read-in of the desired record.

### Disk Loader Waiting Loop and Disk Channel Routines

Chart BS shows the waiting loop routine CALPOW, and the four channel routines: CALSY, CAL6, CAL7, and CAL9. These channel routines write a checkpoint record on the spare disk surfaces, read SLBL from its disk location, read the checkpoint record back from disk, and read the desired record into the scatter load routine.

**CALPOW** tests for an I/O Check, Sequence Check, Unusual End, or an Adapter Check while waiting for the disk operation to be completed. If any of these are found, control is returned to the main program at such a point as to cause the channel program to be restarted for another try. After a successful read, control will pass to the location following the one which called this subroutine.

**CALSY** seeks the spare surfaces on the disk, and writes a checkpoint record on these surfaces. This will insure that the area used when reading in the SLBL will be restored after this table is used.

**CAL6** reads in the SLBL for comparison with the BCI designation in SYSREC in the calling sequence for SYSLSG. This table consists of two word entries containing the BCI designation for each system record, and the corresponding binary track address for that record.

**CAL7** restores the checkpoint record written by CALSY.

**CAL9** issues a seek for the desired SYSREC, and causes it to be read into its proper storage locations using the scatter load routine.

### CALPOW — Disk Loader Waiting Loop

**Block BS01:** An SRCX instruction is executed which causes the contents of the diagnostic channel to be stored in ISSCH. The contents of the channel check register are stored in positions 6 through 11 of this location by the SRCX command. The sense indicators are loaded from ISSCH so that they may be tested.

**Block BS02:** Bit positions 6, 7, 8, and 11 of the sense indicators are tested to determine if there has been an I/O, Sequence, or Adapter Check or an Unusual End. If there has been any one or more of these conditions, control passes to Block BS05 to modify R2 for an error exit.

**Block BS03:** When no error conditions exist, the channel is tested to see if the operation is complete. Until the operation is complete, the program will loop back to Block BS02 and continue to check for error conditions.

**Block BS04:** When the channel operation is completed, the channel is reset. Control transfers back to the main program at either the location immediately after that which called the subroutine (1, 2 = no error)
or the location immediately prior to the location which called the subroutine \((-1, 2 -\text{error found})\). In the latter case, this will cause the channel operation to be restarted and another attempt made.

*Block BS05*: The, which is being used for the return linkage, is modified to provide the error exit, and control passes to **IBLAST** (Block BS04) for eventual return to the main program.

**CAL8Y — Load System Record**

*Blocks BS06 and BS07*: This is the channel program used to load in a desired system record. Unusual End and Attention Interrupts are inhibited and a seek is issued for the **VSYSEC** desired.

*Blocks BS08 to BS10*: The channel check register is tested to determine when the seek is completed. On completion, a “control read” is given to indicate to the file control a “prepare to verify for cylinder” operation. Control transfers to the scatter load routine which loads the record into its proper storage locations.

**CAL5Y — Write Checkpoint Record**

*Blocks BS11 and BS12*: This is the start of the channel program used to write a checkpoint record on the spare surfaces on disk. Unusual End and Attention Interrupts are inhibited, and a seek is issued for the spare surfaces.

*Blocks BS13 and BS14*: The channel check register is tested to determine when the seek is complete. On completion, a “control write” is given to indicate to the file control a “prepare to verify for cylinder” operation.

*Blocks BS15 and BS16*: A “copy and disconnect” is given to cause the checkpoint record to be written, (or the **SLTAB** to be read, or the checkpoint record to be read if entry was made to either **CAL6Y** or **CAL7Y**). The channel waits until the operation is completed and then disconnects.

**CAL6Y — Load SLTAB**

*Block BS17*: This is the start of the channel program to read **SLTAB** into core when the calling sequence for **SYSQ** requests that a specific **SYSREC** is to be loaded. A “control read” is given to indicate to the file control a “prepare to verify for cylinder” operation, and control transfers to Block BS15 for the “copy and disconnect.”

**CAL7Y — Read Checkpoint Record**

*Block BS18*: This is the start of the channel program to read the checkpoint record that was written by **CAL5Y**. A “control read” is given to indicate to the file control a “prepare to verify for cylinder” operation, and control passes to Block BS15 for the “copy and disconnect.”

**System Return and Print Tape Positions Routines**

*Chart 1*: shows the **IBRET** routine, used to return control to the current system, as specified by **SYSP**. If the position indicated in **SYSP** is valid, a **EXECUTE** (**SYNAM**) card will be simulated, and control will be given to the execute routine to locate and bring this in the system. If this position is not valid, a message will be printed on-line indicating that the current system name is missing and the operator should restore with an **EXECUTE** (**SYNAM**) card in the card reader.

Also on Chart 1 is shown the routine used to print the positions for all output units when **IBSY** is given control after executing a system other than the dump or the editor. If the previous system was the dump, control will be given to **IBRET** without printing the tape positions. If the return was from the editor, the tape positions will not be printed but control will be given to **IBSY**.

**IBRET — System Return Routine**

*Blocks BT01 and BT02*: This is the entry point for the **IBRET** routine. The dump switch is cleared, and the current system position is picked up from **SYSP**. This position is tested for zero; if it is found to be zero, control transfers to **DMPFR** (Block BT07) for an error print-out.

*Blocks BT03 to BT06*: If a valid system position is specified in **SYSP**, the unit address is checked to determine if the current system unit is disk. If the unit specified is disk, control transfers to **SVSLDB** with **N** equal to 2 to indicate that the system denoted in **SYSP** is to be loaded. If the system is not on disk, the **SYNAM** table is searched for an entry which has the matching system position. This table contains two word entries comprised of the system name and its location. If a match cannot be found, control transfers to **DMPFR** (Block BT07) for an error print-out. If the match is found, the system name is picked up from the table and a dummy **EXECUTE** (**SYNAM**) card is simulated. Control then transfers to **IBXCC** (Block BS01).

*Blocks BT07 to BT09*: If the system position is invalid, subroutine **PROUT** is used to print the following error message: **CURRENT SYSTEM NAME MISSING, RESTORE WITH EXECUTE SYNAM IN SYSCMD**. Subroutine **PAUSE** is used to provide an operator action pause. When the operator has completed his operation and depresses **START** on the console, the input switch will be cleared to indicate that the input is to come from the card reader. Control then transfers to **IBRES** (Block BS01). The operator must have a **TAPE** card in his on-line deck in order to have the monitor return to tape input if he was using this prior to the error print-out.
TPPS — Print Tape Positions Routine

Blocks BT10 and BT11: The system name table is moved from its position immediately after the system origin to a working area. The tag bits of each second word entry in the table are moved to the address part of the word and the tag positions are cleared.

Blocks BT12 and BT13: During the loading of the dump program, the dump switch is set to mze. This is interpreted as being off, since the other tests of the dump switch check to see if it is zero and do not check the sign position. The sign is tested here to determine if the dump program has just been executed. If the sign is minus, control transfers to IBNET (Block BT01) without printing the peripheral tape positions. When the return is not from the dump program, the interrupt switch is set to plus.

Blocks BT14 and BT15: The original value of INTSW is tested for a minus condition which indicates a return from the editor. In this case, control transfers to IBSVS (Block BK01), without printing the tape positions. If the return is not from the editor, the UCB address for SYSOU1 is picked up from the system units table.

Blocks BT16 to BT19: This address is tested to determine if there has been a unit assigned to this function. If no unit has been assigned, control transfers to Block BT25. For an assigned unit, subroutine SYUNCV is used to convert this binary address to a symbolic address in the form of channel and unit. This symbolic designation is stored in the print line for this system unit. The file number is picked up from word 3 of the UCB and subroutine BCVDEC is used to convert it to decimal. The file number is placed into the print line and the record number is picked up from word 3 of the UCB. Subroutine BCVDEC is used to convert the record number to decimal and this BCD designation is placed into the print line.

Blocks BT20 and BT21: IN1 is tested to determine if all three units have been processed. If all three units have not be processed, control transfers to Block BT22. When all three units have been processed, subroutine PROUT is used to print the heading line and the three lines of information. Control then transfers to IBNJBA (Block BK01).

Blocks BT22 to BT24: When the three units have not been processed, the unit address for SYSPP1 is picked up. IN1 is tested to determine if this unit has been processed (second unit). If it has not been processed, control transfers to TP0S1 (Block BT16). If the SYSPP1 unit has been processed, the unit address for SYSIN1 is picked up. Control then transfers to Block BT16. If there is a unit assigned to SYSPP1 and/or SYSIN1, the symbolic conversion is made and is stored in the print image.

Block BT25: This block is entered whenever it is found that there is no unit assigned for a particular SYSUNI function. The BCD designation for NO is picked up and placed into the print line in place of the normal symbolic designation. Control transfers to RNXPO (Block BT21) to determine if all units have been processed prior to the message print-out.

Skip Job Routines

Chart BU shows the routine used to allow the operator to skip jobs on an input tape. The operator places sense switch 6 down, and places the number of the next job desired in the entry keys. If the input unit is SYSSN1, this sense switch will be interrogated. The following three conditions must be met in order for the request to be honored: (1) the input unit must be SYSSN1; (2) sense switch 6 must be down and (3) the unit assigned to SYSSN1 must be tape. If all conditions are not met, control transfers to IBSYS. The operator's entry is then checked to see if it is valid. A request of zero causes return of control to IBSYS. A high request (greater than 15) causes an on-line print-out, and a pause for operator clarification. After a valid request, SYSSN1 is rewound and subroutine POUT is used to print an on-line message telling the operator to place sense switch 6 up. Subroutine PAUSE is then used to provide a pause for the operator to perform this action. When the operator depresses START, control transfers to IBSYS.

IBPPIN will transfer control to IBNJBA on each entry while IBNCT is not zero. This routine will save the most recent EXECUTE card and keep count of all ID cards that are encountered. Each ID card will cause IBNCT to be reduced by 1. When IBNCT goes to 1, the EXECUTE card will be simulated. The proper system will be called in after backspacing the input tape over the last ID card read in. A STOP card encountered before the count goes to zero causes an on-line print-out and a transfer to IBSTP for a permanent stop.

IBNJBA Routine

Block BU01: The input switch is tested to determine whether the input unit is SYSSN1. If the input is not SYSSN1, control transfers to IBSYS.

Blocks BU02 to BU04: When the input unit is SYSSN1, sense switch 6 is tested to see if the operator has requested that jobs be skipped. If this switch is up, indicating that he does not want to skip jobs, control transfers to IBSYS. If sense switch 6 is down, the unit address for SYSSN1 is tested to make certain that the actual device attached as SYSSN1 is a tape unit. If a non-tape unit is attached, control transfers to IBSYS. When SYSSN1 is tape, the job number is picked up from the entry keys.
Blocks BU05 to BU07: The job number is tested to determine if the number entered was zero. If zero is specified, control transfers to IBSYS. If the number was not zero, it is stored in IBNCT. Subroutine ODECVA is used to convert this number to decimal, and the result is stored in the print image. Subroutine FROUT is used to print this image informing the operator that the request has been acknowledged.

Blocks BU08 and BU09: On the initial pass through the skip job routine, IBSYS is zero, indicating that the job request has been read the first time. A test is then made to determine if the job number stored in IBNCT is greater than 15. If the job number is 15 or less, control transfers to IBNJ1 (Block BU10) to rewind SYSIN1.

After IBNJ1 is tested the first time, it is made non-zero. Therefore, when the location is tested during a second pass through the routine, control is transferred directly to IBNJ1 regardless of the job number stored in IBNCT.

Blocks BU10 to BU12: If the initial request was less than 15, or after a second request for a number greater than 15, subroutine FROUT is used to print an on-line message telling the operator to place sense switch 6 up, and subroutine PAUSE is used to provide time for the operator to perform this operation. When the operator depressing START, control transfers to IBSYS (Block BC01).

Blocks BU13 and BU14: On the first pass through the skip job routine when the job number exceeds 15, operator clarification is required. Subroutine FROUT is used to print an on-line message which indicates to the operator the job number requested and asks for clarification of the job number. Subroutine PAUSE provides time for operator intervention so that the entry keys may be checked. The operator has the option of making no change to the entry or of changing the job number. When START is depressed, control returns to Block BU02 for a second pass through the routine.

IBNJ2 Routine

Blocks BU15 and BU16: When a skip job request is indicated in IBSYS (IBNCT is not zero), the first word in the input card image is picked up and compared to $EXECUTE. If the comparison is low, indicating that the card read-in is not one of the cards requiring special attention, control transfers to IBSYS to get another card. If there is an equal comparison, routine IBNJ3 is used to save the indicated system for subsequent execution. If the comparison is high, additional checks are made.

Blocks BU17 to BU19: The card image is next compared to $STOP. If it is a $STOP card, control transfers to IBNJ4 (Block BU31) for a message print-out prior to a stop procedure. When the card is not $STOP, it is compared to $ID. If the comparison is not equal, control transfers to IBSYS to get another card. If a $ID is indicated, the count in IBNCT is reduced by 1.

Blocks BU20 to BU23: IBNCT is now tested to determine if the count has been reduced to zero. If the count has not been reduced to zero, control transfers to IBSYS for another card. When the count has been reduced to zero, subroutine NDSEL is used to backspace the input tape over the card just read. Subroutine FROUT is used to print the name of the system to be executed, and a $EXECUTE SYSNAM card will be simulated from the information previously stored. Control transfers to IBXCC (Block BC03) to call in and execute this system.

Block BU24: The block is entered if a $STOP card is encountered before the desired number of jobs have been skipped. Subroutine FROUT is used to print an on-line message to the operator indicating that a $STOP card was found prematurely. Control transfers to IBSTRP to provide a permanent stop to the system.

Block BU25: This block is entered when a $EXECUTE card is encountered during skipping. The system name is saved in order to simulate this card when the desired $ID card is located. Control then transfers to IBSYS to get another card.

Locate Desired System and Validate Channel and Unit Designation

The IFIND routine shown on Chart BV scans the system name table for the system specified on an input card. If the name is found in the table, this name will be stored in SYSCUR as the current system name. If a unit has been assigned to this function, the system position will be stored in SYSPOS. Subroutine NGTPO will position the proper system tape in front of the desired system, and the routine will return to the main program with $M2 set to the location of the SYSPR entry for the required system.

IVAL checks the channel and unit specified in the accumulator in the form PZE CHANNEL, , UNIT. If the designation is valid, the routine will exit to the main program with $M2 set to point to the VCB. The sense indicators are loaded with the contents of word 1 of the VCB and the accumulator contains the address of the VCB. Specification of an invalid or non-existent unit or channel causes an exit with the accumulator set to zero.

IFIND — Locate Desired System Routine

Blocks BV01 to BV03: $M1 and $M4 are saved. They are reloaded with the location and length of the system name table. A comparison of the name specified in the input card image is made with each entry in the system name table. If no equal comparison is found, control transfers to IFND2 (Block BV11).
Blocks BV04 and BV05: If the name is found in the system name table, the name used for comparison is stored in SYSCUR as the current system name. The second word of the entry in the system name table is checked to determine if a unit was assigned to this system. If no unit was assigned, control transfers to IFNDX (Block BV11).

Blocks BV06 and BV07: If a unit is assigned to the system, the system position is picked up from the second word of the entry in the table and stored in SYSPOS. The file position is also saved temporarily to be used in setting up the parameters for subroutine BCTP0. The location of the SYSUNI entry for the required library unit is calculated, and this entry is checked to determine if a unit has been assigned. If no unit has been assigned, control transfers to IFNDX (Block BV13).

Blocks BV08 to BV10: If a unit has been assigned to the SYSUNI entry, the file number is placed in the decrement of the accumulator. Subroutine BCTP0 is used to position the required library unit in front of the desired system. IR1 and IR4 are restored and control returns to the main program at one location after the instruction that called this subroutine.

Blocks BV11 and BV12: If the system name specified on an input card is not contained in the system name table or if no unit is assigned to a valid specified system, the system name is stored in the following error message, "***** UNKNOWN SYSTEM." The address of the message is placed in MQ and control transfers to IBEERR (Block B114) for an error print-out.

Blocks BV13 and BV14: If no unit has been assigned to the SYSUNI entry, the system name is stored in the error message, "***** NO ASSIGNMENT MADE." The address of the message is placed in MQ and control transfers to IBEERR (Block B114) for an error print-out.

IBVAL — Validate Channel and Unit Designation Routine

Blocks BV15 and BV16: IR1 and IR4 are saved and IR1 is loaded with the complement of the channel number. The contents of IR1 are tested to determine if a valid channel (1 through the highest available) has been specified. If an invalid channel is specified (greater than the highest number available), control transfers to Block BV31 prior to returning to the main program.

Blocks BV17 to BV20: If the specified channel is legitimate, the unit designation is loaded into IR2 to be checked for validity. If the designation is greater than 16, indicating a possible disk or Hypertape unit, control transfers to INVI (Block BV22) for additional checking. When the designation is 16 or less, IR2 is tested again to determine whether it is greater than zero and less than 10. A unit designation equal to zero is illegal and causes a transfer of control to Block BV31 prior to returning to the main program. A unit designation greater than 10 indicates a possible card machine and control transfers to INVA (Block BV26) for additional checking. A unit designation less than 10 indicates a tape unit. The total number of units available on the channel is calculated. The number of cards units available is added to the unit location desired. This new value is in IR2 and is transferred to IR4 in complement form. The total number of units available is added to IR4; if the result is between zero and 12, a valid tape unit exists on the channel. If the unit does not exist (IR4 greater than 12), control transfers to Block BV31 prior to returning to the main program.

Blocks BV21 and BV22: If a valid tape unit exists on the channel, the location of the UCB is calculated and placed in IR2 in complement form. This location also remains in the address portion of the accumulator in true form.

Block BV23: The contents of word 1 of the UCB are loaded into the sense indicators.

Block BV24: This is the exit from the IBVAL routine. IR1 and IR4 are restored and control returns to the main program.

Block BV25: If a possible disk unit is indicated, the desired address is compared with the first word of each UCB for this channel. If a valid comparison cannot be found, control transfers to Block BV31 to exit from the routine. If a valid comparison is found, control transfers to Block BV30 with IR2 set to point to the desired UCB.

Blocks BV26 to BV29: Entry is made to INVA with IR2 equal to some value between 1 and 6 indicating a possible card machine. (The value in IR2 is the result of an initial setting of 11 to 16 because the transfer to INVA caused IR2 to be reduced by 10.) IR2 is now tested to determine if it is greater than 3. If it is greater than 3 (indicating an original value of 14 to 16 which is illegal), control transfers to Block BV31 to exit from the routine. When the contents of IR2 (1, 2, or 3) indicate a legal card device, the number of devices and the address of the first UCB are picked up from the proper UCB locator. The address is placed in IR4 to be used in checking for the desired unit and the number of devices is placed in IR4. IR4 is then tested to determine if there are any card devices on the channel. If there are none, control transfers to Block BV31. If there are legal card devices on the channel, the unit number in IR2 is used to form the unit address to be used for comparison with each UCB. These UCB's for the card machines on the channel are tested to determine if the machine
indicated in m2 exists. If it does not exist, control transfers to Bv31.

Block BV30: If a legal card device or disk unit is indicated, the location of the UCB is placed in the accumulator. Control then transfers to Block BV23.

Block BV31: This block is entered whenever there has been an illegal unit specified or an illegal channel specified. The accumulator is cleared and control transfers to Block BV24 to exit from the routine.

INITUS — Initialize System Tape Usage Routine

Subroutine INITUS is used to initialize the routines requiring the following system tapes: SYSLB1, SYSOU1/2, SYSCK1/2, and SYSUT4. It will also initialize the loader for disk if SYSLB1 is assigned to disk.

Entry to the subroutine is made with m2 pointing to an indirect reference to the UCB for the unit desired. If m2 is pointing to a reference word with a zero address, indicating that there is no unit assigned, a further check will be made to determine if the unit should be a dump tape. If the unit being initialized is a dump tape and no unit has been assigned, SYSMP and location 00002 will be altered to bypass dump requests. An online message will also be printed to inform the operator that dumps cannot be taken.

For a unit that is assigned to tape, the density will be set as indicated by the sign of the SYSUNI entry. The I/O addresses in the routines requiring this unit will be set for the proper unit.

If the unit is SYSLB1 and assigned to disk, the disk orders will be generated and the channel codes will be set up for disk in SYSLD8.

The return from this subroutine is via 1, 1.

Blocks BW01 to BW03: Entry is made with m2 pointing to a UCB locator in the SYSUNI table. If the address of this locator is zero, indicating that there is no unit assigned, control transfers to INITL3 (Block BW09). If a unit is assigned, the availability and no-repositioning flags are reset. Bits 4 and 11 of the unit address are checked to determine if the unit is disk. If the unit is disk, control transfers to DINLB1 (Block BW18).

Blocks BW04 to BW06: If a tape unit is assigned, subroutine NDSKL is used to set the density of the desired unit according to the sign of the entry in the SYSUNI table. m2 is tested to determine if the unit being referenced is SYSLB1. If the unit is SYSLB1, control transfers to INITL1 (Block BW15). If the unit is not SYSLB1, m2 is tested next to determine if the unit is SYSUT4. If the unit is SYSUT4, control transfers to INITL2 (Block BW16).

Blocks BW07 and BW08: If the assigned tape unit is neither SYSLB1 nor SYSUT4, m2 is tested to determine if the unit could be one of the system output units (i.e., SYSOU1, SYSOU2, SYSCK1, or SYSCK2). If it is not one of the output units, control returns to the calling program via 1, 1. If the assigned unit is any one of the output units, the no-repositioning flag is set and control returns to the calling program via 1, 1.

Blocks BW09 to BW14: If a unit is not assigned, m2 is tested to determine if the unit is either SYSOU1 or SYSUT4. If it is neither of these units, control returns to the calling program via 1, 1. If it is one of these units, m2 is tested to determine which one it is. If the unit is SYSOU1, subroutine PROUT will be used to print an online message indicating that SYSOU1 is unassigned, and no dump can be taken. If the unit is SYSUT4, a similar message is printed. Location SYSMP in the communications table, and location 00002 are modified to bypass the dump and transfer directly to the loader. Control then exists to the calling program via 1, 1.

Block BW15: This block is entered from Block BW05 if the unit specified is SYSLB1. All the I/O commands in the loader (CALLBD) will be set up to reference this unit and control will exit to the calling program via 1, 1.

Blocks BW16 and BW17: These blocks are entered from Block BW06 if the assigned tape unit is SYSUT4. All the I/O commands in the dump caller are set up to reference this unit. The linkage to the dump caller is placed in SYSMP in the communications table, and in location 00002. Control exits to the calling routine via 1, 1.

Blocks BW18 to BW22: These blocks are entered if the unit assigned is disk. If the assigned disk unit is not SYSLB1, control exits to the calling routine via 1, 1 without performing any setup operations. If the unit is SYSLB1, m1 and m2 are saved. The disk orders for the system loader are generated and stored in the SYSLD8 routine. The SMS masks and the channel codes are setup in SYSLD8, and m1 and m2 are restored. Control exits to the calling program via 1, 1.

SYUCU — Symbolic Unit Conversion Routine

Chart BX shows the SYUCU routine used to convert the symbolic unit designated in columns 16 to 18 of the input card image to the form: PRE CHAN, UNIT. Exit is made from this routine with this new form in the accumulator and in location LATUN. The return is made via 2, 4 for a normal return, and via 1, 4 for an error return if the unit specified is illegal.

A prefix of MXE indicates a model IV or VI tape drive. If the unit is one that is attached to a 7000 channel, the unit address is in a compressed form. This compressed
form indicates the channel, device, interface, access, and module.

**Blocks BX01 to BX05:** The symbolic unit designation is picked up from the input card image and placed in the **m0**. The channel letter (if the unit is tape or disk) is placed in **m1**. If the unit is a card machine, **m1** will contain either an “R” or a “P”. **m1** is tested to determine if the value of the channel letter is less than an “A”. If it is, indicating that the channel designated is numeric or one of certain special characters, control exits to the calling program via 1, 4.

In the process of testing **m1** for a channel designation lower than “A”, **m1** is reduced by an octal 20 if the channel designation was “A” or higher. **m1** is now tested to determine if the present value lies above 26 octal, which would indicate an original letter of “P” or higher. If the letter was “P” or higher, control transfers to **SYUC6** (Block BX12) to check and determine if a card machine was specified. If the specified designation is lower than “P”, **m1** will contain a number from 1 to 26 for the letters “A” through “O” which might have appeared on the input card. A channel specification higher than “H” but lower than “O” will be permitted by this routine, but will be indicated as an error by the **INVAL** routine which is called immediately after this routine returns to the main program.

The second character of the symbolic designation is placed in **m2**. **m2** is tested to determine if this character is numeric. If the character is not numeric, control transfers to **SYUC6** (Block BX17).

**Blocks BX06 to BX09:** **m2** is tested to determine if the unit specified is 0. If the unit is 0, **m2** is reset to 10. The model designation is picked up from the input card image and tested to determine if the model type specified is model **r** or **v**. If it is **r** or **v**, control transfers to **SYUC3** (Block BX10). If it is not **r** or **v**, the “P” bit is set in the accumulator, to indicate model **v**, and control transfers to Block BX11.

**Block BX10:** The accumulator is cleared. This is required to remove any prefix or tag. The only case where a prefix should be present is if the unit specified is not a model **r** or **v** tape unit. In this case, this one instruction will be bypassed.

**Block BX11:** The contents of **LATUN** are set up as follows: The prefix is taken from the accumulator and would be minus for a model **r** or **v** tape drive. The decrement is taken from **m2** which contains the unit number; and the address is taken from **m1** which contains the channel number. Control exits to the calling program via 2, 4 for a normal return in this case.

**Blocks BX12 to BX16:** **m1** is tested to determine if a “P” or an “R” was specified as the first character to indicate a designation of a card machine. If neither was specified, control exits to the calling program via 1, 4. If a “P” or an “R” is specified, the second character of the designation is compared with the characters “R” “U”, and “D”. If no comparison is found, control exits to the calling program via 1, 4. If a valid letter is found, the unit number will be calculated and placed in **m2**, (i.e., 11 for **MFX**, 12 for **FUX**, and 13 for **PUX**). The channel letter is then shifted out and reduced by an octal 20 and placed in **m1**. Again, an invalid letter that is higher than “H” will not be detected now, but will be found in the **INVAL** routine used later in the calling program. A numeric or special character lower than “A” in the **KCD** representation within the machine will cause the error exit to be taken and a return to the calling program via 1, 4. Otherwise, control transfers to **SYUC3** (Block BX10) to set up a normal exit via 2, 4.

**Blocks BX17 to BX19:** If a possible disk or Hypertape unit is indicated, the channel number is stored temporarily in **SYUC6** (address), after being shifted left 5 places. The characters denoting access are shifted into the accumulator, and **m2** is tested to determine if the type unit specified is greater than “D”. If the type is greater than “D”, control transfers to **SYUC7** (Block BX23). If the type is not greater than “D”, **m2** is tested to determine if the exact character is “D”. If it is not a “D”, control exits to the calling program via 1, 4.

**Blocks BX20 to BX22:** If the type unit specified is “D” (disk), **m1** is loaded from the address portion of the accumulator. It is tested to determine if an access greater than 1 was specified. If there was a specification other than 0 or 1, control exits to the calling routine via 1, 4. If the access designated was legal, the designation will be on'ed with the channel designation temporarily stored in **SYUC6**. The next word from the input card image is loaded into the **m0** and the first character denoting the module is shifted into the accumulator, and then loaded into **m2**. **m2** is tested to determine if a proper module has been designated. Any character higher than a “9” will cause an exit to the calling program via 1, 4. A legal character will be on'ed with the compressed disk address being constructed in **SYUC6**. The interface designation is tested in the **m0**, and, if required, the interface bit will be on'ed to the word in the accumulator. Additional bits are added to the constructed word to indicate a 79009 channel and a disk device. This constructed word is now in **m2** and control transfers to **SYUC3** (Block BX10).

**Blocks BX23 to BX25:** **m2** contains the second character of the symbolic unit designation. At this point it is known that the character is higher than a “D”. **m2** is tested to determine if the character is an “H”. If the character is not an “H”, control exits to the calling pro-
gram via 1, 4. If the character is an “H” (Hypertape), the unit number is in the accumulator and is loaded into m2. m2 is tested to determine if a legal tape number has been specified. If a legal tape number (0 through 9) has not been specified, control exits to the calling program via 1, 4. For a legal tape number, the number will be appended onto the channel number that is temporarily stored in svuc6. The character denoting interface is tested in the mq, and, if required, the interface bit will be or’ed to the compressed address now in the accumulator. This compressed address is placed into m2 and then incremented to add the bit denoting a 7909 channel. Control then transfers to svuc3 (Block bxi0).

**RSYUR — Remove Unit from Availability Chain Routine and Sample Availability Chain**

Chart by shows the rsysr routine which removes a unit from the availability chain. It is entered with the location of the ucbl for the unit to be removed located in the address of the accumulator. All index registers are preserved, and if the accumulator address is zero, the exit will be taken from the routine with no work performed.

Chart by also shows the makeup of a typical channel availability chain.

**RSYUR Routine**

*Blocks BY01 to BY03*: Index registers 1, 2, and 4 are saved, and the location of the desired ucbl is saved in the compare cell, rsv3. rsv3 is tested to determine if there was an address specified in the accumulator on entry. If the accumulator address had been zero, control returns immediately to the calling program via 1, 4. If there is a valid address in rsv3, the address of the desired ucbl is loaded into m2 to obtain the unit address. The channel is extracted from this address, and used to calculate the availability chain locator for the desired channel.

*Blocks BY04 to BY06*: The address in the availability chain locator (i.e., ucbl for channel A) is compared with the address stored in rsv5. If the addresses are equal, control transfers to rsv3 (Block BY07) to remove the unit reference from the chain. If the addresses are unequal, the address just compared to rsv5 is loaded into m4 to be used in obtaining the next ucbl in the chain. This address is tested to determine if the end of the chain has been reached (i.e., last address is zero). If we are at the end of the chain, control transfers to rsv2 (Block BY08). If the end of the chain has not been reached, control returns to rsv1 for another comparison of this new address with the address in rsv5.

*Block BY07*: When a ucbl is found which has an address which compares with the one desired, the address of the unit immediately after the one to be removed is picked up and placed in the prior ucbl or the locator. The prior ucbl or locator was originally pointing to the unit that is being removed.

*Block BY08*: After finding the specified unit in the availability chain or after checking all units in the chain without success, the index registers are restored and control exits to the calling program via 1, 4.

**Sample Availability Chain**

The sample on Chart by shows the chaining of addresses in the availability chain locators and the availability chain itself.

ucwav indicates the location of a table of availability chain locators (i.e., ucca, ucbb, etc.). This table is comprised of one entry for each channel, and its length may be found in the decrement of ucwav.

The address portion of each entry in the locator table contains the address of the first ucbl in the availability chain. The address portion of word 1 of this ucbl contains the address of the next ucbl in the chain. The address portion of word 1 of the last ucbl in the chain is zero. If the chain contains no entries, the address portion of the locator (i.e., ucca) is zero.

Within the chain, the addresses of the ucbl’s will not necessarily be in the order in which they appear in memory.
Chart BB. Cold Start and Initialization, Sheet 2
Chart BC. $IBSYS card routine
Chart BD. $RESTORE card routine
Chart BE. Control card interpretation routine, IBCDIN/IBTPIN
Chart BF. $ATTACH and $DETACH card routines
Chart BG. $EXECUTE card routine
Chart BJ. Miscellaneous card routines and error routines
Chart BK. $SSWITCH card routine
Chart BL, SUNITS card routine
Chart BM. $RELEASE$ and $AS$ card routines
CHART BN. $1BET$ card routine and General tape positioner routine
Chart BP. CALIB1, SYSRPT and DUMP routines
Chart BR. SYSLDR (CALIB3) routine
Chart BS. Disk Loader, waiting loop, and channel routines.
Chart BT. IBRET and TPS routines
Chart BU. IBNJB and IBNJ2 (Skip Job) routines
Chart BV. IFIND and IBVAL routines
Chart BW. INITUS routine
Chart BX. SYUCU symbolic convert routine
Chart BY. RSYUR routine and sample chain layout
**ACTV/(ACTIV — Activate Routine**

Charts CA and CB describe the activate subroutine used to initiate an I/O operation on a unit assigned to the specific file designated in the calling sequence.

**Block CA01:** Subroutine TEST is used to check for a valid file. If the unit is unavailable or is an illegal unit, there will be no return to the ACTV routine. A message showing the illegal unit reference will be printed online, and the program will halt using the STOP subroutine.

**Blocks CA02 and CA03:** On a return to ACTV, traps are disabled and IR1 is stored in the proper exit routine (ACTX1 for the non-trap exit, and ACTV1 for the trap exit).

**Blocks CA04 to CA06:** The channel number is extracted from word 1 of the UCB and IR1 is loaded with – (channel number – 1). IR2, IR4, and the sense indicators are saved. IR5 is set to -L (UCB).

**Block CA07:** The channel activity cell (CHXAC) is tested to determine if the channel is dormant. If it is not, control transfers to ACT6 (Block CB01).

**Block CA08:** If the channel is dormant, word 1 of the UCB is loaded into the sense indicators. Bit position 4 of the indicators is tested to determine if this is a 7909 channel. If it is a 7909 channel, control transfers to ACT4 (Block CA12).

**Blocks CA09 and CA10:** If the dormant channel is not a 7909 channel, word 2 of the UCB is picked up. The location of the select routine is placed in IR4, which is tested to determine if there is a select routine specified. If there is not, control transfers to ACTX2 (Block CB14).

**Block CA11:** If a select routine is specified, the I/O indicators (TCOX, TRCX, and TRFX) are tested. BRX is set to non-zero to indicate normal redundancy mode and control passes to ACT3 (Block CA16).

**Block CA12:** If the dormant channel is a 7909 channel, word 3 of the UCB is tested to determine if the AF flag is on. If the flag is on, control transfers to ACT5 (Block CB07).

**Blocks CA13 to CA15:** After the channel checks are made and a select routine is indicated, the select routine address is placed in IR4. The contents of IR4 are tested to determine if an address is present. If no select routine is specified, control transfers to ACTX2 (Block CB14). If an address is present, BRX is cleared, and control transfers to ACT3 (Block CA16).

**Blocks CA16 and CA17:** If a select routine is specified, UVAX is set to zero to reset the redundancy count. The channel activity cell (CHXAC) is set to the location of the UCB, IR4 is stored in the exit address as the exit to the user's select routine, and control is given to this routine as a subroutine. The normal return from the select routine is via 1, 4 to Block CA18. (For an explanation of the other two returns that may be made from a select routine, refer to the IBM Reference Manual, Basic Monitor (IBSYS), Form C28-6245).

**Block CA18:** Subroutine BOUND is used to permit redundancy trapping if the operation is a write operation. The return from this subroutine is via 2, 4 to ACTX2 (Block CB14).

**Block CA19:** This is entered if the return from the select routine was via 2, 4. The channel activity cell, CHXAC, is cleared and control passes to ACTX2 (Block CB14).

**Block CB01:** This block is entered from Block CA07 if the channel is in use. The prefix of word 1 of the calling sequence is checked to determine if the entry is PZE or MZE. If the entry is PZE, control transfers to ACTX2 (Block CB14).

**Block CB02:** If the entry is MZE, the address of the UCB is picked up and stored in CHXSP to give that unit priority on this channel.

**Block CB03:** This block is entered from Blocks CB02 and CB11. IR2, IR4, and the sense indicators are restored in preparation for returning to the using program.

**Blocks CB04 and CB05:** TRPSW is tested to determine if this is trap time. If TRPSW is not zero, indicating trap time, control transfers to ACTY1 (Block CB16). If this is non-trap time, traps are enabled. CHXAC is checked to determine when XRAP has entered SEL+ or when the channel becomes dormant. The program loops, making these checks until either condition occurs. Control then transfers to ACTX1 (Block CB06).

**Block CB06:** This is the non-trap exit from ACTV. IR1 is restored, traps are enabled, and control is returned to the using program via 2, 4.

**Blocks CB07 and CB08:** Entry is made to these blocks from Block CA12 if the AF flag is on or for a dormant 7909 channel. CHXAC is set to the address of the UCB to indicate channel activity. Word 1 of the calling sequence is checked to determine the type of entry to ACTV. If the entry is PZE, control transfers to ACT5A (Block CB12).

**Blocks CB09 to CB11:** If the entry to ACTV is an MZE entry, the address of the UCB is stored in CHXSP to give
this unit priority. Subroutine $\text{issk}_1$ is used to issue one
seek. $\text{stcx}$ is executed to allow the channel to proceed
and enable attention interrupts. Control then transfers
to $\text{acty}$ (Block $\text{cb}03$).

Blocks $\text{cb}12$ and $\text{cb}13$: With a $\text{pze}$ entry to $\text{acty}$,
subroutine $\text{issk}$ is used to issue seeks as required.
$\text{stcx}$ is executed to allow the channel to proceed
and enable attention interrupts.

Block $\text{cb}14$: At $\text{actx}2$, the sense indicators, $\text{mr}2$, and $\text{mr}4$, are restored in preparation for returning to the
using program.

Block $\text{cb}15$: $\text{trpsw}$ is tested to determine if this is
trap time or non-trap time. If $\text{trpsw}$ is zero, control
passes to $\text{actx}1$ for a non-trap exit. If $\text{trpsw}$ is not zero,
control transfers to $\text{acty}1$ for the trap exit.

Block $\text{cb}16$: $\text{mr}1$ is restored and control is returned to
the using program at $\text{s}, \text{s}$ (trap exit).

SAVE/XTRAP — Trap Supervisor
This routine monitors traps occurring on the data
channels as a result of the completion of an I/O operation or
as a result of a redundancy on a channel when re-
dundancy trapping is enabled. An attempt is made to
recover when errors are detected, and the position of the
tape is updated in the $\text{ucb}$ if tape movement is in-
volved.

Block $\text{cc}01$: This entry point is used whenever there
is an illegal channel trap from a non-existent channel.
Location $\text{coof}$ is set to non-zero to indicate this error
condition.

Blocks $\text{cc}02$ to $\text{cc}03$: $\text{frsw}$ is tested to determine
if the trap is the result of an operation that was forced
on the on-line printer or punch. If $\text{frsw}$ is not zero,
control transfers to $\text{frv}$ to complete the operation
which may have been interrupted by the use of $\text{punch}$
or $\text{fruto}$. If $\text{frsw}$ is zero, $\text{dchsw}$ is tested to deter-
mine if the trap is the result of an invalid address re-
cover or a no-record-found recovery from disk. If
$\text{dchsw}$ is non-zero, indicating a disk recovery, control
transfers to $\text{dend}$ (Block $\text{cc}17$). If $\text{dchsw}$ is zero, the
machine registers are saved along with the transfer
trapping mode and overflow conditions.

Blocks $\text{cc}04$ to $\text{cc}06$: These blocks are entered either
from Block $\text{cc}03$ or from the $\text{frout}$ routine to complete
the processing of a channel A trap which had been inter-
rupted, or from the exit routine out if channel A
had been active when an entry was made to $\text{frout}$. The
trap code word is placed into the accumulator and sense indicators. If $\text{frsw}$ is zero, indicating an initial
trap on a channel without the concurrent use of $\text{frout}$,
the address portion of this code word is stored in $\text{exit}$
as the exit address from $\text{xtrap}$. If $\text{frsw}$ is non-zero, the
address of the code word is not moved, and control
transfers directly to Block $\text{co}07$.

Blocks $\text{cc}07$ and $\text{cc}08$: $\text{mr}2$ is loaded with the con-
tents of location $\text{coof}$. If $\text{mr}2$ is not zero (result of an
illegal-channel trap), control transfers to $\text{tst}1 \text{-} 3$. A
message is printed to indicate an illegal unit reference
followed by a dead stop. If $\text{mr}2$ is zero, $\text{mr}1$ is set to
$-1$ (channel number $-1$).

Blocks $\text{cc}09$ and $\text{cc}10$: $\text{chxac}$ is tested to determine
if there is activity on the channel. If there is no activity,
control transfers to $\text{end}$ (Block $\text{ce}02$). If the channel
is active, the trap switch is set and the channel results
are moved from $\text{acom}$ to $\text{comm}$.

Blocks $\text{cc}11$ to $\text{cc}13$: $\text{frsw}$ and $\text{frsw}$ are both
tested for a zero condition. If both are zero, indicating
that there is not a wait to print an on-line message, the
channel information will be stored in $\text{comm}$ by the use
of the indirectly addressed store channel instruction.
If either or both ($\text{frsw}$ and $\text{frsw}$) are not zero, the
store channel instruction will be bypassed.

Block $\text{cc}14$: $\text{trpsw}$ is set to point to the $\text{ucb}$ at this
time and is now used to pick up the actual unit address
from word 1. Bit position 4 of this word is tested to
determine if the unit is disk. If a disk address is present,
control passes to Block $\text{cc}01$. If a non-disk address is
present, control transfers to $\text{ndend}$ at Block $\text{cc}15$.

Blocks $\text{cc}15$ to $\text{cc}18$: A test for an I/O Check is
made. If there was NO I/O Check, control transfers
immediately to $\text{end}2$ (Block $\text{cc}19$). If there was an I/O
Check, subroutine $\text{mrwr}$ is used to print I/O CHECK 7607
and subroutine $\text{paus}$ is used to allow an operator action
pause. At the completion of the pause, the contents of
$\text{trpsw}$ are replaced in the accumulator and control
transfers to $\text{end}2$ (Block $\text{cc}19$).

Block $\text{cc}19$: $\text{mr}2$ is set to $-1$ (UCW1). The sense in-
dicators contain the trap code word which includes the
trap type in bits 15, 16, and 17. The remainder of the
left half of the indicators is cleared and the sign is
set to one.

Block $\text{cc}20$: Bits 15 and 16 of the sense indicators
are tested to determine if the trap resulted from either
an EOF or a redundant write. If neither condition is
present, control passes to Block $\text{cc}01$. If either condi-
tion is present, control transfers to $\text{end}$ (Block $\text{cc}21$).

Block $\text{cc}21$: Bit 16 of the sense indicators is tested
to determine if the trap resulted from a redundant
write. In this case, control transfers to $\text{redw}$ (Block $\text{cp}01$).

Blocks $\text{cc}22$ and $\text{cc}23$: At this point the trap must
be the result of an EOF. Bit 1 of the sense indicators is
set to one to indicate this fact, and the redundancy
indicator is turned off. The old tape position is saved in
$\text{ltpos}$, and the record count is set to 777777. Control
now passes to $\text{end}$ (Block $\text{ce}11$).
Blocks CD01 to CD03: These blocks receive control from Block CC14 if this is a trap from a 7909 channel. DCHSW is set to L(UCB), and the "store diagnostic channel" command is executed. The sense indicators are loaded with the data obtained from the SDCX instruction, and position 35 is reset to 0. (The contents of the six-position channel check condition register are presently in positions 6 through 11.) Positions 6, 7, and 11 are tested to determine if there has been an I/O, Sequence, or Adapter Check. If any of these conditions are present, control transfers to DEND1 (Block CD06).

Block CD04: When none of the check conditions is present, bits 8, 9, and 10 of the sense indicators are tested to determine if there has been an Unusual End or an Attention Interrupt. If neither condition is present, control transfers to DEND2 (Block CD16).

Blocks CD05 to CD07: If either an Unusual End or an Attention Interrupt is indicated, COMM is examined to determine if this is a SYSTWT trap. If it is a SYSTWT trap, control transfers directly to subroutine DATNS. If it is not a SYSTWT trap, bit 35 of the sense indicators is set to 1 before transfer to DATNS. This subroutine is used to perform an sns sequence and to obtain the sense data words from the channel being tested.

Block CD08: Bit 8 of the sense indicators is again tested to determine if there has been an Unusual End. If there has been an Unusual End, control transfers to EKREV (Block CP01); otherwise, control passes to DEND2 (Block CD16).

Blocks CD09 to CD15: These blocks are entered from Block CD03 if there has been an I/O, Sequence or an Adapter Check. The error bits are placed in adjacent positions in the accumulator and MQ, and packed to eliminate the unused bits. Each error bit is tested in turn, and an error message is printed by subroutine POUT for each bit that is found on. The I/O Check indicator is turned off if it was on, and subroutine PAUSE gives the operator a chance to correct the difficulty. Control then transfers to Block CD08 for an Unusual End check.

Blocks CD16 and CD17: Entry to DEND2 is made if no Unusual End is indicated. The sign of the accumulator is set minus for entry into subroutine ISSEK. This subroutine is used to issue any pending seeks.

Blocks CD18 to CD21: Bit 35 of the sense indicators is tested to determine if this is a SYSTWT trap. If it is not, control transfers to END9 (Block CE22). If this is a SYSTWT trap, the Unusual End test word (DUELL) is reset to zero, and M2 is set to point to the UCB. The first four bytes of sns data are placed into the sense indicators in positions 12 through 35. (Positions 0 through 5 contain the 7909 control counter and positions 6 through 11 contain the contents of the channel check register.) Control transfers to END14 (Block CE12) to enter SEL--.

Block CE01: This block is entered from Block CC20 if there has not been an EOF or redundancy encountered while writing. The accumulator contains the contents of the channel activity cell, and is tested to determine if any blank tape has been written. If the sign is minus (indicating that blank tape has been written), control transfers to ENDS (Block CE20).

Block CE02: If blank tape has not been written, the MQ is loaded with the second word of the UCB and the sign of the MQ is tested. If the sign is plus (indicating a read operation), control transfers to RED12 (Block CH01).

Blocks CE02.5 to CE04: For a write operation, bit 3 of the sense indicators is set to 1. An ETR instruction is executed. If the end-of-tape marker has been sensed, bit 18 in word 1 of the UCB and bit 1 of the sense indicators are set to 1.

Block CE05: After the end-of-tape test, subroutine NSTR is used to check the word count and determine if the record was a noise record. If the record is a noise record, the sign bit of the sense indicators will be set to 0.

Block CE06: The proper bit for redundancy trap on this particular channel is turned off in TRAP.

Block CE07: This block is entered from Block CE06 and also from Block CH05 in the event that a read redundancy is found to be a noise record. The contents of the sense indicators are saved in RWTNS for later use.

Blocks CE08 to CE10: These blocks are entered from Block CE07 and also from Blocks CE18 and CH02 in the read redundancy-recovery routine whenever no redundancy is found or redundancy is disabled. The tape position is picked up from word 3 of the UCB. The sense indicators are tested to determine if the operation is read or write. If the operation is a read operation, control transfers directly to ENDS (Block CE11). If the operation is a write operation, bit positions 19 and 20 are masked out of the record count and control transfers to ENDS.

Block CE11: This block is entered from Blocks CE09 and CE10 as described above, and also from Block CE23 on an end-of-file condition. The record count is incremented by 1 and stored in word 3 of the UCB.

Block CE12: The address of the user's select routine is picked up from word 2 of the UCB and placed in the subsequent transfer instruction to allow IOEX to transfer control to that routine.

Block CE13: This is the user's select routine which has three possible returns. The first return (1, 4) is the normal return which transfers control to END9 at Block CE22. The second return (2, 4) transfers control to ENDS (Block CE14). This return is used for both reading and
writing when the record in question is considered to be a noise record. The third return (3, 4) to Block CE23 is used for reading when the normal redundancy recovery procedure is to be used.

**Blocks CE14 to CE16:** When the record in question is considered to be a noise record, DCHSW is tested to determine if this is a disk operation. If it is, control transfers to END9 (Block CE23). If this is not a disk operation, the contents of the sense indicators are restored from RWIND. Position 3 of the sense indicators is tested to determine if this is a write operation. If it is, control transfers to END12 (Block CE21).

**Blocks CE17 to CE19:** If a write operation is not indicated, subroutine RE10 is used to print the file and record location and NOISE RECORD discarded. The redundancy count is reset to zero. The noise record bit is added to word 3 of the UCB and the record count is reduced by 1.

**Block CE20:** The contents of the channel activity cell are picked up. Control transfers to BG1 + 1 to read the next record from this unit if priority is still assigned to this unit, or to search for another unit that is waiting and initiate an I/O operation on that unit.

**Block CE21:** During a write operation, subroutine RE10 is used to print the file and record count and the message SHORT RECORD WRITTEN.

**Block CE22:** The redundancy count is set to zero as required, and control passes to BG1 (Block CF01).

**Blocks CE23 and CE24:** These blocks are entered when return 3 (3, 4) is taken by the user's select routine. DCHSW is tested to determine if the unit in use is disk. If the unit is disk, control transfers to Block CE20 to eventually re-enter the SEL+ routine. If the unit in use is not disk, the record count in word 3 of the UCB is reduced by 1, and control transfers to Block CE16, to enter the read-redundancy-recovery routine after the noise record test.

---

**BG1 — Initiate Next Channel Operation Routine**

Charts CF and CC show the routine used to initiate the next operation on a channel. It is entered from the trap supervisor XTRAP when it is desired to start another operation after a prior operation has been completed.

The channel priority cell is checked to determine if the next operation is to be initiated on a specific unit. If the priority cell is not set to indicate a specific unit, a search is made for another unit on this channel which is awaiting an I/O operation. When it is determined which unit is to be the next one requiring an I/O operation, SEL+ is entered for that unit. If no unit requires an I/O operation and there are no pending seeks to be issued for disk, the activity cell is cleared, and operation ceases on that channel.

The key registers and status conditions are restored before returning to the point interrupted by the data channel trap. If there are any seeks remaining to be issued, they will be issued and the activity cell will not be cleared before restoring the machine registers and status conditions.

**Block CF01:** The address of the priority UCB is picked up from the channel priority cell, CHSP, for the proper channel.

**Blocks CF01.5 and CF02:** The disk channel switch, DCHSW, is tested to determine if the trap resulted from an operation on a 7909 channel. If a 7909 channel caused the trap (DCHSW is not zero), control exits to Block CF03. If the trap was not from a 7909 channel, the checkpoint switch (CHSPW) will be tested to determine if the program is in the process of checkpoint. If a checkpoint is in process, control transfers to Block CF17. If a checkpoint is not in process, control transfers to Block CF03.

**Blocks CF03 to CF08:** At this point the accumulator contains the contents of the priority cell for the desired channel. If this address is zero, indicating that priority was not assigned to a unit at this time, control transfers to BG2 (Block CF09) to search for a unit which can be given priority. If the priority cell was not zero, it will be used to locate the address of the select routine being used by that unit. If the location of the select routine is not zero, control transfers to Block CF10. If the location of the select routine is zero, the priority cell word 2 of the UCB are cleared.

**Block CF09:** IRA is set to point to the UCB for the unit that just completed an operation. The accumulator is set to zero except for the "P" bit. Control passes to Block C001 to search the UCB's for a unit with an I/O operation waiting to be performed.

**Blocks CF10 to CF12:** DCHSW is tested to determine if the channel is a 7909 channel. If this is not a 7909 channel (DCHSW is zero), control transfers to BG5 (Block CF13). If it is a 7909 channel (DCHSW is not zero), word 3 of the UCB is tested to determine if the activity flag is on for this unit. If the activity flag is on, control transfers to BG6 (Block CF20). If the flag is not on, the accumulator is cleared in preparation for entering SEL+.

**Blocks CF13 to CF16:** These blocks will be entered with the accumulator cleared for a 7909 channel, and not cleared for a 7807 channel. The contents of the accumulator are stored in the redundancy check control cell for the proper channel. Clearing this cell suppresses redundancy checking and recovery while reading. The trap switch and channel activity cell are both set, and the user's select routine is entered. The normal return
from the select routine is via 1, 4. Subroutine RDUN is used to set up the trap indicators in TRAP to allow redundancy trapping on a write operation. If the operation is a read operation, redundancy trapping is not permitted and TRAP is not altered. A non-data select operation returns from the select routine via 2, 4, and control transfers to GCI (Block CG01) to initiate another operation on the same channel. Upon return from RDUN, control transfers to Block CG01 to restore the key registers and status conditions and then return to the program at the point interrupted by the data channel trap.

Blocks CF17 to CF19: These blocks are entered if a checkpoint has just been processed on a 7907 channel, or at the unsuccessful completion of the search for a unit awaiting priority. The activity cell is cleared and DCMSW is tested to determine if there are any more seeks waiting to be issued. If DCHSW is zero and/or plus, indicating that there are no more seeks to be issued, control transfers to Block CG01 to return eventually to the point interrupted by the channel trap. If there are seeks remaining to be issued, the activity cell is reset.

Blocks CF20 and CF21: Subroutine issx is used to issue any pending seeks. Attention interrupts are enabled before transferring control to Block CG01 to return to the trapped location.

Block CG01: This block is entered when it is necessary to search the vcb's for a waiting unit. The disk channel switch, DCMSW, is tested to determine if this is a 7909 channel. If it is a 7909 channel (DCMSW is not zero), control transfers to Block CG07, and MZ is decremented to permit testing the activity flag.

Block CG02: If a 7909 channel is not indicated, word 2 of the vcb is tested to determine if there is an address of a select routine present. If word 2 is not zero, indicating that there is an address of a select routine, control transfers to Block CG06.

Blocks CG03 to CG05: If word 2 is zero, MZ is tested to determine if the last vcb has been tested. If the last vcb has not been tested, control returns to Block CG01 to test the next one. When MZ is reduced to 4 or less, indicating that the last vcb has been tested, the accumulator is tested to determine if the entire set of vcb's for this channel have been tested. If all the vcb's have not been tested, MZ is reset to initiate another search from the beginning of the table and control returns to Block CG01. If the accumulator is minus, indicating that this second attempt has already been performed, control transfers to Block CF17.

Block CG06: This block is entered when an address of a select routine is found in word 2 of a vcb. The actual location of the vcb is stored into the priority cell and control transfers to Block CG01.

Blocks CG07 to CG09: The activity flag in vcb word 3 for a 7909 channel is tested to determine if a seek has already been issued for this unit and the seek has not responded with an attention signal. If an attention signal is still awaited, the sign of DCHSW is set to minus and MZ is stepped to test the next unit and control transfers to Block CG03. If an attention signal is not awaited, MZ is reset to test for a select routine in this vcb, and control transfers to Block CG02.

Blocks CG10 to CG13: The trap switch and DCMSW are cleared. The sense indicators and MZ are restored. FRSW1 is tested to determine if the program may return to the point interrupted by the trap at this time. If FRSW1 is not zero, control transfers to Block CG14. Otherwise, FRSW2 is cleared, and the remaining index registers, together with the original contents of the accumulator and MZ, are restored. The overflow indicator and trap transfer status are restored. Traps are restored to their original state, and control returns to the point interrupted by the original data channel trap.

Blocks CG14 to CG16: FRSW1 is cleared and FRSW2 is tested to determine if the program must return to complete processing the trap that was interrupted by an on-line message. If FRSW2 is not zero, control returns to XTRAP, Block CG04, to complete processing of the first trap. If FRSW2 is zero, FRSW1 is cleared, and control transfers to PRX2, Block CW32.

Read Redundancy Recovery Routine

The read redundancy-recovery routine for tape performs the following major operations: determines whether a record was a noise record and returns to the trap supervisor if it was a noise record, performs a tape cleaner operation on the tape every ten times if the noise record bit is off; and continues trying to recover until the maximum number of tries as indicated in AWRX1 has been reached.

The tape cleaner operation will not be performed if the record count is less than 2 in this particular file.

Blocks CH01 to CH05: The proper TRC instruction is picked up from the TRCX table, and inserted in this routine. (As assembled, the listing shows a TRCA, but the operation code will be modified for the proper channel by this procedure.) The TRC instruction is executed to determine if there has been a tape redundancy. If there has not been any redundancy, control transfers to END16 (Block CS08). If there has been a redundancy, the redundancy check control cell for this channel in table BCTX is tested to determine if redundancy recovery is disabled. If the entry in BCTX is zero, indicating that redundancy recovery is disabled, control transfers to END16 (Block CS08). If redundancy recovery is allowed, subroutine WTRX is used to determine if the record in question is a noise record. If bit 0
in the sense indicators is on upon return from this subroutine, a noise record is indicated and control transfers to END15 (Block CE07).

**Blocks CH06 to CH09:** When a record is not noise, subroutine RED50 is used to increment the redundancy count and set up RED14 and the accumulator with the unit address for recovery procedures. IN2 will contain the count of the number of redundancies upon exit from this subroutine. The unit address is placed in the read select instruction for recovery procedures, and IN2 is tested to determine if the limit for the number of retries has been reached. This limit is indicated by RDRN1. If this limit has not been passed, control passes to RED14 (Block CH14), to backspace the tape. Otherwise, the record count is incremented by 1.

**Blocks CH10 to CH12:** Bits S and 1 of word 3 of the UCB are tested to determine if the redundancy message is to be suppressed. If either bit is on, control transfers to Block CH13. If both bits are off, the message will be printed. Subroutine RED10 is used to print the redundancy message that indicates the tape unit number as well as the tape position where the error was encountered. The permanent redundancy bit is set in the VWRX table.

**Block CH13:** Bit 2 of the sense indicators is set to 1 to indicate to SEL—there has been a permanent redundancy, and control transfers to END14 (Block CE12).

**Blocks CH14 to CH16:** The tape is backspaced over the record that caused the redundancy. IN2, which contains the count of the number of redundancies, is reduced by 10 until it is 10 or less. IN2 is then tested to determine if this retry is the first attempt, or some multiple of ten plus one. If IN2 does not meet these conditions, control transfers to END13 (Block CE20).

**Blocks CH17 to CH21:** On the first entry, or any entry that is a multiple of ten plus one, word 3 of the UCB is picked up. This word contains the tape position with the noise record indication in the sign position. The sign is tested to determine if there has been a noise record. If there has been one, control transfers to END13 (Block CE20). For no noise record, the record count in UCB word 3 is placed in IN14 and tested to determine if the count is less than or equal to 1. If a record count less than or equal to 1, control transfers to END13 (Block CE20) to bypass the tape cleaner operation.

If IN14 is greater than 1, a tape cleaning operation may be attempted depending on the record count. IN14 is further tested to determine if the record count is in complement form indicating that the tape is positioned at an end-of-file mark. The tape cleaner operation will be bypassed, and control transfers to END13 (Block CE20).

**Blocks CH22 to CH25:** If a tape cleaner operation is specified, a **BRR** instruction is executed twice and then an **RRE** is executed twice. The channel is tested to determine when this operation is completed. When the channel is no longer busy, the redundancy indicator in the channel is turned off (using a **TNC** instruction) and control transfers to END15 (Block CE20). The purpose of this series of instructions is to move the tape backwards in an attempt to bring the bad area of the tape into the vacuum column and dislodge any particles that might have caused the read redundancy.

**Write Redundancy Recovery Routine**

The write redundancy recovery routine first backspaces the tape over the bad record, and then writes an erase area and checks it for redundancies. If redundancies are found while writing the erase area, a message is printed on-line for the operator. If no erase redundancies are found, the routine rewrites the record that could not be written earlier. This series of operations is repeated until the record can be written properly or until an **EOR** condition is sensed. After each 25 erase areas are written, a message is printed on-line for the operator.

**Blocks CJ01 to CJ05:** The accumulator is tested to determine if there had been a redundancy encountered during an erase operation. If the accumulator is minus, indicating a redundant erase, control transfers to RED50 (Block CJ15). If the redundancy was not encountered during an erase operation, the sign of the accumulator is set minus, and the accumulator stored in the channel activity cell for the proper channel. Subroutine RED50 is used to increment the redundancy count. Upon exit from this subroutine, the accumulator and RED14 contain the unit address and IN2 contains the redundancy count. The unit address is placed in RED6 for recovery operations to follow.

**Blocks CJ06 to CJ09:** The tape is backspaced over the bad record and IN2 is tested to determine if this is the first redundancy. If this is the first redundancy, control transfers to Block CJ10. If this is not the first redundancy, the **EOR** indicator in the channel is tested. If the **EOR** has been sensed, control transfers to RED9 (Block CJ15). If the **EOR** has not yet been reached, the redundancy count in IN2 is reduced by 25 until it is 25 or less.

**Blocks CJ10 and CJ11:** IN2 is tested to determine if the count remaining is exactly 25. This would be the case on every 25th redundancy encountered. If IN2 is not exactly 25, control transfers to RED5 (Block CJ12). If the count is 25, an on-line message is printed for the operator, indicating that 25 erases occurred during a recovery attempt.
Blocks CJ12 to CJ14: The address of an instruction which will cause a blank area of tape to be written is stored in the RCHX table. Traps are disabled, the tape unit is selected, and the RCHX instruction is executed to write the blank tape. Bit 17 in the URAX table entry for this channel is set to 1 to indicate that an erase area has been written. Control transfers to OVT (Block CE10).

Blocks CJ15 to CJ17: These blocks are entered when an EOT is sensed during an attempt to recover from a redundant write. The unit address is picked up from the UCB, and subroutine CVFPR is used to print this tape address and a message to the operator indicating that the end of tape has been reached. The STOP subroutine is used to bring the program to a dead stop.

Block CJ18: This block is entered from Block C91 if a redundancy was encountered attempting to write an erased area. Subroutine RED10 is used to print an on-line message to the operator to inform him of his problem and control transfers to EN13 (Block CE20).

Redundancy Subroutines

Chart cx shows four subroutines used by the redundancy recovery routines. Subroutine RED50 is used to increment the redundancy count, and set up the I/O commands for the proper unit and channel in operation. Subroutine RED10 is used to print an on-line error message to the operator informing him of an error, and giving the tape unit number and the position of the error on the tape. Subroutine NRTST is used to compute the length of the record in question, and determine if it is a noise record. If the word count is less than or equal to 2, it is considered to be a noise record, and the noise record indicator is reset in the sense indicators. Subroutine RDUN is used to set the proper bit in the TRAP cell to allow redundancy trapping on a write operation.

RED50 — Increment Redundancy Count Subroutine

Blocks CK01 to CK03: The redundancy count in the URAX table is incremented by one, returned to the table, and also placed in IRA. The unit address is obtained from the UCB and placed in RED14 and the accumulator. Control exits to the calling program via 1, 4.

RED10 — Print Redundancy Error Message Subroutine

Blocks CK04 to CK13: IRA and IRI are saved. The address of the available part of the message is obtained from the RED10 calling sequence and placed in IRA. The variable part of the message is then obtained and placed into the print image for the error message. The file position is picked up from word 3 of the UCB, converted to BCD by subroutine BCVDEC, and stored in the print image. The record position is next picked up from word 3 of the UCB, converted to BCD by subroutine BCVDEC, and stored in the print image. The tape unit number is picked up from word 1 of the UCB. Subroutine CVFPR is entered with the tape number in the left half of the MQ. This subroutine converts the tape number to its symbolic unit designation, and prints the error message desired with this tape designation. Upon return from CVFPR, IRI and IRI are restored, and control returns to the calling program via 1, 4.

NRTST — Noise Record Test Subroutine

Blocks CK14 to CK16: This is the entry point for the NRTST subroutine. This subroutine checks the length of a record just read or written to determine if it was a noise record. IRI is saved and the channel data is picked up from COMM. Then the original I/O command is subtracted from this.

Blocks CK17 to CK19: The resulting word count is placed in IRI and tested to determine if it is less than or equal to 2. If it is less than or equal to 2, bit position 0 of the sense indicators is cleared to indicate a noise record condition. If the word count is greater than 2, the sense indicators are not changed. Control exits to the calling program via 1, 4.

RDUN — Permit Write Redundancy Trap Subroutine

Blocks CK20 to CK23: IRA is saved. The location of the UCB is picked up from CRIXAC and used to obtain word 2 of the UCB for the unit now in use. The sign of word 2 is tested to determine if this is a read or a write operation. If the operation is a read operation (sign is 0), control transfers to Block CK21. If the operation is a write operation (sign is 1), a bit is on'd to the TRAP cell to allow redundancy trapping on this one particular channel.

Block CK24: IRA is restored, and control returns to the calling program via 1, 4.

NDSEL — Non-Data Select Routine

Chart cx shows the non-data select routine that is used to perform on a tape unit specific operations that do not cause the transfer of data. A test is made for the validity of the UCB as specified by the FILE designation, in the calling sequence. If the unit reference is illegal, the program will come to a halt after printing the reason for the halt. An attempt to perform a non-data operation on a disk file will not cause the error halt, but will cause an exit from the routine with no operation performed.

Block CL01: Subroutine TEST is used to test the validity of the file specified in the calling sequence. An invalid UCB address will cause an error halt, and
the subroutine will not return to the non-data select routine.

**Blocks CL02 to CL07:** This may also be used as an entry point if it is not desired to test for a valid file designation. Traps are disabled, and word 1 of the UCB is picked up. The trap switch, TRAPSW, is tested to determine if this non-data select request is during trap time or non-trap time. If TRAPSW is zero, indicating that it is non-trap time, INA1, INA2, and the sense indicators are saved for the non-trap exit. Location WEF1 is set to the location of the UCB. If the trap switch is not zero, the save and setup instructions for the non-trap exit are bypassed. In either case, however, INA1, INA2, and the sense indicators are saved for the trap exit, and location WEF2 is set to the location of the UCB.

**Blocks CL08 to CL11:** INA1 is set to the non-data select code as indicated in the calling sequence. The exit address is calculated for all operations except the write end-of-file operation. INA2 is tested to determine if the operation requested is WEF. If it is a WEF operation, the exit address is modified for this particular use of the subroutine. (INA1 contains the proper exit address at this time.) If the operation is not WEF, no change is made to the previously calculated address in INA1.

**Blocks CL12 to CL14:** Following the write end-of-file check, INA2 is tested to determine if the operation requested is NOP. If the operation is NOP, control transfers to Block CL31 to return to the calling program. If the operation is not NOP, the unit address from word 1 of the UCB is loaded into the sense indicators and tested to determine the type of unit. If the unit is card or disk, control transfers to Block CL31 for an exit to the calling program.

**Blocks CL15 and CL16:** For a tape unit, TRAPSW is tested to determine if this operation is during trap time or non-trap time. If this operation is during trap time, control transfers to Block CL17. If this operation is during non-trap time, word 2 of the UCB is tested to determine if the unit is active. If word 2 is zero, indicating that the unit is not active, control transfers to NDS1 (Block CL20).

**Blocks CL17 to CL19:** If the unit is active, subroutine ACTV is used to assign channel priority to this unit. The subroutine is entered in such a manner to bypass testing for the validity of the file, since this has already been done when the non-data select routine was entered initially. Word 2 of the UCB is tested to determine when the unit activity has ceased. Until the unit activity has ceased, control remains within this test loop. When all activity on this unit has ceased, traps are disabled.

**Blocks CL20 to CL22:** INA2 is tested to determine if the operation is WEF. If the operation is WEF, control transfers to Block CL23. If the operation is not WEF, bit 18 of the sense indicators is tested to determine if the end-of-tape marker has been sensed. If an end-of-tape marker has not been sensed, control transfers to Block CL23. Otherwise, INA1 is reset to the location of the error exit that is to be taken when a WEF operation is requested at the end of tape.

**Blocks CL23 to CL27:** TRAPSW is tested to determine if the program is in trap time. If TRAPSW is not zero, indicating trap time, control transfers to Block CL28. Otherwise, the non-data operation code is stored in UCB word 2 with the address of the select routine for this operation. Subroutine ACTV is entered at a point which bypasses the file validity check and assigns channel priority to the required unit. Upon return from ACTV, the contents of INA1 are stored as the exit address. INA2, INA4, and the sense indicators are restored, and control will return to the calling program via either 1, 4, or 2, 4 as computed earlier. (The normal return is 1, 4 for all except WEF operations. This operation exits to 2, 4 normally and 1, 4 if an EOT has been sensed.)

**Blocks CL28 to CL30:** A comparison of the channel presently in trap time and the channel required for the unit designated to receive the non-data operation is made. If both channels are not the same, control transfers to NDERR (Block CL33). If both channels are the same, the non-data code is placed in the accumulator, and INA2 is set to point to an indirect reference to the UCB. A direct entry is made to the select routine at Block CM02 to initiate the non-data operation requested.

**Blocks CL31 and CL32:** INA1 is stored as the exit address as previously calculated. INA1, INA2, the sense indicators, and trap status are restored. Control exits to the calling program at either 1, 4 or 2, 4 as required for the particular operation and conditions.

**Blocks CL33 to CL35:** If the unit channel and trap channel are not the same, the unit address is picked up to be used in the print-out of an error message for the operator. Subroutine CVFPR is used to print an on-line error message indicating that there was an illegal non-data select at trap time. The symbolic designation for the requested tape unit is also printed out with this message. After the message is printed out, subroutine SROP causes a dead stop.

**Select Routine for Non-Data Operations (Except NDWEF)**

Chart CM shows the select routine used for all the non-data operations initiated by NDSEL except for the NDWEF operation which is shown on Chart CN. Portions of the chart are shown within boxes. One or the other of these boxes will not effectively exist in any particular program, depending on the value given to the symbol "INM" on the EQU card at the beginning of the symbolic
deck. The following instructions will overlay the undesired instructions in one box as a result of the nss pseudo-instruction with the symbol IBM in its variable field. The instructions which are conditional pertain to the set density operations that are not legal on the IBM 709 computer. If the computer for which the program was assembled is an IBM 709, a test is made for either of the set density operations. Control exits without performing these operations if either of the operations is requested. The instructions pertaining to their execution are overlaid so that space is not lost in the assembly due to unrequired instructions.

**Block CM01:** HR2 is set to point to the UCB, and word 2 of the UCB is picked up. The original word 2 of the UCB is cleared.

**Blocks CM02 to CM05:** HR4 is saved, and the non-data code is placed in HR4. The unit address is obtained from word 1 of the UCB and positioned in the accumulator address. HR4 is tested to determine if the non-data code is SBNL. If the non-data code is SBNL, control transfers to Block CM06. If the non-data code is any code other than SBNL, a bit is on to position 31 of the accumulator to force the binary address which is required for all other non-data operations.

**Block CM06 to CM09:** The unit address is stored in the non-data select table, SELX. Block CM07 is only present if the program was compiled with IBM EQU 709. In this block, HR4 is tested to determine if the non-data code is either of the set density operations. If either of the set density operations has been requested, control transfers to NDSK1 (Block CM28). If the non-data code is neither of the set density operations, (or if Block CM07 does not exist in this assembly), HR4 is tested to determine if the non-data code is WEW. If the non-data code is WEW, control transfers to the WEW select routine at NDWFEF (Block CM01). If the operation is not WEW, the desired operation in the SELX table is executed.

**Blocks CM10 to CM13:** This block is entered after the desired operation has been executed, or from the alternate select routine for a WEW operation as shown on Chart CN. Word 1 of the UCB is placed in the sense indicators, and word 3 is placed in the accumulator. A long right shift with a count of zero is executed to save the noise record indication in the sign position of the MQ.

The sign of the accumulator is made plus in case the sign was previously set minus to indicate a noise record condition. The next instruction is an execution into an indexed table of instructions. Each of the instructions in this table will perform a different function that is required for the non-data operation being performed.

Any of the tape movement operations will cause the new tape position to be calculated. Either of the set density operations will cause a transfer to a routine which will properly set the density indicators in word 1 of the UCB. An SDNL operation causes a transfer of control to UPD6 (Block CM22) and an SDNH operation causes a transfer of control to UPD7 (Block CM24). A tape movement operation causes the new tape position to be calculated and left in the accumulator.

**Blocks CM14 to CM16:** The accumulator is tested to determine if the operation resulted in a minus tape position. If the tape position is not minus, control transfers to Block CM17. If the tape position is minus, this position is tested to determine if it is also zero, which indicates that the tape is rewound and unloaded. If the tape is rewound and unloaded, control transfers to Block CM18. If the tape is not rewound and unloaded, the accumulator is cleared.

**Block CM17:** The accumulator is tested to determine if the tape is positioned at load point. If the tape is at load point, control transfers to Block CM19.

**Blocks CM18 and CM19:** At this point it is known that the tape is positioned at some point other than the beginning of the tape, and that it is not unloaded. The end-of-tape indicator is turned off, if it is on. The noise record indication which was stored in the sign of the MQ is replaced in the accumulator, using a long left shift with a count of zero.

**Blocks CM20 and CM21:** The calculated tape position in the accumulator is replaced in word 3 of the UCB. A test is made to determine if the tape is rewound but not unloaded. If the tape is rewound but not unloaded, control transfers to UPD3 (Block CM26).

If the tape is not rewound, control transfers to NDSK1 (Block CM28). (If the program is compiled for a system with IBM EQU 709, Block CM21 and the test do not appear. Control will pass from Block CM20 directly to Block CM27 in this case.)

**Blocks CM22 and CM23:** The bit indicating current density is cleared. (This indicates low density.) The operation code in HR4 is tested to determine if the operation was an SDBL. If the operation was an SDBL, control transfers to Block CM25.

**Block CM24:** This block is entered if the non-data operation code is SDBH. The density bit indicating density at current position is set to 1 to indicate high density.

**Block CM25:** With the indicator for the density at the current position set up, the tape position in word 3 of the UCB is tested to determine if the tape is at load point. If the tape is not at load point, control transfers to Block CM27.

**Block CM26:** If the tape is at load point, the density bit in position 19, which indicates load point density,
is set equal to bit 20, which indicates the density at the current position.

*Block CM27:* Word 1 of the ucb is restored from the sense indicators with the corrected density indications if they had been made.

*Block CM28:* IN4 is restored and control exits to the calling program via 2, 4.

**Select Routine for NDWEF**

Chart CN shows the select routine used for WEF operation. It is entered from the select routine shown on Chart CM for a WEF operation. The routine writes an end-of-file on a desired tape and checks for a redundancy. If a redundancy is encountered, the tape is backspaced, and another attempt is made. On all except the first attempt, a check is made for an end-of-tape condition. Every 25th time that an attempt is made, an on-line message is printed for the operator indicating that there have been 25 erases during a write.

*Block CN01:* This block is entered from the non-data select routine with the unit address in the accumulator. This address is stored in the i/o command requiring it. The TRC operation code for the proper channel is picked up from the TRCX table, and inserted in the TRC instruction in this routine. IN4 is set to count 25 attempts.

*Blocks CN02 to CN05:* The WEF instruction is executed, and the proper entry in the TRCX table is set to loop within this program until the channel is not busy. When the channel is not busy, the channel redundancy check indicator is tested. If a redundancy has occurred while writing the end-of-file, control transfers to WEFS (Block CN09).

*Blocks CN06 to CN08:* If no redundancy occurs, the end-of-tape indicator in the channel is tested. If the indicator is on, bit 18 in word 1 of the UCB is set to indicate EOT. If the indicator is off, this step is bypassed. IN4 is reset to the non-data operation code, and control returns to NSDS (Block CM10) in the regular non-data select routine.

*Blocks CN09 to CN11:* These blocks are entered if there has been a redundancy on writing an end-of-file. The tape is backspaced and IN4 is tested to determine if this is the first time the error has occurred. If this is the first time the error has occurred, control transfers to Block CN15, bypassing the end-of-tape test. If this is not the first time the error has occurred, the end-of-tape indicator in the channel is tested. If the end-of-tape has been sensed, control transfers to WEF8 (Block CN15).

*Blocks CN12 to CN14:* These blocks are entered on the first attempt or on any other attempt to write an end of file if an EOT was not sensed. IN4 is tested to determine if 25 attempts have been made. When 25 attempts have been made, subroutine REDIO prints an on-line message indicating that there have been 25 erases during write. IN4 is then reset for 25 more attempts. If 25 attempts had not been made, the print-out is bypassed and IN4 is not reset. Control returns to WEFS+1 (Block CN02).

*Blocks CN15 to CN17:* These blocks are entered when an EOT is sensed on any attempt after the first attempt to write an end-of-file. The tape unit address is placed in the XQ. Subroutine CVPTR is used to convert this unit address to a symbolic designation for the unit and print a message indicating an end-of-tape sensed during an erase. Subroutine STOP is then used to provide a dead stop halt.

**Disk Recovery Routines**

Charts CE and CN show the recovery routines used when errors are encountered using disk. One of four corrective actions is used to recover from an error. Action 1 attempts the same command sequence one additional time; if the same error occurs the second time SEL- is entered with the appropriate indication in the sense indicators.

Action 2 attempts the same command sequence up to four times. Two seeks are then issued to recalibrate the arm. The original seek issued by the user is repeated and up to four more attempts are made to perform the operation. If the same error occurs on the fourth attempt after the second seek, SEL- is entered with the appropriate indication in the sense indicators.

Action 3 attempts the same command sequence up to four times. If the same error occurs after the fourth attempt, SEL- is entered with the appropriate indication in the sense indicators.

Action 4 issues a seek to move the arm from the CE track. The original seek requested by the user is re-issued, and SEL+ is re-entered once to attempt the command sequence again. If the same error occurs, SEL- is entered with the appropriate indication in the sense indicators.

Figure 3 shows the recovery action taken for each type of error encountered.

*Blocks CP01 to CP05:* These blocks are entered from XTRAP when an Unusual End is received during a disk operation. Bit 35 of the sense indicators is set to 0, and the redundancy count from UBRX, 1 is placed in IR2. The SNSNTA bits are picked up and masked out as indicated by the the contents of RCTX, 1 which contains the address of a mask of the bit positions that are to be ignored. If RCTX, 1 is 0, all the bit positions in
Blocks CP17 to CP19: These blocks are entered to take Action 3. The mask for the data check bits is picked up and placed in the MQ. The accumulator is tested to determine if the errors are the same as the last time. If the errors are not the same as the last time, control transfers to Block CP12 to store the new error information and issue another seek. If the errors are the same, M2, which contains the error count, is tested to determine if there have been four attempts to correct the same error. If there have been four attempts, control transfers to DUEN2 (Block CP21) for an error message print-out. If there have not been four attempts, control passes to DUEN1 (Block CP13) to issue another seek.

Blocks CP20: This block is entered whenever Action 1 is required, and the errors are the same as those last obtained. M2 is tested to determine if this is the first error. If this is the first error, control transfers to DUEN1 (Block CP13) to issue another seek.

Blocks CP21 to CP23: These blocks are entered whenever it is necessary to print an on-line error message. The unit address is converted using subroutine SYUNC, and stored in the message image. Word 2 of the VCB is tested to determine if this operation is a read or write operation, and the appropriate message is added to the error message. The track address is picked up from words 3 and 4 of the VCB and stored in the error message.

Blocks CP24 to CP27: The SNSDTA bits are compressed to remove those bits in the word that have no meaning. These compressed bits are examined one bit at a time, and a composite error message is built up containing all the error indications found. Subroutine PROUT is used to print this message, and the sense indicators are restored before returning control to the trap supervisor at DENG2 (Block CD16).

Blocks CR01 to CR05: This is the entry point to perform Action 2. The mask for the bit that indicates no-record-found is loaded into the MQ. The accumulator is tested to determine if the errors are the same as those that occurred on the last attempt. If the errors are different, control transfers to Block CP12 for another seek and an entry into SEL+. If the errors are the same, M2 is tested to determine if there have been less than four errors. With less than four errors, control transfers to DUEN1 (Block CP13) to eventually re-enter SEL+. If the error count is four or greater, M2 is tested to determine if the count is eight or greater. When such a count is indicated, control transfers to DUEN2 (Block CP21) for an error message print-out. If the count is less than eight, M2 is next tested to determine if there are more than four errors. Because prior tests have determined there were not less than four errors nor greater
than seven errors, this test will indicate if there are exactly four or from five to seven errors. If the error count is from five to seven, control transfers to DUEN1 (Block CP13) to clear out the error indications and eventually return to SEL+. If there were exactly four errors, the original track address is saved and an illegal address placed in the UCB. IR4 is set to zero to cause two seeks to be issued before reissuing the original seek requested by the user. Control then transfers to DUEN7 (Block CR13).

Blocks CR09 to CR12: This is the entry point to perform Action 4, which is required for an invalid address type of error. The mask for the bit indicating invalid address is placed in the M9. The accumulator is tested to determine if the errors are the same as those that occurred on the last attempt. If the errors are different, control transfers to Block CP12 for another seek and an entry into SEL+. If the errors are the same, IR5 (which contains the number of recovery entries into SEL+ as found in URBX, 1) is tested to determine if the count equals zero. If the count is zero, control transfers to DUEN1 (Block CP13) for a return to SEL+. If more than one recovery attempt has been made, IR4 is set to two to indicate that another seek is to be issued to re-calibrate the arm. IR5 is set to point to the UCB.

Blocks CR13 and CR14: Words 3 and 4 of the UCB are set to minus. The sign of word 3 (minus) indicates to IOEX that a seek is requested by the user. The sign of word 4 (minus) indicates to IOEX that the track address is already contained in words 3 and 4 in BCD. Therefore, IOEX does not have to convert the binary address specified by the address in word 4. IR4 is saved and subroutine ISK1 is entered to issue one seek.

Blocks CR15 and CR16: Attention interrupts and traps are enabled on this one channel, and the main program loops, waiting for the interrupt.

Blocks CR17 to CR19: This is the point at which control is returned from the SAVE routine after an interrupt. Subroutine DATNS is used to process all attentions received. The AF flag is turned off by this routine for each attention received. The desired AF flag is tested to determine if the proper attention signal was received. If the desired signal was not received, control transfers to DUEN5 (Block CR13) to enable interrupts again and wait for the next trap. If the desired AF flag is now off, IR4 is reset to the value of “K”. The value of “K” is used to determine if the proper number of seeks have been issued in an attempt to re-calibrate the arm before reissuing the original seek requested by the user.

Blocks CR20 to CR22: IR4 is tested to determine if the value of “K” is greater than 4. This would occur after the second seek is received during an invalid address recovery. In this case, control transfers to Block CP15 to reissue the original seek and try the operation again. If “K” is not greater than 4, IR4 is tested to determine if “K” is equal to 0. “K” would be equal to 0 if the recovery attempt is for a no-record-found error. If “K” is not zero, indicating that the first recalibrate seek has been issued for a no-record-found recovery, control transfers to DUEN10 (Block CR23). If “K” is equal to zero, the original track address is replaced in words 3 and 4 of the UCB.

Block CR23: If “K” is less than 4 and greater than or equal to 0, it is incremented by 3, and control transfers to DUEN7 (Block CR13) to eventually issue another seek.

Issue Seek Routines

The issue seek routines will issue one seek for an arm whose UCB is indicated, or will issue seeks for all arms on the channel that have their AF flag on and their PF flag off. The sign of word 4 of the UCB indicates to the routine the status of the track address contained in the UCB. If the sign of word 4 is +, the binary number contained in the address of word 4 is converted to BCD to form the track address. If the sign is −, the routine considers that the track address has already been converted to BCD and appears in words 3 and 4 of the UCB. Upon exit from the routine, the sign of UCB word 4 will always be reset to +.

ISSK1 — Issue One Seek Routine

Block CS01: This is the entry point for one seek. IR1 is saved, the SRCX instruction is picked up from the SRCX table, and IR1 is set to 1.

ISSEK — Issue Multiple Seeks Routine

Blocks CS02 to CS05: This is the entry point if the user desires to issue seeks for all arms with their AF flag on and PF flag off. IR2 and IR4 are saved; IR1 is set to the location of the channel program, which is in the accumulator at this time for an ISSK1 entry. The accumulator is tested to determine if the entry is from ISSK1 or ISSSEk. If the accumulator is plus, the entry is from ISSK1 and control transfers to ISS1 (Block CS06). If the accumulator is minus, the entry is from ISSSEk. IR1 is saved, and IR2 is set to point to the first disk UCB for this channel. IR1 is set to the location of the channel program, and IR1 to the number of arms on the channel.

Blocks CS06 to CS11: The seek request flag, AF, and the seek issued flag, PF, are picked up from word 3 of the UCB for both the ISSK1 and ISSSEk routines.

These flags are tested for the following condition: AF ON, and PF OFF. If this exact condition is not met, control transfers to ISSK (Block CS13). If this condition is met, word 4 of the UCB is picked up. The sign of
word 4 is tested to determine if the track address contained in this vcb is already converted to bcd. If the sign is 1, indicating that the address is already in bcd, control transfers to iiss3 (Block CS12). If the address has to be converted, subroutine bcdnvdc is used to perform the conversion, and the resulting bcd address is stored in words 3 and 4 of the vcb. The two high-order characters are placed in word 3 and the two low-order characters are placed in word 4.

Blocks CS12 to CS14: Word 4 of the vcb is set to plus, and iir2 is saved. A c7l instruction with its address set to the location of word 3 of the vcb is generated, and placed in the channel sequence. The pf flag is turned on.

Blocks CS15 to CS18: iir1 is tested to determine if there are any more vcb's to check. If there are more vcb's to be checked, (as on an issek entry), control transfers to iiss1 (Block CS06) to check the af and pf flags of this vcb. Otherwise, the wtr instruction is placed into the channel sequence as the last instruction. iir1 is restored, and the desired sdcx (store diagnostic channel) instruction is placed in iiss7. The desired stcx instruction is placed in the bchx table.

Blocks CS19 to CS24: The bchx instruction is executed, and the tcox instruction in the tcox table is set to return to this routine. The tcox instruction is executed to determine when the channel completes its operation. When the operation is complete, the sdcx instruction is performed to store the results of this operation in isshe. The contents of isshe are picked up and the bits indicating an unusual end condition are extracted from this word. The accumulator is then tested to determine if there was an unusual end on the last seek operation. If there was an unusual end, control transfers to iiss8 (Block CS25). Otherwise, iir2 and iir1 are restored and control returns to the calling program via 1, 4.

Blocks CS25 to CS28: On an unusual end, the old setting of the enable switch, enbws, is saved in isshe temporarily, and enbws is set. Subroutine prout is used to print an on-line message indicating an unusual end during a seek. Subroutine pause then provides a temporary stop to allow for operator intervention. When the operator depresses start, subroutine prout is used to print the second line of the message. The original setting of enbws and iir1 are restored. The entry switch, pstrw, is cleared and enbws is tested to determine if traps should be enabled upon exit from this subroutine. If enbws is zero, trpsw is tested. If trpsw is also zero, traps are enabled. If trpsw is not zero or if enbws is not zero, traps are not enabled. Control exits to the calling program via 1, 4.

STOP Routine

Blocks CT11 to CT14: This is the entry for the STOP routine, which is used to provide a halt from which there is no normal recovery. Traps are disabled, iir4 is saved, and the enable switch, enbws, is set. Subroutine prout is used to print the on-line message CANNOT PROCEED.

After returning from subroutine prout, enbws is cleared, iir4 is restored and control transfers to a "halt-transfer" instruction which will retain control each time start is depressed. The only recovery from this halt would be to manually transfer to one of the locations in the nucleus that would restart the program. One typical use would be to manually transfer to location 115, which would call the dump program and restore ibsys after completion of the dump.

Machine Pause and Stop Routines

Chart CT shows the two routines in tcox used to provide a temporary stop for operator action, and a dead stop from which there is no normal recovery. Subroutine pause has two entry points: one for each of two possible on-line messages that may be printed before and after the temporary stop is received. Subroutine stop prints one on-line message indicating that the program cannot proceed, and transfers to a "halt-transfer" instruction which transfers to its own location.

PAUSE/PAWS Routines

Block CT01: This is the entry point labeled pause. A message will be printed which accompanies a request for operator intervention. Traps are disabled, and the entry switch is set to provide the desired message print-out.

Blocks CT02 to CT06: This is the entry point labeled paws. A message will be printed which accompanies a notice to the operator of some difficulty that might require termination of the program. The operator generally can either ignore the problem and continue with the program or delete the job. Traps are disabled, iir4 is saved, and the setting of the enable switch (enbws) is saved. (Disabling traps is redundant if the entry was at pause, but is provided at both entries to prevent a trap from interrupting the sequence once it has started). enbws is cleared to prevent the trap supervisor from enabling traps when the print-out is completed. The entry switch, pstrw, is tested to determine which entry was used. The address of the desired message is placed in the prout calling sequence and subroutine prout is used to print this message on-line. A "halt-proceed" causes a temporary program stop to allow the operator to check what has been printed and take any action that might be indicated.

Blocks CT07 to CT10: When the operator depresses start, subroutine prout is used to print the second line of the message. The original setting of enbws and iir4 are restored. The entry switch, pstrw, is cleared and enbws is tested to determine if traps should be enabled upon exit from this subroutine. If enbws is zero, trpsw is tested. If trpsw is also zero, traps are enabled. If trpsw is not zero or if enbws is not zero, traps are not enabled. Control exits to the calling program via 1, 4.
TEST — File Validity Test Subroutine

Subroutine TEST, shown on Chart CV, is used to test the validity of a file specified in an ACTV of NDSEL calling sequence. The address of the UCB specified is checked to determine if it lies within the range of the valid addresses. If the UCB address specified is valid, control returns to either ACTV + 1 or NDSEL + 1. If the unit specified was invalid, the address of the reference is converted to BCD and printed on-line with a message indicating an illegal unit reference. Subroutine STOP provides a dead stop halt, because this is a condition from which there is no normal recovery.

Blocks CU01 to CU04: Traps are disabled and IR is saved on entry to this subroutine. The address of the UCB is picked up from the file reference in the calling sequence, and placed in IR1. IR1 is then compared with the upper and lower limits of the valid UCB addresses to determine if the specified UCN lies within this range. If the UCB address is not within this range, control transfers to TST1 (Block CU08).

Blocks CU05 to CU07: If a valid UCB address is indicated, the availability flag in word 1 of the UCB is tested to determine if the unit is available. If the unit is not available, control transfers to TST1 (Block CU08). If the unit is available, the exit address is calculated using the value set in IR4 by the original transfer to ACTV or NDSEL. IR1 is restored and control returns to either ACTV + 1 or NDSEL + 1, as determined earlier.

Blocks CU08 to CU13: The effective address of 1, 4 is calculated using the value of IR4 as set by the original transfer to ACTV or NDSEL. This address is then converted to BCD using subroutine BCD5, and stored into the error message. ENBSW is set, and subroutine MWBR is used to print an on-line message indicating an illegal unit reference at the location determined. (MWBR is another entry label which provides the entry to subroutine PRUT.) Subroutine STOP provides a dead stop halt, because the error indicated would not normally allow the program to continue to completion. The operator at this time might be instructed to transfer to location 115 to provide a dump of the program. The dump program would return control to ISSYS after completion of the dump to enable the next job to be processed.

Perform SNS and Form Disk Order Subroutines

The DATNS routine on Chart CV is used to perform an SNS on a particular channel. IR1 contains the complement of the channel number minus one on entry to the routine. The subroutine will check the attention bits received by the SNS routine, and turn off both the AF and PR flags for all arms which received an attention bit.

The FDAMT routine on Chart CV is used to form a disk address and generate the disk order to locate this address. The calling sequence for the routine is shown on the chart along with the values required in IR4 and the MQ at entry time. The results at exit are also shown on the chart in the same block.

DATNS — Perform SNS Routine

Blocks CV01 to CV09: IR4 is saved to be used in returning to the calling program. IR2 is set to twice the value in IR1, to be used in locating the channel interrupt cell. The channel interrupt cell is set to transfer to the second entry point in the channel sense program. The proper STCX instruction is executed for this channel to start the SNS operation. The TCXO instruction is set to return to this routine, and then executed. When the channel is no longer busy, the channel interrupt cell is restored, and the attention bits that were received are packed and placed in the high-order position of the MQ. IR2 is set to point to the first disk UCB for this channel, and IR4 is set to the number of arms attached to this channel.

Blocks CV10 to CV13: Each attention bit is checked individually. If an attention signal was received, the AF and PR flags are turned off for that UCB. After the last bit is tested, IR4 is restored and control returns to the calling program via 1, 4.

FDAMT — Form Disk Order Routine

Blocks CV14 to CV20: This is the entry point for the FDAMT routine. The last four characters in the first word of the calling sequence are masked off in the calling sequence itself. The first two characters in the MQ are shifted off which leaves the BCD track address in the high-order positions of the MQ. The unit address is picked up from word 1 of the UCB and placed in the accumulator address.

The access and module are shifted into the MQ. The interface number is masked off and added to word 1 of the calling sequence. The access is shifted back into the accumulator, and zone bits are added to the module designation before shifting the access back into the MQ. The MQ now contains the BCD representation for AMTIT.

Blocks CV21 to CV26: A CRQ instruction is executed to convert all BCD zeros to 12. The 12 is the configuration required in the channel for a zero. The address of the second word of the disk order is calculated and placed in the instruction which will store this part of the order. All except the first two characters of the
original disk order are masked out. The characters denoting AMIT are shifted into the accumulator and added to the first two characters in the first word of the order. The two last characters of the track address are shifted into the accumulator, and combined with the characters denoting MAX in the calling sequence. This result is stored in the second word of the disk order. The character shown as "X" will appear in the second word of the disk order but is ignored by the channel. Control returns to the calling program via 3, 2.

**On-Line Print, Punch, and CVPRT Routines**

Chart Cw shows the routine used to print a line or punch a card on-line. In addition, this chart shows one of the smaller routines used by PROUT and the CVPRT routine used to print an error message concerning a particular tape unit.

Subroutine PROUT causes an immediate print-out of the desired on-line message before returning to the calling program. If the printer channel is active, the subroutine will wait for the trap on this channel only, and then print the message without processing the prior trap. After the message has been printed, the original trap will be processed.

Subroutine CVPRT generates a PROUT calling sequence for a desired message print-out. The tape number in the left half of the MQ is converted to its symbolic form and the phrase TAME XX is appended to the error message. Subroutine PROUT is used to print the completed message.

FBSW1, FBSW2, and FBSW3 are switches used for communication between the PROUT subroutine and XTRAP. They indicate that the PROUT routine is to be completed prior to processing a trap, and that there are one or more traps to be processed before returning to the original program that called the PROUT subroutine. There are four cases where the PROUT subroutine may be used with some configuration of channel A and another channel either active or trapped.

Case A consists of an entry to PROUT with channel A dormant. Traps are disabled on entry to the routine, so it does not matter if any other channel is active. The PROUT request is processed and control returns to the calling program after enabling the traps that had been disabled.

Case B consists of an entry to PROUT with channel A active and also trapped. This situation is handled in exactly the same manner as case A.

Case C consists of an entry to PROUT under the following conditions: (1) channel A active and not trapped, and (2) no other channel trapped. FBSW1 is set and traps are enabled on channel A only. The program then waits for the trap on channel A. When this trap is received, the channel information is saved in ACOMM, and the PROUT request is processed before completing the processing of the channel A trap. After the PROUT request has been processed, FBSW3 is set and control returns to the SAVE routine to complete the processing of the trap which was forced on channel A. After processing the trap, FBSW1 is cleared and because FBSW2 is not set, FBSW3 is also cleared. Control returns to PROUT so that PROUT may return to the program that originally made the PROUT request.

Case D consists of an entry to PROUT with channel A active and another channel trapped. FBSW1 is set and traps are enabled on channel A only. The program then waits for the trap on channel A. When this trap is received, the channel information is saved in ACOMM, and the PROUT request is processed before completing the processing of the channel A trap. After the PROUT request has been processed, FBSW2 is set and the subroutine exits to complete processing of the original channel trap (not channel A). When exiting from XTRAP at the completion of the original trap, FBSW1 is cleared, and because FBSW2 is not zero, control returns to XTRAP to process the channel A trap that had to be forced. XTRAP will now use the information that was saved in ACOMM at the time the trap was forced. At the completion of this trap processing, FBSW2 is cleared, and the normal exit from XTRAP in OUT will be taken.

**PROUT and PUNCH Routines**

*Blocks CW01: This is the entry point for the PUNCH routine. Traps are disabled and the punch switch, PUNSW, is set.*

*Blocks CW02 to CW07: This is the entry point for the PROUT routine. Traps are disabled, and the exit address is computed using the value in R1 and the value contained in the first word of the calling sequence. The exit address is stored as the regular exit address if FBSW3 is zero. If FBSW3 is not zero, indicating that there is another on-line message to be printed, the original setting of the exit address will not be changed. The value of R2 will also be saved in the proper exit location as indicated by FBSW3. R2 is then set to the number of calling sequence words as indicated in the first word of the calling sequence. FBSW1 is tested to determine if it is necessary to wait for a trap on the printer channel before starting the message print-out. If it is not necessary to wait for a trap, control transfers to Block CW12. If it is necessary to wait for a trap, the channel activity cell, CHXAC, is tested to determine if there is any activity on the printer channel. If there is no present activity on this channel, control transfers to Block CW12. If there is activity, the contents of TRPSW
are compared with chxac to determine if the printer channel is the channel that is trapped. If the printer channel is the channel that is trapped, control transfers to Block CW12.

Blocks CW08 to CW10: If the printer channel is not the channel that is trapped but is active at this time, rswsw is set and traps are enabled on the printer channel only. The main program waits for the channel trap. When the trap routine save is entered following a trap, rswsw is tested; because it is now not zero, control will return to the prout subroutine without processing the data channel trap.

Block CW11: This is the point at which control is returned from the save routine when it finds rswsw not zero. The printer channel information is temporarily saved in acomm.

Block CW12: This block is entered after a trap on the printer channel or whenever it was not necessary to force a trap on the printer channel. The accumulator is cleared.

Blocks CW13 and CW14: Subroutine sprr is used to generate the desired sprr or spwr instruction. Upon exit from this subroutine the accumulator will contain the desired instruction if the sprr specified in the calling sequence was not zero.

The accumulator is zero if no sprr was specified in the calling sequence. The channel is tested to determine if the channel is busy. While the channel is busy, control remains in this test.

Block CW15: When the channel becomes not busy; one calling sequence word is picked up from the calling program. nz will cause this instruction to step to each successive word as it is required.

Blocks CW16 and CW17: Subroutine bcdhc is used to convert the bcd image specified to the line or card image form. There are two returns from this subroutine. The return at 1, 4 indicates that the line image is not full, and the return at 2, 4 indicates that the line has been completed. If the return is made to 1, 4, nz is tested to determine if all calling sequence words have been converted. If all have not been converted, control returns to Block CW15 to pick up another calling sequence word.

Blocks CW18 to CW21: When the return from bcdhc is made to 2, 4 or the last calling sequence word has been converted, the print indicator, print, is cleared. print is used by subroutine bcdhc to indicate the column where the next portion of the image is to start. When print is cleared on an entry to bcdhc, that subroutine clears the print image, and resets its counters to start on a new line. After print is cleared, the punsw is tested to determine if the entry was to prout or punch. Either the printer or punch is selected for writing as determined by the results of this test. The location of the spwr instruction is tested to determine if an spwr instruction is required for this line of print. If this location is zero, it is skipped. If this location is not zero, the instruction will be executed. (Subroutine sprr places the desired instruction in the spwr location.) The image is printed or punched as required, and nz is tested to determine if there are any more calling sequence words in the list. If there are more in the list, control transfers to prtx (Block CW13) to set up the next spwr instruction and convert the next line.

Blocks CW22 to CW25: When the list of calling sequence words is completed, subroutine sprr is used to set up the last spwr instruction one more time. Upon return from sprr, the accumulator is tested to determine if an spwr instruction was requested. If none was requested, control transfers to Block CW16. Otherwise, punsw is tested again, and either the printer or punch is reselected as determined by the setting of punsw. The spwr instruction is executed, and the channel is tested to cause a loop until the channel is not busy.

Blocks CW26 to CW29: punsw is cleared to indicate that future entries to this subroutine will be considered entries to spwr unless they enter at punch and set punsw. Fsrsw1 is tested to determine if there might have been traps forced on the printer channel at the beginning of this subroutine. If Fsrsw1 is zero (no forced trap), control transfers to prtx (Block CW32), to set up the exit to the calling program. If Fsrsw1 is not zero, trnsw is tested to determine if the subroutine is being used during trap time. If trnsw is not zero, indicating trap time, control transfers to Block CW30 to set up the proper exit. During non-trap time, Fsrsw3 is set, and control transfers to save1 (Block CC32) to complete the processing of the printer channel trap which had been previously interrupted.

Blocks CW30 and CW31: During trap time, Fsrsw3 is tested to determine if there is another on-line message waiting to be printed. If Fsrsw3 is not zero, indicating another message is waiting, control transfers to prtax (Block CW31), to set up the exit to the calling program. If no additional message is waiting, Fsrsw2 is set.

Blocks CW32 and CW33: nz is restored, and trap status is restored to the condition it was when the subroutine was originally entered. Control transfers to the calling program at the point immediately after the last word of the calling sequence (N+2, 4).

Block CW34: If there is an on-line message waiting to be printed, nz is restored, and control transfers to the main routine which called prout. In this case, traps would not be restored since this use of prout is during a time when traps should not be enabled. The return is to the location immediately after the last word of the calling sequence (N+2, 4).
Blocks CW35 to CW39: This is the sprt subroutine used by prout. It is entered with the calling sequence word in the accumulator. The location spr is cleared to remove the old instruction. The accumulator is shifted right to move the bcd image location and word count out of the accumulator, and leave the desired spr code right adjusted. The accumulator is tested to determine if any spr code was requested. If none was requested, control exits to the prout subroutine via 1, 4. If some spr code is present, punsw is tested to determine if the resulting code is to be used for the printer or punch. As a result of this test, the basic sense instruction is added to the specific code to produce the desired sense instruction for that particular unit. The instruction thus generated is stored in the spr location to be executed when necessary.

(CVPRT Routine

Blocks CW40 to CW44: This is the cvprt subroutine used to print a tape error message on-line. It is entered with the tape unit address in the left half of the mq. i4 is saved, and the prout calling sequence is generated. Subroutine syuncv is used to convert the tape address to the symbolic form of channel/unit (i.e., A1, B4, C0, H4; RDA, PUC, PRE, EDH, PUF; CD00/0, AD00/1, BD00/1, FD00/1, DD00/0, EF00/1. This symbolic form is stored in the message image following the word unit. The message is printed along with the message indicated in the cvprt calling sequence by subroutine prout. At the completion of subroutine prout, i4 is restored, and control returns to the calling program via 2, 4.)
Chart CA. ACTV routine, Sheet 1
Chart CC. SAVE and XTRAP routines, Sheet 1
Chart CD. SAVE and XTRAP routines, Sheet 2
Chart CE. SAVE and XTRAP routines, Sheet 3
Chart CF. BG1 routine, Initiate Channel Operation, Sheet 1
Chart CG. BG1 routine, Initiate Channel Operation, Sheet 2
Chart CH. Read Redundancy Recovery routines
Chart CJ. Write Redundancy Recovery routines
Chart CK. Redundancy Recovery Subroutines
Chart CL. NDSEL, Non-Data Select routines
Chart CM  Select routines for NDSE1, except NDWEF
Chart CN. Select routine for NDWEF operation
Chart CP. DKRCV, disk recovery routine, Sheet 1
Chart CR. DKRCV, disk recovery routine, Sheet 2
Chart CS. ISSEK and ISSK1, Issue seek routines
Chart CT. PAUSE/PAWS and STOP routines
Chart CV. DATNS and FDAMT routines
The format of the following two lists of on-line messages is as follows: The first line or lines that are capitalized constitute the message as it appears in print. The next line indicates the block locations on the charts in the manual that show the logical flow that causes the message to be printed. The sentence or paragraph following this gives the reason for the message.

**During IBSUP Operation**

**CONTROL CARDS NEEDED IN CARD READER**  
OPERATOR ACTION PAUSE  

BCE3, BCF3  
An EOF has been sensed in the card reader while attempting to read in a control card.

**NOT A BASIC MONITOR CONTROL CARD**  
PRESS START TO CONTINUE  

BEH3, BEH4  
A card with an illegal operation code has been read in. The card in error is printed immediately before the error message.

**ILLEGAL SYSUNI DEFINITION**  
**PROVIDE CLARIFICATION IN CARD READER**  
OPERATOR ACTION PAUSE  

BJC2, BJF3, BJG3  
The system unit specified on either a $RELEASE or $AS card cannot be found in the SYNUM table.

BJE4/BJG4, BJF3, BJG3  
The disk origin or number of cylinders on a $AS card contains a character greater than 9. (In the released disk version, the same error condition occurs on a $RELEASE card.)

BKB2/BKD2, BJF3, BJG3  
One or both of the system units specified on a $SWITCH card cannot be found in the SYNUM table.

BHC4, BJF3, BJG3  
The system unit specified on a $REWIND, $SENDFILE, or $REMOVE card cannot be found in the SYNUM table.

**JOBS• STARTING WITH JOB XXXXXX• WILL BE PROCESSED**  

BUH1  
Sense switch 6 has been sensed down between jobs, indicating a skip job request.
IF INCORRECT, CLARIFY IN KEYS OR PUT $S.6 UP TO IGNORE REQUEST
OPERATOR ACTION PAUSE

BUC2, BUB2
A skip job request with a count greater than 15 (decimal) has been entered.

PLACE $S.6 UP
OPERATOR ACTION PAUSE

BUF3, BUG3
A skip job request has been interpreted and determined to be valid. A valid request would be a first request not greater than 15, or any second request.

ILLEGAL UNIT SPECIFIED
PROVIDE CLARIFICATION IN CARD READER
OPERATOR ACTION PAUSE

BFA2/BFD2, BJF3, BJG3
An illegal or invalid unit is specified on a ATTACH card.

BFA4/BFC4, BJF3, BJG3
An illegal or invalid unit is specified on a DETACH card.

MIXED HARDWARE SWITCH INVOLVING SYSLB1 IS ILLEGAL
PROVIDE CLARIFICATION IN CARD READER
OPERATOR ACTION PAUSE

BKJ2, BJF3, BJG3
The two units specified on a SWITCH card that involves SYSLB1 are not both of the same type, (i.e., both are not tape, or both are not disk).

SYSUT4 IS NOT ASSIGNED. NO DUMP CAN BE TAKEN

BWC4
In INITUS, the unit being initialized is SYSUT4 and is unassigned.

SYSOU1 IS NOT ASSIGNED. NO DUMP CAN BE TAKEN

BWC3
In INITUS, the unit being initialized is SYSOU1 and is unassigned.

CURRENT SYSTEM NAME MISSING
RESTORE WITH $EXECUTE SYSNAM IN SYSCRD.
OPERATOR ACTION PAUSE

BTB2/BTE2, BTF1, BTG1
syspos equals zero on an entry to SYSPC (IPRT), or the system name in SYSPOS cannot be found in the SYSNAM table.
***** UNKNOWN SYSTEM
PROVIDE CLARIFICATION IN CARD READER
OPERATOR ACTION PAUSE

BGG2, BVIC2/BVE2, BJF3, BJG3
Routine *EXCC used *FIND to locate correct system and could not find the desired system name in the *SYSNAM table, or when the name is found, the second word of the table entry is cleared.

BCF4, BVCL1, BJF3, BJG3
Routine *EXCC searched for system name in *SYSNM table and could not find it.

***** NO ASSIGNMENT MADE
PROVIDE CLARIFICATION IN CARD READER
OPERATOR ACTION PAUSE

BGG2, BVIC2, BJF3, BJG3
Routine *EXCC used *FIND to locate correct system and no assignment was made to the system unit required for a particular system.

BBH3, BCD2, BVIC1, BJF3, BJG3
During cold-start, the input tape is designated as the input unit and is unassigned or unattached.

BCD1, BCD2, BVIC1, BJF3, BJG3
In *SYSX, no input unit is assigned to the required input unit as designated by the input switch (*SYSICD or *SYSIN1).

$PAUSE
OPERATOR ACTION PAUSE
(MACHINE PAUSE)
ACTION COMPLETED

(BJG1), BJG2, CTE2, CTG2
A $PAUSE card has been read in during monitor operation.

$STOP
END OF JOBS
CANNOT PROCEED

(BJB1), BJA2, CTB4,
A $STOP card has been read in during monitor operation.

During IOEX Operation

OPERATOR ACTION PAUSE
(MACHINE PAUSE)
ACTION COMPLETED

CTE2, CTG2 (entered at CTA2)
A user has transferred to the $PAUSE subroutine.
PRESS START TO CONTINUE
(MACHINE PAUSE)
CONDITION IGNORED

CTE2, CTG2 (entered at CTB2)
A user has transferred to the PAWSX subroutine.

CANNOT PROCEED

CTB4
A user has transferred to the STOPX subroutine.

ILLEGAL TRAP OR UNIT REQ-ST AT XXXXX
CANNOT PROCEED

CCH2, CUJ2, CUK2
A data channel trap was received from a non-existent channel.
CAA2/CLA1, CUD2, CUJ2, CUK2
A use of ACTIV or NDSEL has specified an illegal unit in its calling sequence.

UNIT XX ILLEGAL NON DATA SELECT AT TRAP TIME
CANNOT PROCEED

CLE4, CLG5, CLH5
A non-data select operation has been attempted during trap time on a channel other than the channel that is trapped.

UNIT XX .......................... USERS MESSAGE ......................
CWA5, CWD5
A user has used the CVRPT subroutine to convert a tape address to symbolic and print his own message along with it.

XXXXXX YY ERR* TRK ZZZZ XXXXX -- SYMBOLIC UNIT
(ONE OR MORE OF THE ABBREVIATIONS LISTED BELOW) YY -- RD/WR
ZZZZ -- TRACK NUMBER
PRGCK DATCK EXCND I-SEQ I-CDE
FRMCK NOREC I-ADR RSPCK DCOMP
P/CK INOPR NTRDY DKCRK CUCRK

CPB4/CPK3/CRD2, CPH5
An unsuccessful recovery attempt has been made on a disk error. The abbreviation(s) for the type of error(s) is (are) printed on the line following the symbolic unit, type of error, and track number.

U. E. ON A SEEK—PRESS START TO RETRY.
OPERATOR ACTION PAUSE
CSJ4, CSH5, CSG5
An Unusual End has been received on a seek issued by the ISSEx or ISSK1 routines.
UNIT XX FILE XXXXX REC. XXXXX NOISE RECORD DISCARDED
CEF4, CEF3
A user's SEL- routine has used the 2, 4 return to indicate that the noise record on a read operation is to be discarded.

UNIT XX FILE XXXXX REC. XXXXX SHORT RECORD WRITTEN
CEF4, CEH4
A user's SEL- routine has used the 2, 4 return to indicate that the short record written is to be ignored.

UNIT XX FILE XXXXX REC. XXXXX PERMANENT READ REDUNDANCY
CHA3, CHC3, CHD3
A permanent read redundancy has been encountered during redundancy recovery, and the message is not suppressed.

UNIT XX FILE XXXXX REC. XXXXX NOISE ON ERASE
CJA3, CJA2
A redundancy has been encountered while attempting to write an erase area.

UNIT XX FILE XXXXX REC. XXXXX 25 ERASES DURING WRITE
CJD4, CJE4
25 erases have been made while attempting to recover from a write redundancy.

UNIT XX EOT ON ERASE
CANNOT PROCEED
CJG3, CJG2, CJH2
An EOT has been sensed on some write redundancy recovery attempt other than the first.

I/O CK*, 7607
PRESS START TO CONTINUE
CCA4, CCA5, CCB5
An I/O check has been detected on a 7607 channel following a data channel trap.

INTF CK*, 7909
SEQ CK*, 7909
I/O CK*, 7909
PRESS START TO CONTINUE
CDC2, CDB3, CDC3, CDG3
The indicated error conditions were detected on a 7909 channel following a data channel trap.
Appendix B — Assembly Parameters

**Machine and System Definitions**

<table>
<thead>
<tr>
<th>Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>Machine type (709 or 7090)</td>
</tr>
<tr>
<td>IBSOFG</td>
<td>Basic Monitor origin for nucleus (minimum is last machine trap/inter-rupt cell +1)</td>
</tr>
<tr>
<td>IOXORG</td>
<td>IOEX origin (minimum is last UCB cell +1)</td>
</tr>
<tr>
<td>SYSORG</td>
<td>System origin (minimum is last installation trap routine cell +1)</td>
</tr>
<tr>
<td>SYSEND</td>
<td>System upper core limit (move down to allow room for installation accounting routines, etc.)</td>
</tr>
<tr>
<td>HIGHLO</td>
<td>Density to be assumed on SAS card if not specified (1 indicates high, 0 indicates low)</td>
</tr>
<tr>
<td>EJECT</td>
<td>Printer board sense exit for eject (1 through 10)</td>
</tr>
<tr>
<td>DBLSP</td>
<td>Printer board sense exit for double space (1 through 10)</td>
</tr>
<tr>
<td>RDUNRT</td>
<td>Number of recovery tries for each read redundancy</td>
</tr>
<tr>
<td>ETM0DE</td>
<td>Transfer trap mode (1 indicates trap mode to be saved by IOEX, 0 indicates trap mode remains in effect on IOEX)</td>
</tr>
</tbody>
</table>

**Intermediate Definitions (Calculated from Prior Definitions)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTA1-HTH1</td>
<td>Number of Hypertapes on interface 1</td>
</tr>
<tr>
<td>HTAA1-HTHA1</td>
<td>Hypertape attachment indicators; similar to CHAT</td>
</tr>
<tr>
<td>HTA2-HTH2</td>
<td>Number of Hypertapes on interface 2</td>
</tr>
<tr>
<td>HTAA2-HTHA2</td>
<td>Hypertape attachment indicators; similar to CHAT</td>
</tr>
<tr>
<td>DFA1-DFH1</td>
<td>Number of modules on interface 1 (0 through 10)</td>
</tr>
<tr>
<td>DFAA1-DFHA1</td>
<td>Disk module attachment indicators; similar to CHAT</td>
</tr>
<tr>
<td>IFA1-IFH1</td>
<td>Model of 7631: 1-Model III or IV, 0-all others</td>
</tr>
<tr>
<td>DFA2-DFH2</td>
<td>Number of modules on interface 2 (0 through 10)</td>
</tr>
<tr>
<td>DFAA2-DFHA2</td>
<td>Disk module attachment indicators; similar to CHAT</td>
</tr>
<tr>
<td>IFA5-IFH2</td>
<td>Model of 7631: 1-Model III or IV, 0-all others</td>
</tr>
</tbody>
</table>

**Unit Definitions**

<table>
<thead>
<tr>
<th>Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHA1-CHH1</td>
<td>Number of tapes, (0 through 10)</td>
</tr>
<tr>
<td>CHAAT-CHHAT</td>
<td>Tapes attached ........ 1 bit per unit signifying whether unit is attached to machine, (i.e. legally dialable). Bits go left to right (CR, RV, FR, tapes 1 to 10) and equal 1 if not attached</td>
</tr>
<tr>
<td>CHAMD-CHHMD</td>
<td>Model types of units on channel: 1-Model IV, 0-Model II; same arrangement as CHAT</td>
</tr>
<tr>
<td>PRNTA-PRNTH</td>
<td>Printer; 1-printer exists</td>
</tr>
<tr>
<td>PNCHA-PNCHH</td>
<td>Punch; 1-punch exists</td>
</tr>
<tr>
<td>CORDA-CORDH</td>
<td>Card reader; 1-card reader exists</td>
</tr>
<tr>
<td>DUCA-DUCA</td>
<td>Number of disk arms on each channel</td>
</tr>
<tr>
<td>TDUAC</td>
<td>Total number of disk arms on all channels</td>
</tr>
<tr>
<td>DUEX</td>
<td>Equals 1 if disk exists on any channel</td>
</tr>
<tr>
<td>CHA2-CHH2</td>
<td>Number of 7909 devices on each channel</td>
</tr>
<tr>
<td>CHA3-CHH3</td>
<td>Channel type: 1 indicates 7909, 0 indicates any other</td>
</tr>
<tr>
<td>UG1-UG4</td>
<td>1 bit in a word for each card machine that exists (Figure 4)</td>
</tr>
<tr>
<td>CHA0-CHH0</td>
<td>Number of I/O devices on each channel</td>
</tr>
<tr>
<td>NOUCW</td>
<td>Total number of I/O devices on all channels</td>
</tr>
<tr>
<td>CHA4-CHH4</td>
<td>Existing channel indicators: 1-channel exists, 0-channel does not exist</td>
</tr>
<tr>
<td>CHH5</td>
<td>Total number of channels that exist</td>
</tr>
</tbody>
</table>

Figure 4 shows the bit configuration for the key cells required by cold-start to generate the Unit Control Blocks. These words are constructed during compilation from the unit and intermediate definitions.
### AAMT

| 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| CR | PU | PR | tape | C8 | P8 | P9 | P10 |

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH - MD</td>
<td>1 - Model IV</td>
</tr>
</tbody>
</table>

### ROCTST

<table>
<thead>
<tr>
<th>UG1</th>
<th>UG2</th>
<th>UG3</th>
<th>UG4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRDA</td>
<td>P2A</td>
<td>CRDA</td>
<td>P2A</td>
</tr>
<tr>
<td>CRDA</td>
<td>P2B</td>
<td>CRDA</td>
<td>P2B</td>
</tr>
<tr>
<td>CRDA</td>
<td>P2C</td>
<td>CRDA</td>
<td>P2C</td>
</tr>
</tbody>
</table>

### NTONCH

<table>
<thead>
<tr>
<th>CHA1</th>
<th>CHB1</th>
<th>CHC1</th>
<th>CHD1</th>
<th>CHE1</th>
<th>CHF1</th>
<th>CHG1</th>
<th>CHH1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTA1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTA2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFA1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDA1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFA2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDA2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### DFONC1

<table>
<thead>
<tr>
<th>AADF1</th>
<th>AADF2</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF - 1</td>
<td>DF - A1</td>
</tr>
<tr>
<td>IF - 2</td>
<td>DF - A1</td>
</tr>
</tbody>
</table>

### AAHT1

<table>
<thead>
<tr>
<th>AAHT1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HT - A1</td>
</tr>
</tbody>
</table>

### AAHT2

<table>
<thead>
<tr>
<th>AAHT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>HT - A2</td>
</tr>
</tbody>
</table>

Figure 4.
Appendix C — Channel Information and Operation Tables

These tables are comprised of one entry for each channel that exists. The following tables have indirect references within the IOEX communication region and may be referenced in this manner by the user: CHXAC, URRX, RCTX, RCHX, TOOX, TRCX, ETTX, and TEFX. When using the indirect references, IN must contain the channel index, which is the complement of the channel number minus one.

**CHXAC**  
Channel activity cells  
Cleared—no activity on a channel or non-standard use of channel  
L(UCB)—location of UCB for the unit that is active on the channel

**CHXSP**  
Channel priority cells  
Cleared—no priority assigned  
L(UCB)—priority is assigned to the unit indicated in that UCB

**CHXCW**  
Channel control cells  
PZE ADDR, 2, length of all UCB’s for channel. ADDR points to word 2 of these UCB’s

**UCBA-UCBH**  
Locators for UCB’s by channel  
Prefix—number of card machines on channel  
Decrement—total number of I/O devices on channel  
Address—address of first UCB on this channel

**UCAA-UCAH**  
Locators for available unit chains by channel  
Address—address of UCB for first unit in the availability chain.

**RCTX**  
Redundancy check control cells  
Zero—redundancy checking disabled  
Not zero—redundancy checking allowed (always zero for a 7909 channel)

**URRX**  
Redundancy count cells  
PZE N1, N2  
N1—number of recovery entries to SEL+  
N2=0—no permanent redundancies (READ) or erase areas (WRITE)  
N2=1—permanent redundancy (READ) or one or more erase areas (WRITE)  
(N1=0 if N2=0, and N2 always 0 for disk I/O)

The following tables are instruction tables for use by the program and the user so that the recovery routines will be able to repeat the operation when required. The recovery routines need only to execute these instructions, which have been set up by the user’s SEL+ routine.

**ETTX**  
End-of-tape test table

**RCHX**  
Reset and load channel table

**SCHX**  
Store channel table

**SPCX**  
Store diagnostic channel table

**STCX**  
Start channel table

**TCOX**  
Test channel in operation table

**TEFX**  
Test end-of-file table

**TRCX**  
Test channel redundancy table
Appendix D — Key Switches and Storage Locations

ACOMM Store Channel Results
Subroutine PROUT places the SCHA data into this location when forcing a trap on the printer channel. XTRAP will move this data into COMM. If either FBSW2 or FBSW3 is not zero, the SCHX in XTRAP is bypassed.

CHPSW Checkpoint Switch
Set by user when taking a checkpoint. Indicates to BG1 not to attempt to initiate another operation on the channel that just caused the trap unless it is a 7909 channel.

COLD Cold Start Executed Switch
First location of cold-start routine which is cleared to indicate that the routine has been executed. It is cleared in both the cold-start and IBRS routines. It indicates to INXXC that SYSLTH must be initialized when cold-start has been executed.

COMM, COMMMD Store Channel Results
XTRAP stores the channel results in this location on each channel trap unless FBSW2 or FBSW3 is not zero. In this case, the data from ACOMM is used. This data is used by NRTST to determine the length of the record in question.

DCSWSW Disk Channel Switch
XTRAP sets this location to contain the location of the UCB for the operation causing the trap if the unit is on a 7909 channel. When this location is zero, a non-7909 channel is indicated. This location is cleared on exit from XTRAP in the OUT routine.

DMPSW Dump Switch
This location is set by the dump caller to indicate that two records are to be read in. It is cleared when the dump program is scattered loaded into core.

DSYSNM Disk System Name Table
This table is comprised of two word entries for each system on disk. The format is as follows:
BCI 1, SYSSAM
EZE , , DOBG

DUELL Last Disk Error Indicators
This location is used in the disk recovery routine to save the error conditions obtained on one entry to the routine, so that they may be compared with the errors found on the next entry.

ENBSW, GENBSW Enable Switch
When zero, this location indicates to IOEX that it may enable traps. IBSUP will enable traps before going to SYSTRA after loading in a new system. When this location is not zero, IOEX may not enable traps. This location is set by the user.

FORCE SWITCHES A detailed explanation of the three force switches and their interaction is given in the explanation of the PROUT subroutine.

FRSW1 Force Switch 1
This location is set to non-zero in the PROUT routine whenever it is necessary to force a channel trap on channel A. It causes the SAVE routine to transfer immediately to PR3 in the PROUT routine without processing the trap which has been forced, or even setting TRPSW at this time. This location is cleared on exit from the trap supervisor if it is found non-zero.

FRSW2 Force Switch 2
This location is set to non-zero on an exit from the PROUT routine when both FRSW1 and TRPSW are not zero and FBSW3 is zero. This condition indicates to PROUT and XTRAP that there is one trap waiting to be processed and one trap being processed. This location is
cleared on exit from the trap supervisor if _FBSW1_ is zero.

**FBSW3**

**Force Switch 3**

This switch is set in the exit routine in _FR3UT_ if _FBSW1_ is not zero and _TRPSW_ is zero. This condition indicates to _FR3UT_ and _XT3AP_ that there is a channel A trap waiting to be processed. This location is cleared on exit from the trap supervisor when _FBSW1_ is not zero and _FR3SW2_ is zero. Control then returns to the exit routine in _FR3UT_.

**GOOF**

**Illegal Trap Switch**

This is an instruction in _XT3AP_ whose address will be set to non-zero when any trap occurs on a non-existent channel. When it is tested and found non-zero, it will cause a system halt after an on-line print-out indicating the error.

**IBASX**

**Prior Assignment of Newly Assigned System Unit**

This location contains the _UCB_ address for the prior assignment to a logical function. This is used in the _IBREL_ and _IBASC_ routines.

**IBOPT, IBOPX**

**Legal Operation Code Tables**

These two tables contain the legal operation codes which may be used on basic monitor control cards, and the addresses of the routines used to process each particular control card.

**IBAV**

**Original Sense Switch 1 Setting**

This is a location within the disk limits table where the original setting of sense switch 1 is saved. It is also set according to the requirements of a _SCARDS_ or _TAPE_ control card.

- **Zero — Down** (CARDS)
- **Non-Zero — Up** (TAPE)

**IBSINP**

**Input Switch**

This switch indicates cards or tape mode of input. Zero indicates cards, non-zero indicates tape. A _RESTORE_ card does not change this setting when restoring the nucleus.

**IBWAT**

**Control Card Wait Loop**

This location may be set to one of four possible values, as shown:

- **TRA** * Set by _IBSYS_ while waiting for the data channel trap.
- **TRA IBTPIN**
- **TRA IBODIN** Set by the _SEL_ routine in _IBSYS_ to give control to the proper control card interpretation routine.
- **TRA IBRET** Set by the _SEL_ routine in _IBSYS_ when an _EOF_ is sensed in the card reader.

**Interlock Switch**

This switch prevents an interrupt unless the position of sense switch 1 has been alternated. For example, if the operator has used sense switch 1 to cause control cards to be taken from the card reader on initial loading and has not replaced this switch in the up position, an interrupt will not occur. If the operator has placed the switch down for an interrupt and receives the interrupt, but does not return the switch to up, another interrupt will not occur until the switch is alternated.

If the switch is not zero, the interlock is on. It is set to non-zero during cold-start if sense switch 1 is used to indicate card input. It may also be set to non-zero in _IBXCC_ when returning from an interrupt. It will be cleared on the first entry to the _INTUP_ routine when it is found set and sense switch 1 is up.

**Last Unit Attached**

This location contains the address of the _UCB_ for the last unit attached. The address is set by the _IBATC_ routine, and can be cleared by the _IBREL_ routine. It is used by both the _IBASC_ and the _IBREL_ routines.

**List Switch**

This switch indicates that control cards are to be printed when it is on.
Zero — on: Control cards printed on-line.
Non-zero—off: Control cards not printed on-line, (except PAUSE, STOP, SLEET and illegal cards).

MAPSW Map Switch
This switch is used by IBUNI to indicate when any data has been printed under a particular heading.
Zero — no; Non-zero — yes.

ONETRP Single Channel Enable Trap Location
This location is used when it is desired to enable traps on one channel only (PROUT, PUNCH, DKRV routines).

PRINT Image Printed Switch
This switch is used within the BCONIC routine. Zero indicates that the on-line image area is to be cleared before converting any more data. Non-zero indicates that only part of the desired image is converted, and it has not yet been printed. In this case the image cannot be cleared.

PSITSW Pause Switch
This switch determines which pair of messages will be printed on-line with the use of the machine pause routine. Non-zero indicates that the entry to the routine was through PAUSE, and the operator intervention messages are to be printed. Zero indicates that the entry was through PAWS and the error messages are to be printed. This location is cleared on all exits from the pause routine.

PUNSW Punch Switch
This switch indicates that the on-line message is to be punched instead of printed. It is set to non-zero to indicate this if the entry is made to the PUNCH routine. It is set to zero on all exits from either the PUNCH or PRROUT routines.

RCOLDX Alternate Cold-Start Exit
During the initial use of cold-start, this location contains a TRA *+1. If the cold-start routine is being used for a RESTORE operation, this location will contain a TRA IBSYS. This operation bypasses the setting of the input switch and the operations concerning the input unit.

RUPSW Interrupt Switch
This switch is set to non-zero by INTRUP and cleared by INXXC. When set, it indicates to IBXSYS to return to the current system when an EOF occurs in the card reader during an interrupt. When set, it also indicates to INXXC to set the interlock switch, INTSW, to prevent another interrupt until sense switch 1 is alternated.

RWIND Temporary Sense Indicator Storage
XTARP saved the sense indicators in this location if a read redundancy is found to be a noise record.

SYSORG + 100 FZE SYSLBI limits (disk)
SYSORG + 101 FZE Channel index, , Unit address
SYSORG + 102 FZE L.UCB, , TTIP of spare tracks of IBSYS cylinder.

SYSUNI, SYSLBI + 19 System Units and Disk Limits Tables
The system units table contains a one word entry for each logical system unit. When loaded, the format of an entry in this table is FZE CHAN, , UNIT.
The format of this table when in use is FZE L.UCB. In both formats, the prefix is used to indicate the density of the logical unit.
MZE stands for high density, and PZE stands for low density.
The disk limits table contains a one word entry corresponding to each entry in the SYSUNI table. It is in the format of FZE DORC, , DEND.
The locations corresponding to SYSCHR, SYSFRT, and SYSPCH are used for switch storage and labelled ISSEH, INTSW, and IB7AV, respectively.

SYNUM System Units External Name Table
This table contains a one word entry for each logical system unit.

TRAP Current Traps Enabled
This location contains the word used to enable traps at this time.
Trap Switch

When this location contains \texttt{L(UCB)} as set in \texttt{XTRAP}, it indicates that a trap is being processed. When this location is cleared on an exit from \texttt{XTRAP}, it indicates that the trap has been completely processed. The sign of \texttt{TRPSW} is set to 1 on an erase operation on tape.

Indirect Reference to \texttt{TRAP, CTRAPS}

This location in the \texttt{IOEX} communications region provides an indirect reference to \texttt{TRAP, CTRAPS} containing the current traps enabled. This cell may be changed by the user and then reset to its original value if the user desires to indicate to \texttt{IOEX} to use another trap cell for enabling traps. (\texttt{IOCS} makes use of this feature.)

Enable Printer Channel Trap Cell

This location contains a 1 in both the address and decrement and is used to enable traps on the printer channel by \texttt{PROUT} when a channel A trap must be forced. It is also used by \texttt{DKRCV} and shifted to the desired position to enable traps on the desired channel only.
## Appendix E — Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>ACCUMULATOR</td>
</tr>
<tr>
<td>ADD</td>
<td>ADDRESS</td>
</tr>
<tr>
<td>ADDR</td>
<td>ADDRESS</td>
</tr>
<tr>
<td>AF</td>
<td>SEEK REQUEST FLAG (ACTIVITY FLAG)</td>
</tr>
<tr>
<td>AVL</td>
<td>AVAILABILITY, AVAILABLE</td>
</tr>
<tr>
<td>BCD</td>
<td>BINARY CODED DECIMAL</td>
</tr>
<tr>
<td>CE</td>
<td>CUSTOMER ENGINEER (TRACK)</td>
</tr>
<tr>
<td>CHAN</td>
<td>CHANNEL</td>
</tr>
<tr>
<td>CHK</td>
<td>CHECK</td>
</tr>
<tr>
<td>CHN</td>
<td>CHAIN</td>
</tr>
<tr>
<td>C(UCW)</td>
<td>CONTENTS OF UNIT CONTROL WORD</td>
</tr>
<tr>
<td>DEND</td>
<td>DISK END</td>
</tr>
<tr>
<td>DORG</td>
<td>DISK ORIGIN</td>
</tr>
<tr>
<td>EOF</td>
<td>END OF FILE</td>
</tr>
<tr>
<td>EOT</td>
<td>END OF TAPE</td>
</tr>
<tr>
<td>EQ</td>
<td>EQUAL TO</td>
</tr>
<tr>
<td>GT</td>
<td>GREATER THAN</td>
</tr>
<tr>
<td>HI</td>
<td>HIGH</td>
</tr>
<tr>
<td>IF</td>
<td>INTERFACE</td>
</tr>
<tr>
<td>IND</td>
<td>INDICATORS</td>
</tr>
<tr>
<td>INFO</td>
<td>INFORMATION</td>
</tr>
<tr>
<td>INSTALL</td>
<td>INSTALLATION</td>
</tr>
<tr>
<td>INSTR</td>
<td>INSTRUCTION</td>
</tr>
<tr>
<td>IR</td>
<td>INDEX REGISTER</td>
</tr>
<tr>
<td>I/O</td>
<td>INPUT/OUTPUT</td>
</tr>
<tr>
<td>LOC</td>
<td>LOCATION</td>
</tr>
<tr>
<td>LRC</td>
<td>LONGITUDINAL REDUNDANCY CHECK</td>
</tr>
<tr>
<td>LT</td>
<td>LESS THAN</td>
</tr>
<tr>
<td>L(UCB)</td>
<td>LOCATION OF UNIT CONTROL BLOCK</td>
</tr>
<tr>
<td>MISC</td>
<td>MISCELLANEOUS</td>
</tr>
<tr>
<td>NO</td>
<td>NUMBER</td>
</tr>
<tr>
<td>OP</td>
<td>OPERATION</td>
</tr>
<tr>
<td>ORIG</td>
<td>ORIGINAL</td>
</tr>
<tr>
<td>PF</td>
<td>SEEK ISSUED FLAG (PENDING FLAG)</td>
</tr>
<tr>
<td>SEQ</td>
<td>SEQUENCE</td>
</tr>
<tr>
<td>SPR</td>
<td>SENSE PRINTER (SENSE PUNCH)</td>
</tr>
<tr>
<td>TTTT</td>
<td>TRACK ADDRESS</td>
</tr>
<tr>
<td>UCB</td>
<td>UNIT CONTROL BLOCK</td>
</tr>
<tr>
<td>UCW</td>
<td>UNIT CONTROL WORD</td>
</tr>
<tr>
<td>UE</td>
<td>UNUSUAL END</td>
</tr>
<tr>
<td>VIA</td>
<td>BY WAY OF</td>
</tr>
<tr>
<td>WEF</td>
<td>WRITE END OF FILE</td>
</tr>
<tr>
<td>W/O</td>
<td>WITHOUT</td>
</tr>
<tr>
<td>.XEC</td>
<td>EXECUTE</td>
</tr>
</tbody>
</table>
Appendix F—Glossary

Activate File: To cause an I/O operation to be started on a channel, on the unit specified in the file designation.

Active: Status of a channel or a file when an I/O operation is being performed on that channel or file.

Activity Flag, AF (Seek Request Flag): The sign position of word 3 of a disk unit control block which is used to indicate when a seek is requested by a user, or a seek was issued and an attention signal is awaited.

Adapter Check: A condition in the 7909 channel which can cause an interrupt.

Attached, Unattached: The physical condition of an I/O device, referring to whether or not the device is actually attached to the system. Bit position 1 of word 1 of the ucb is used to indicate this status when the sign position is set to 1. (See ucb layout in IBMRS Reference Manual, Form C28-6248.)

Attention Interrupts: This is a signal which indicates a change in status of an attached I/O device. An attention signal is generated at the completion of a seek operation by an access mechanism.

Availability Chain: This is a logical list of attached units which have not been assigned to a specific system unit function. The address of the first unit in this chain for each channel is found in the table uc.a through uc.an in the nucleus. The address of subsequent ucb’s is found in the address part of word 1 of each entry in the chain. (Refer to Chart BY.)

Available, Unavailable: The logical status of an I/O device which is attached. The sign position of word 1 of the ucb is used to indicate this status. (See ucb layout, IBMRS Reference Manual, Form C28-6248.)

Calling Sequence: An instruction that gives control to a subroutine with one or more instructions or constants which are provided to supply the parameters for the subroutine operation.

CE Track: A track on disk reserved for Customer Engineering use, which cannot be used by the programmer, but to which the access mechanism will move whenever an illegal address is specified.

Channel Index: A numeric value used to index through a table of instructions which is as long as the number of channels attached to a system. The value is equal to the complement of the desired channel number minus one. This value is usually placed in ir1, and the reference to the start of one of these tables is tagged with ir1.

Checkpoint, Checkpoint Record: A record written at periodic intervals by a program containing the complete contents of core storage and all the machine registers and machine status conditions. This record may be later used to provide a restart without necessitating a complete rerun of the program.

Clear: When speaking of a switch or a location in storage, the term “clear” refers to an all zero condition.

Dead Stop: A type of halt provided in the program from which there is no normal recovery. Depressing start causes no change because the halt consists of a HTR.

Diagnostic Channel Data: The execution of a “store diagnostic channel” instruction causes the contents of the channel check register to be placed in positions 6 through 11 of the location specified. The status of these bits may then be tested without using the TCm instruction normally used to check these conditions.

Disk Limit: The starting and ending addresses of a system on disk.

Dormant: Not active, which means no I/O operation is in progress on a channel or file at this time.

Entry Point: The first instruction in a subroutine that is executed. This is not necessarily the first actual instruction found in the routine, nor need it be the same for each use of the routine.

Erase Area: A blank area on magnetic tape produced when either the word count in a desired I/O command is zero, or when no I/O instruction follows a WNS. This blank area is then approximately 3-3/4 inches long.

Initialized, Initialization: The preliminary setup of an area in storage, or of key locations and status indications prior to the actual beginning of a routine or subroutine.

Initiate File: To start an I/O operation on a channel that is already active. This is performed at the completion of another I/O operation without allowing the channel to become dormant. I/O Check: A condition in a 7909 channel which can cause an interrupt. Tape—A condition in a 7047 channel which can cause a data channel trap if this type of trapping is enabled.

Inter-system Reserve Unit: A logical I/O device that is given a reserve designation by one system to indicate that the unit may not be returned to the availability chain. It may be referenced later by another system when called for using the same reserve designation.

Machine Registers: Those registers in the machine which are saved prior to the execution of a routine, and restored at the end of that routine. These include the accumulator, MQ register, index registers, and the sense indicators.

Machine Status Conditions: Those status conditions which are saved prior to the execution of a routine, and restored at the end of that routine. These may include some or all of the following: Transfer trapping mode, overflow conditions, divide check status, and I/O check status.

Noise Record: A record either read or written which contains less than three words. This would be equivalent to 18 BCT characters.

Non-data Select Operation: An I/O operation that does not cause movement of data. These operations include: Set density high or low, Rewind, Unload, Backspace Record, Backspace File, and Write End-of-File.

Non-trap Time: That time when the program is not processing a data channel trap.

Parameter: A specification provided as input to the compiler when assembling the program which would affect the manner in which the compilation was performed. It may also be the specifications provided as input to a subroutine in the subroutine calling sequence.

Pending Flag, FFF (Seek Issued Flag): Bit position 1 of word 3 of a disk ucb which is used to indicate when a seek has been issued and an attention signal is awaited.

Permanent Read Redundancy: A redundancy has been encountered on tape and the number of recovery attempts specified in ROUNT or ROUNT has been attempted unsuccessfully.

Point to, Set to Point to: When referring to a location in storage or an index register, the term “point to” indicates that the register or location is loaded with the complement of the desired value. This value allows the use of an index register to easily step through a block of information starting at the desired location.

Posting, Post Completion: That operation performed by the user’s Select Minus routine to check the operation that just caused a trap, and to set up the indicators for TOES to determine what recovery actions should be taken if there was an error detected.
**Reset:** When referring to a switch or storage location, the term "reset" indicates a cleared or zero condition.

**Restart:** The process of starting an operation over again, possibly from some point other than the start of the program.

**Save:** To pick up and store information which will be required later.

**Save Record:** That record written on SYSUT by the dump caller, which contains the contents of the machine registers and machine status conditions, and that portion of core storage which will later be occupied by the dump program. This information is needed for the dump program in compiling its print-out.

**Scatter Load Routine:** That portion of the system loader that enables one physical record to be broken up into smaller segments which can be loaded into different areas in core storage which are not necessarily adjoining.

**Select Routine, sel+, sel-:** The routine provided to perform an I/O operation under control of IOEX. It is comprised of two sections: sel+ and sel-.- sel+ starts an I/O operation and returns control to IOEX, while sel- posts completion of an I/O operation and then returns control to IOEX.

**Set:** When referring to a switch or storage location without specifying an actual value, the term 'set' indicates a non-zero condition.

**Sequence Check:** A condition in the 7909 channel which can cause an interrupt.

**Stacked Jobs:** A series of jobs placed peripherally on the same input tape, which may call for different systems under the same monitor system, and which will be processed in sequence unless directed otherwise by the operator.

**Tape Cleaner:** An operation performed by the read redundancy recovery routine in an attempt to recover when a bad record is detected. It consists of two extra BSR operations followed by two RUS operations which cause the tape to move the suspected bad area back into the vacuum columns. This is done in an attempt to dislodge any dirt particles which could possibly have caused the redundancy.

**Trap Supervisor, IOEX:** That portion of the Basic Monitor which remains in storage and receives control from each data channel trap or interrupt.

**Trap Time:** That time when the program is processing a data channel trap.

**Unit Control Blocks, UCB's:** A series of four-word blocks in IOEX that contain the information pertaining to a particular I/O device. (See UCB layout in IBSYS Reference Manual, Form C2S-6248.)

**Unit Control Word:** One of the four words in the Unit Control Block.

**Unusual End:** A condition in the 7909 channel which can cause an interrupt.
**Appendix G — Autochart Cross Reference Tables**

This section contains three cross reference tables which may be used with the AUTOCHART diagrams within this manual. Their use will enable a reader to locate any label appearing on any chart, all references to a particular entry connector, and all blocks where a subroutine is used.

**Table A — Labels**

This table lists all labels that appear on all charts. To the right of each label is the chart and block location where the label appears.

<table>
<thead>
<tr>
<th>LABEL</th>
<th>BLOCK</th>
<th>LABEL</th>
<th>BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>IACTIV</td>
<td>ABA2</td>
<td>A3</td>
<td>CCG2</td>
</tr>
<tr>
<td>IACTVX</td>
<td>ABB2</td>
<td>A4</td>
<td>CGA3</td>
</tr>
<tr>
<td>IBODSR</td>
<td>AC2</td>
<td>BCD5</td>
<td>ACJ3</td>
</tr>
<tr>
<td>IBODSX</td>
<td>AC1J2</td>
<td>BCDVDC</td>
<td>ACG3</td>
</tr>
<tr>
<td>ICVPRT</td>
<td>AC2</td>
<td>BG1</td>
<td>ABJ3</td>
</tr>
<tr>
<td>IDECVA</td>
<td>AC2G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDECV0</td>
<td>AC2F2</td>
<td>BG1+1</td>
<td>CFA3</td>
</tr>
<tr>
<td>IFDAMT</td>
<td>AB2E</td>
<td>BG2</td>
<td>CFH2</td>
</tr>
<tr>
<td>IIBDOZ</td>
<td>ACK2</td>
<td>BG3</td>
<td>CFB4</td>
</tr>
<tr>
<td>INOE1L</td>
<td>ABC2</td>
<td>BG4</td>
<td>CFD3</td>
</tr>
<tr>
<td>INOE1LX</td>
<td>AB2D</td>
<td>BG5</td>
<td>CFG3</td>
</tr>
<tr>
<td>IPAMO5X</td>
<td>AC2B</td>
<td>BG6</td>
<td>CFE4</td>
</tr>
<tr>
<td>IPRODT</td>
<td>ACC2</td>
<td>CALIB1</td>
<td>BPΑ2</td>
</tr>
<tr>
<td>ISTIPX</td>
<td>ACA2</td>
<td>CALIB2</td>
<td>BPD2</td>
</tr>
<tr>
<td>IVSMYN</td>
<td>AC2E</td>
<td>CALIB3</td>
<td>BRA2</td>
</tr>
<tr>
<td>ACTV</td>
<td>CA2A</td>
<td>CALPOW</td>
<td>BSA1</td>
</tr>
<tr>
<td>ACTXD</td>
<td>CB2J</td>
<td>CAL1</td>
<td>BPB2</td>
</tr>
<tr>
<td>ACTX2</td>
<td>CBG4</td>
<td>CAL10Y</td>
<td>BRE1</td>
</tr>
<tr>
<td>ACTX1</td>
<td>CBJ4</td>
<td>CAL11Y</td>
<td>BRE2</td>
</tr>
<tr>
<td>ACT1</td>
<td>CAD2</td>
<td>CAL12Y</td>
<td>BRF5</td>
</tr>
<tr>
<td>ACT3</td>
<td>CAE5</td>
<td>CAL2</td>
<td>BPC2</td>
</tr>
<tr>
<td>ACT4</td>
<td>CAH3</td>
<td>CAL2Y</td>
<td>BRO2</td>
</tr>
<tr>
<td>ACTS</td>
<td>CB35</td>
<td>CAL3</td>
<td>BPG2</td>
</tr>
<tr>
<td>ACTA5</td>
<td>CBE4</td>
<td>CAL3Y</td>
<td>BRC3</td>
</tr>
<tr>
<td>ACT6</td>
<td>CBA2</td>
<td>CALSY</td>
<td>BSA4</td>
</tr>
<tr>
<td>ACT8</td>
<td>CBF2</td>
<td>CAL6Y</td>
<td>BSA5</td>
</tr>
<tr>
<td>A1</td>
<td>CG2A</td>
<td>CAL7Y</td>
<td>BSD5</td>
</tr>
<tr>
<td>A2</td>
<td>CGB2</td>
<td>CAL8Y</td>
<td>BSA3</td>
</tr>
<tr>
<td>CAL9Y</td>
<td>BRC1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHPN+1</td>
<td>BAA3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOD</td>
<td>BAA2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVPRT</td>
<td>CA5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATCK</td>
<td>CPH5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATNS</td>
<td>CV1A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATN1</td>
<td>CVG1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATN2</td>
<td>CVF2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATN3</td>
<td>CVJ2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEND1</td>
<td>CDA3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEND2</td>
<td>CDA4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENDA</td>
<td>CDB3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEN1B</td>
<td>CDD3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFMN</td>
<td>CGF3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMBR2</td>
<td>C E4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMRV</td>
<td>CP2A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMPER</td>
<td>BT1F1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMPTO</td>
<td>BD5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRCDL</td>
<td>BAE2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE21</td>
<td>CPC3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE22</td>
<td>CPC5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE23</td>
<td>CPF5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE24</td>
<td>CPG5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE25</td>
<td>CPN5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE26</td>
<td>CPN5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE27</td>
<td>CRE3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE28</td>
<td>CRG3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE29</td>
<td>CRA5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE34</td>
<td>CRC5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUE10</td>
<td>CRG5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUMP</td>
<td>BPA5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDTS</td>
<td>BNA2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>END1</td>
<td>CCA5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>END12</td>
<td>CEG4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>END13</td>
<td>CEJ3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>END14</td>
<td>CEJ4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>END15</td>
<td>CEG2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>END16</td>
<td>CEGH2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

110
<table>
<thead>
<tr>
<th>LABEL</th>
<th>BLOCK</th>
<th>LABEL</th>
<th>BLOCK</th>
<th>LABEL</th>
<th>BLOCK</th>
<th>LABEL</th>
<th>BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBNJ4</td>
<td>BUJ3</td>
<td>IBGDE</td>
<td>BAJ2</td>
<td>IBOUT</td>
<td>ARK3</td>
<td>RSYUR</td>
<td>BYA1</td>
</tr>
<tr>
<td>IBN8IN</td>
<td>BGD2</td>
<td>IBINIT1</td>
<td>BWE3</td>
<td>PAUSE</td>
<td>CTM2</td>
<td>RSY2</td>
<td>BYF2</td>
</tr>
<tr>
<td>IBOF</td>
<td>BCF5</td>
<td>IBINIT2</td>
<td>BWF3</td>
<td>PAWS</td>
<td>CBI2</td>
<td>RSY3</td>
<td>BYE2</td>
</tr>
<tr>
<td>IBON</td>
<td>BCA3</td>
<td>IBINIT3</td>
<td>BMA3</td>
<td>PROUTO</td>
<td>CMT3</td>
<td>RSY4</td>
<td>BIA2</td>
</tr>
<tr>
<td>BOINC</td>
<td>BCG5</td>
<td>IBINIT5</td>
<td>BWC3</td>
<td>PRP5S</td>
<td>CT3</td>
<td>RUBC3</td>
<td>BLA2</td>
</tr>
<tr>
<td>IBPAC</td>
<td>BJF1</td>
<td>IBINIT6</td>
<td>BWC6</td>
<td>PRTAX</td>
<td>CMT7</td>
<td>SAVE</td>
<td>AG3</td>
</tr>
<tr>
<td>IBERL</td>
<td>BMA2</td>
<td>IBINIT7</td>
<td>BWD4</td>
<td>PRTX2</td>
<td>CMJ5</td>
<td>SAVE</td>
<td>CCA2</td>
</tr>
<tr>
<td>IBRES</td>
<td>BDA2</td>
<td>IBINITUS</td>
<td>BWA2</td>
<td>PRT1</td>
<td>CWC2</td>
<td>SAVE+1</td>
<td>CGD2</td>
</tr>
<tr>
<td>IBRET</td>
<td>BTA2</td>
<td>IBINTRUP</td>
<td>BPA3</td>
<td>PRT2</td>
<td>CWF3</td>
<td>CK1</td>
<td></td>
</tr>
<tr>
<td>IBREW</td>
<td>BHA2</td>
<td>IBSEK</td>
<td>CSB2</td>
<td>PRT3</td>
<td>CWA2</td>
<td>SPRT</td>
<td>CWA4</td>
</tr>
<tr>
<td>IBRN1</td>
<td>BHC4</td>
<td>IBSS5</td>
<td>CSA4</td>
<td>PUNCHO</td>
<td>CVA1</td>
<td>STOP</td>
<td>CT4</td>
</tr>
<tr>
<td>IBRS1</td>
<td>BDC4</td>
<td>IBSSK1</td>
<td>CSA2</td>
<td>RCLD</td>
<td>CVA2</td>
<td>STOPD</td>
<td>CD4</td>
</tr>
<tr>
<td>IBRUN</td>
<td>BHA4</td>
<td>IBSSX1</td>
<td>SCC4</td>
<td>RCOLD</td>
<td>BAH2</td>
<td>SYTRA</td>
<td>RP4</td>
</tr>
<tr>
<td>IBSPM</td>
<td>BCE6</td>
<td>IBSSX2</td>
<td>CSX4</td>
<td>RCOLDX</td>
<td>BAG2</td>
<td>SYTR</td>
<td>HPA</td>
</tr>
<tr>
<td>IBSTP</td>
<td>BJ1A</td>
<td>IBSS1</td>
<td>CSF2</td>
<td>RDUN</td>
<td>BB2</td>
<td>SYUCU</td>
<td>BHA1</td>
</tr>
<tr>
<td>IBSWC</td>
<td>BKA2</td>
<td>IBSS3</td>
<td>CSE3</td>
<td>RDUN1</td>
<td>CK5</td>
<td>SYUC2</td>
<td>BXC1</td>
</tr>
<tr>
<td>IBSW2</td>
<td>BKC4</td>
<td>IBSS6</td>
<td>CSE4</td>
<td>RED10</td>
<td>CKAS</td>
<td>SYUC3</td>
<td>Bjas</td>
</tr>
<tr>
<td>IBSW3</td>
<td>BKF2</td>
<td>IBSS7</td>
<td>CSF4</td>
<td>RED12</td>
<td>CH2</td>
<td>SYUC5</td>
<td>BXC5</td>
</tr>
<tr>
<td>IBSYM</td>
<td>BCT2</td>
<td>IBSS8</td>
<td>CSJ5</td>
<td>RED14</td>
<td>CH4</td>
<td>SYUC7</td>
<td>BXC5</td>
</tr>
<tr>
<td>IBTCC</td>
<td>BIA1</td>
<td>IUEFI</td>
<td>BAG4</td>
<td>RED15</td>
<td>CHB3</td>
<td>TEST</td>
<td>CUA2</td>
</tr>
<tr>
<td>IBTPIN</td>
<td>BEA2</td>
<td>IBEND</td>
<td>CCA4</td>
<td>RED16</td>
<td>CHC2</td>
<td>TP0S1</td>
<td>BTA4</td>
</tr>
<tr>
<td>IBUNL</td>
<td>BKA2</td>
<td>IBERR</td>
<td>CLF5</td>
<td>RED18</td>
<td>CHF2</td>
<td>TP0P5</td>
<td>BTA5</td>
</tr>
<tr>
<td>IBUNT</td>
<td>BIA1</td>
<td>IBSEL</td>
<td>CLA1</td>
<td>RED20</td>
<td>CJA2</td>
<td>BTA6</td>
<td>CUG3</td>
</tr>
<tr>
<td>IBVAL</td>
<td>BV3A</td>
<td>IBX5</td>
<td>CLJ3</td>
<td>RED22</td>
<td>CJA3</td>
<td>TSTX1</td>
<td>CUE2</td>
</tr>
<tr>
<td>IBV2</td>
<td>BVH3</td>
<td>IBX5X4</td>
<td>CMJ4</td>
<td>RED5</td>
<td>CJE6</td>
<td>TST1</td>
<td>CUC2</td>
</tr>
<tr>
<td>IBV3</td>
<td>BVF5</td>
<td>IBX51+1</td>
<td>CAM2</td>
<td>RED6</td>
<td>CJK4</td>
<td>TST2</td>
<td>CUC5</td>
</tr>
<tr>
<td>IBV7</td>
<td>BVD5</td>
<td>IBX53</td>
<td>CMK2</td>
<td>RED7</td>
<td>CKB2</td>
<td>UPD1</td>
<td>CMG5</td>
</tr>
<tr>
<td>IBV8</td>
<td>BJ2J</td>
<td>IBX54</td>
<td>CLB3</td>
<td>RED9</td>
<td>CJK4</td>
<td>UPD3</td>
<td>CMG4</td>
</tr>
<tr>
<td>IBWAT</td>
<td>BGD2</td>
<td>IBX55</td>
<td>CMA3</td>
<td>RGTP0</td>
<td>BNA3</td>
<td>UPD6</td>
<td>CMG4</td>
</tr>
<tr>
<td>IBWEF</td>
<td>BMD2</td>
<td>IBX56</td>
<td>CLF1</td>
<td>RGT1</td>
<td>BMD3</td>
<td>UPD7</td>
<td>CMG4</td>
</tr>
<tr>
<td>IBXCC</td>
<td>BG82</td>
<td>MBT1X1</td>
<td>CLJ4</td>
<td>RGT2</td>
<td>BNE4</td>
<td>WEF4+1</td>
<td>CNE3</td>
</tr>
<tr>
<td>IBX1</td>
<td>BG84</td>
<td>MBT2X1</td>
<td>CMA2</td>
<td>RGT4</td>
<td>BNG3</td>
<td>WEF7</td>
<td>CNE4</td>
</tr>
<tr>
<td>IFIND</td>
<td>BVA2</td>
<td>MBT2W3</td>
<td>CAM2</td>
<td>RGT5</td>
<td>BAK3</td>
<td>WEF8</td>
<td>CNF5</td>
</tr>
<tr>
<td>IFND3</td>
<td>BV2D</td>
<td>MBT3D</td>
<td>CMA3</td>
<td>RGT7</td>
<td>BNP5</td>
<td>XTRAP</td>
<td>ABH3</td>
</tr>
<tr>
<td>IFND2</td>
<td>BVCI</td>
<td>MBT3E</td>
<td>CMA2</td>
<td>RGT8</td>
<td>BNG4</td>
<td>XTRAP</td>
<td>CCE2</td>
</tr>
<tr>
<td>IFNDS</td>
<td>BVG1</td>
<td>MBT3T</td>
<td>CMK4</td>
<td>RNXPD</td>
<td>BTJ4</td>
<td>X2</td>
<td>CGD5</td>
</tr>
</tbody>
</table>
**Table B — Entry Connector References**

This table lists all the off-page entry connectors that are referenced by any off-page exit connector on another chart. The entry connector is listed on the left, and all references to this connector appear to the right (and on subsequent lines if necessary).

<table>
<thead>
<tr>
<th>ENTRY</th>
<th>FROM</th>
<th>ENTRY</th>
<th>FROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAE2</td>
<td>BDH4</td>
<td>BDA2</td>
<td>BTM1</td>
</tr>
<tr>
<td>BBA2</td>
<td>BAJ5</td>
<td>BEA2</td>
<td>BJF2</td>
</tr>
<tr>
<td>BCA1</td>
<td>BBD2</td>
<td>BGB2</td>
<td>BNB1</td>
</tr>
<tr>
<td></td>
<td>BNF2</td>
<td>BJA2</td>
<td>BUB3</td>
</tr>
<tr>
<td></td>
<td>BFX4</td>
<td>BJ4A</td>
<td>BJ4J</td>
</tr>
<tr>
<td></td>
<td>BHH4</td>
<td>BJ4A</td>
<td>BFA2</td>
</tr>
<tr>
<td></td>
<td>BJK4</td>
<td>BJ4B</td>
<td>BFA4</td>
</tr>
<tr>
<td></td>
<td>BJF2</td>
<td>BFC4</td>
<td>BFO2</td>
</tr>
<tr>
<td></td>
<td>BJG2</td>
<td>BJ4C</td>
<td>BHC4</td>
</tr>
<tr>
<td></td>
<td>BJH2</td>
<td>BJC4</td>
<td>BKH4</td>
</tr>
<tr>
<td></td>
<td>BJK3</td>
<td>BJC4</td>
<td>BHD4</td>
</tr>
<tr>
<td></td>
<td>BJK2</td>
<td>BJC4</td>
<td>BM2C</td>
</tr>
<tr>
<td></td>
<td>BKH4</td>
<td>BJC4</td>
<td>BMG4</td>
</tr>
<tr>
<td></td>
<td>BLM4</td>
<td>BJC4</td>
<td>BMO5</td>
</tr>
<tr>
<td></td>
<td>BMC5</td>
<td>BJC4</td>
<td>BND5</td>
</tr>
<tr>
<td></td>
<td>BME5</td>
<td>BJC4</td>
<td>DMD5</td>
</tr>
<tr>
<td></td>
<td>BMD5</td>
<td>BJC4</td>
<td>DME5</td>
</tr>
<tr>
<td></td>
<td>BMD5</td>
<td>BJC4</td>
<td>DMD5</td>
</tr>
<tr>
<td></td>
<td>BPC4</td>
<td>BJC4</td>
<td>BPC4</td>
</tr>
<tr>
<td></td>
<td>BUA1</td>
<td>BJC4</td>
<td>BUA1</td>
</tr>
<tr>
<td></td>
<td>BUB1</td>
<td>BJC4</td>
<td>BUB1</td>
</tr>
<tr>
<td></td>
<td>BUB5</td>
<td>BJC4</td>
<td>BUB5</td>
</tr>
<tr>
<td></td>
<td>BUC1</td>
<td>BJC4</td>
<td>BUC1</td>
</tr>
<tr>
<td></td>
<td>BUC5</td>
<td>BJC4</td>
<td>BUC5</td>
</tr>
<tr>
<td></td>
<td>BUC6</td>
<td>BJC4</td>
<td>BUC6</td>
</tr>
<tr>
<td></td>
<td>BUK4</td>
<td>BJC4</td>
<td>BUK4</td>
</tr>
<tr>
<td></td>
<td>BUC5</td>
<td>BJC4</td>
<td>BUC5</td>
</tr>
<tr>
<td></td>
<td>BUD3</td>
<td>BJC4</td>
<td>BUD3</td>
</tr>
<tr>
<td></td>
<td>BUC6</td>
<td>BJC4</td>
<td>BUC6</td>
</tr>
<tr>
<td></td>
<td>BUC5</td>
<td>BJC4</td>
<td>BUC5</td>
</tr>
<tr>
<td></td>
<td>BUC6</td>
<td>BJC4</td>
<td>BUC6</td>
</tr>
</tbody>
</table>

**BCD2** | **BHN3**
### Table C — Subroutine Usage

This table lists the subroutine names as they appear in each subroutine block on the individual charts. The subroutine name and the block location where it may be found appears on the first line. To the right of this and on subsequent lines is a list of all chart and block locations where this subroutine is used.

<table>
<thead>
<tr>
<th>SUBROUTINE</th>
<th>CONN</th>
<th>BLOCK</th>
<th>SUBROUTINE</th>
<th>CONN</th>
<th>BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>IACTIV</td>
<td>CAA2</td>
<td>BNE2</td>
<td>CVRPT</td>
<td>CWA5</td>
<td>CJG2</td>
</tr>
<tr>
<td>IODECVA</td>
<td>BUG1</td>
<td>BG1</td>
<td>CJK3</td>
<td>CLG5</td>
<td>CNG5</td>
</tr>
<tr>
<td>IODECV0</td>
<td>BRE4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDATA</td>
<td>CLA1</td>
<td>BM02</td>
<td>DATNS</td>
<td>CVA1</td>
<td>CDG2</td>
</tr>
<tr>
<td>PAUSE</td>
<td>CTA2</td>
<td>BTG1</td>
<td>FDAMT</td>
<td>CVA3</td>
<td>BRF4</td>
</tr>
<tr>
<td>PREOUT</td>
<td>CWB1</td>
<td>BUH1</td>
<td>IBVAL</td>
<td>BVA3</td>
<td>BAH4</td>
</tr>
<tr>
<td>ACTV</td>
<td>CAA2</td>
<td>BCH2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTV+1</td>
<td>CAA2</td>
<td>CLG3</td>
<td>IFIGRO</td>
<td>BHA2</td>
<td>BGG2</td>
</tr>
<tr>
<td>BCDHC</td>
<td>CWF2</td>
<td>INITUS</td>
<td>BWA2</td>
<td>BAG5</td>
<td>BCE1</td>
</tr>
<tr>
<td>BCD5</td>
<td>CUF7</td>
<td></td>
<td></td>
<td></td>
<td>BDE2</td>
</tr>
<tr>
<td>BCVDEC</td>
<td>BT4H</td>
<td>CKB3</td>
<td></td>
<td></td>
<td>BGE2</td>
</tr>
<tr>
<td>BCVDEC-1</td>
<td>BTG4</td>
<td>CKB3</td>
<td></td>
<td></td>
<td>BRF4</td>
</tr>
<tr>
<td>CALIB1</td>
<td>BPA2</td>
<td>BDH2</td>
<td>ISSEK</td>
<td>CSB2</td>
<td>CBE4</td>
</tr>
<tr>
<td>CALPW</td>
<td>BSA1</td>
<td>BRD1</td>
<td></td>
<td></td>
<td>CDB4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BRE3</td>
<td></td>
<td></td>
<td>CFE4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BKE3</td>
<td></td>
<td></td>
<td>CFP3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BFS1</td>
<td></td>
<td></td>
<td>CFP3</td>
</tr>
<tr>
<td>CAL5Y</td>
<td>BSA4</td>
<td>BRC1</td>
<td>MWR</td>
<td>CWB1</td>
<td>CCA5</td>
</tr>
<tr>
<td>CAL6Y</td>
<td>BSA5</td>
<td>BNE1</td>
<td></td>
<td></td>
<td>CJU2</td>
</tr>
<tr>
<td>CAL7Y</td>
<td>BSA5</td>
<td>BNE2</td>
<td>NOSEL</td>
<td>CAA1</td>
<td>BHF4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BHK4</td>
</tr>
<tr>
<td>CALBY</td>
<td>BSA3</td>
<td>BF5</td>
<td></td>
<td></td>
<td>BF6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUBROUTINE</th>
<th>CONN</th>
<th>BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDS3</td>
<td>CMB2</td>
<td>CLG4</td>
</tr>
<tr>
<td>NRTST</td>
<td>CKA4</td>
<td>CEE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHD2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CJ42</td>
</tr>
<tr>
<td>PAUSE</td>
<td>CTA2</td>
<td>BCF3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BJG2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BJG3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BUR2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BUR3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CGS5</td>
</tr>
<tr>
<td>PAWS</td>
<td>CFB2</td>
<td>BEH4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CCB8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CDG3</td>
</tr>
<tr>
<td>PROUT</td>
<td>CMB1</td>
<td>BBH4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BHE3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BHE3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BHA2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BFE4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CCA2</td>
</tr>
<tr>
<td>BCDHC</td>
<td>CWB1</td>
<td>BAG5</td>
</tr>
<tr>
<td>BCD5</td>
<td>CKB3</td>
<td>BCE1</td>
</tr>
<tr>
<td>BCVDEC</td>
<td>CKB3</td>
<td>BDE2</td>
</tr>
<tr>
<td>BCVDEC-1</td>
<td>CKB3</td>
<td>BGE2</td>
</tr>
<tr>
<td>CALIB1</td>
<td>BSA2</td>
<td>BHA2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BAG5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCE1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BDE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BGE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BRF4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BAG5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCE1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BDE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BGE2</td>
</tr>
<tr>
<td>CALPW</td>
<td>BSA1</td>
<td>BRC1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MWR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CWB1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CCA5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CJU2</td>
</tr>
<tr>
<td>CAL5Y</td>
<td>BSA4</td>
<td>BFR1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISSEK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CSB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CBE4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CDB4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CFE4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CFP3</td>
</tr>
<tr>
<td>CAL6Y</td>
<td>BSA5</td>
<td>BFR1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISSEK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CSB2</td>
</tr>
<tr>
<td>CAL7Y</td>
<td>BSA5</td>
<td>BFR1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISSEK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CSB2</td>
</tr>
<tr>
<td>CALBY</td>
<td>BSA3</td>
<td>BFR5</td>
</tr>
</tbody>
</table>

Appendix G — Autochart Reference Tables 113
EVALUATION SHEET

IBM 7090 / 7094 IBSYS / IOEX
PROGRAMMING SYSTEMS ANALYSIS GUIDE, FORM C28-6299

FROM

NAME ________________________________

LOCATION ________________________________

1. HOW CAN THIS MANUAL BE IMPROVED TO BETTER SERVE YOUR NEEDS?

2. LIST ERRORS AND DELETIONS (GIVE PAGE NUMBER)
ATTN: PROGRAM ANALYSIS, DEPARTMENT D58