Airport Landside
Volume IV: Appendix A
ALSIM AUXILIARY and MAIN Programs

L. McCabe
M. Gorstein

Transportation Systems Center
Cambridge MA 02142

June 1982
Final Report

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

U.S. Department of Transportation
Federal Aviation Administration

Office of Systems Engineering and Management
Washington DC 20691
This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.
This Appendix describes the Program Logic of the Airport Landside Simulation Model (ALSIM) AUXILIARY and MAIN Programs. Both programs are written in GPSS-V. The AUXILIARY program is operated prior to the MAIN Program to create GPSS transactions representing Enplaning Passenger groups from the input flight schedule. The transactions are written on a JOBTAPE file for subsequent use by the MAIN program. The MAIN Program creates greeter and deplaning passenger transactions and enacts the movements of all passengers and visitors through the landside.

Details of program logic and flow charts at the GPSS block level are provided. A listing of both programs is included.

Other volumes of the Airport Landside report are: Volume I: Planning Guide; Volume II: ALSIM Description and User's Guide; Volume III: ALSIM Calibration and Validation; and, Volume V: Appendix B ALSIM Subroutines.
SUMMARY

This document provides a detailed description of the MAIN and AUXILIARY programs of the Airport Landside Simulation Model (ALSIM). Both programs are written in GPSS-V. The MAIN program simulates the movement of enplaning and deplaning passengers through the landside. Statistics are obtainable from this program for every simulated facility. These include queue length averages and maxima, queueing time averages and distributions, and service characteristics of the facility.

The AUXILIARY program is operated prior to the MAIN program to generate and store transactions representing enplaning passengers. These transactions are fed into the MAIN program as the simulation clock advances to the time of arrival on the landside by the simulated passenger.

This document describes the general structure, input variables and matrices used by each program. The use of passenger transaction parameters and a detailed description of program logic are also contained in this document. Flowcharts and listings for each program are exhibited.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUXILIARY PROGRAM</td>
<td></td>
</tr>
<tr>
<td>General Description</td>
<td>A-1-3</td>
</tr>
<tr>
<td>Program Definition Statement Section</td>
<td>A-1-3</td>
</tr>
<tr>
<td>Enplaning Passenger Creation Section</td>
<td>A-1-17</td>
</tr>
<tr>
<td>Program Control Section</td>
<td>A-1-23</td>
</tr>
<tr>
<td>MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td>General Description</td>
<td>A-1-25</td>
</tr>
<tr>
<td>Program Definition Area</td>
<td>A-1-26</td>
</tr>
<tr>
<td>Control Section</td>
<td>A-1-37</td>
</tr>
<tr>
<td>Deplaning Passenger Logic</td>
<td>A-1-40</td>
</tr>
<tr>
<td>Enplaning Passenger Logic</td>
<td>A-1-56</td>
</tr>
<tr>
<td>Facility Modules Section</td>
<td>A-1-58</td>
</tr>
<tr>
<td>1. Bag Claim</td>
<td>A-1-58</td>
</tr>
<tr>
<td>2. Baggage Unloading Logic</td>
<td>A-1-60</td>
</tr>
<tr>
<td>3. Ticketing and Check-in</td>
<td>A-1-62</td>
</tr>
<tr>
<td>4. Immigration</td>
<td>A-1-64</td>
</tr>
<tr>
<td>5. Customs</td>
<td>A-1-65</td>
</tr>
<tr>
<td>6. Concourse Exit</td>
<td>A-1-65</td>
</tr>
<tr>
<td>7. Deplaning Curb (Cars)</td>
<td>A-1-66</td>
</tr>
<tr>
<td>8. Deplaning Curb (Pax)</td>
<td>A-1-73</td>
</tr>
<tr>
<td>9. Enplaning Curb</td>
<td>A-1-78</td>
</tr>
<tr>
<td>10. Entrance</td>
<td>A-1-84</td>
</tr>
<tr>
<td>11. Exit</td>
<td>A-1-85</td>
</tr>
<tr>
<td>12. Gate (Enplaning Pax)</td>
<td>A-1-86</td>
</tr>
<tr>
<td>13. Ground Transportation (Misc.)</td>
<td>A-1-90</td>
</tr>
<tr>
<td>15. Rent-A-Car</td>
<td>A-1-93</td>
</tr>
<tr>
<td>16. Security</td>
<td>A-1-95</td>
</tr>
<tr>
<td>17. Concessions</td>
<td>A-1-97</td>
</tr>
<tr>
<td>Transfer Flights</td>
<td>A-1-98</td>
</tr>
<tr>
<td>Change Card Reader</td>
<td>A-1-103</td>
</tr>
<tr>
<td>Time Section</td>
<td>A-1-108</td>
</tr>
<tr>
<td>APPENDIX A-2 FLOW CHARTS</td>
<td>A-2-1</td>
</tr>
<tr>
<td>APPENDIX A-3 PROGRAM LISTING:</td>
<td>A-3-1</td>
</tr>
</tbody>
</table>
VOLUME IV

APPENDIX A-1

ALSIM DOCUMENTATION-GPSS-V
AUXILIARY AND MAIN PROGRAMS

L. MCCABE AND R. WALKER

A-1-1/A-1-2
AUXILIARY PROGRAM

General Description

The Auxiliary Program shown in Figure A-1 generates transactions representing enplaning passenger groups. These are stored on a JOBTAPE and utilized by the main simulation program when required. This program assigns attributes to the transaction parameters for representing ground transportation modal choice, ticketed or non-ticketed status, passenger group size and number of accompanying well wishers. The program contains three sections, (1) A Program Definition Statements Section, (2) Enplaning Passenger Creation Section and, (3) A Program Control Section. The Program Definition Statements Section defines matrix sizes, cumulative distribution functions and simulation variables. The Enplaning Passenger Creation Section generates enplaning passenger transactions and assigns well wisher numbers and transaction attributes. The program control section initiates linkage programs between the GPSS Auxiliary Program and the FORTRAN supporting subprogram FORTM. It also creates dummy transactions to begin and end the JOBTAPE.

Program Definition Statement Section

The section begins with an R MULT statement to specify starting numbers for random number generator multipliers. The subrouting FORTM is kept resident in core during the operation of this simulation by a LOAD block. Halfword save value 1 is defined as CLXXH by an EQU statement.
Assistant Program

GENERATE

1,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100

L.1.PB
ASSIGN

2.1.PB

Initialize process function pointer

7.1.PB
ASSIGN

Mark as enplaning pax.

ENPLO

ASSIGN

Flight table (MHL) row subscript

MHL(PHL,1)

TEST 0

If MHL(PHL,1) = 1, then end of flight table

TERMINATE

MHL(PHL,1)

TEST 1

Skip arriving flights

A
SPLIT Spli for next flight

3, MHL (PH, 7, PB) - Type of flight (1, 2, 3 = DOM, COM, INT)

ASSIGN

5, MHL (PH', 2, PH) - Gate no.

ASSIGN

SPLIT

To originating pax logic

MHL (PH, 16) - Fall through if no transfer out of system pax

LE TEST

TERMINATE

8, PB - Mark as transfer out of system.

ASSIGN
All transfers are single pax

... with no visitors

Delay to 40 minutes before departure

Create transfers out of system transactions

Arrival time for transfers out of system

A-1-6
Are there no pax on flight

Set no. of pax to 1 as default

Is departure time before 9:05 am

Delay to 85 minutes before departure for those departing between 4:05 am and 9:04 am

No. of originating pax

No. of pax in party

Subtract from total

Fall through if total less than or equal to zero

A-1-7
Adjust last party size

Is pax in party less than or equal to zero

Set pax in party size to one as default.

Jump out of loop

Go back to create another

Arrive time before flight for pax leaving between 4:05am and 9:00am

Enpl2
Delay to 150 minutes before departure for those departing between 9:05 am and 4:00 pm.

No. of originating pax

No. of pax in party

Subtract from total

Fall through if total less than or equal to zero.

Adjust last party size

Fall through if PBL13 is less than or equal to zero.

Set pax in party size to one as default
EJump out of loop

Go back to create another

Arrival time prior to flight for pax departing between 9:05 am and 4:00 pm

Ground transportation mode section

Number of party

Fall through if curb or park

HELP A FORM 5,RN4,RN5,PB3

ASSIGN

PB6 LE TEST 2 J

K
ASSIGN

Add in well-wishers

WRITE

Write enplaning transactions on jobtape

TERMINATE

GENERATE

Generate dummy transaction for jobtape

HELP C

Call linking subroutines

HELP A

Completes FORTM-GPSS linking, and reads input data
Test for error
in FORMAT

Used to set up
interarrival times

Prevent jobtape
end message

Wait for above transaction
to be written
Data required for Auxiliary Program execution is contained in three matrices and five functions. Matrices used are: halfword matrix MH1, the flight schedule; halfword matrix MH4, the percent of enplaned passengers preticketed; and, floating point matrix ML2, ground transaction modal choice percentages. Information contained in these matrices is provided by input data read and placed in the matrix elements by FORTM.

The numbers of columns of MH1, MH4, and ML2 are represented by the entity symbols CMHO1, CMHO4 and CMLO2 respectively. These are assigned numerical values by SYN statements. These values will be transferred to FORTM by the use of a mnemonic link function which appears in the program control section and the assembler subprogram MNLINK called by FORTM.

Matrix type and dimensions of MH1, MH4, and ML2 are specified by matrix definition statements. The column numbers used must agree with preceding SYN statements. Tables 1 and 2 describe matrix contents.

Functions used by this program are: ARV1F and ARV2F, arrival times at the airport landside of originating passengers PPEF, passengers per enplaning group; WWPPF, well wishers per enplaning group and TOS1F, the arrival times of transfer passengers at a concourse.

Because of a time dependence in arrival rates, two cumulative distribution functions are used to specify the
## Auxiliary Program

### Table 1

Contents of Flight Table Matrix MHL

<table>
<thead>
<tr>
<th>Column</th>
<th>Usage</th>
<th>FORTRAN Data Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Designates arriving or departing flight. Indicates completion of aircraft bag unloading for arrival flights.</td>
<td>ARRV, DEPT</td>
</tr>
<tr>
<td></td>
<td>0 = Arriving Flight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Departing Flight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 = Bag unloading complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1 = End of table</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Flight number.</td>
<td>FLTNO</td>
</tr>
<tr>
<td>3</td>
<td>Airline number.</td>
<td>AIRLIN</td>
</tr>
<tr>
<td>4</td>
<td>Scheduled arrival or departure time.</td>
<td>TIME</td>
</tr>
<tr>
<td>5</td>
<td>Number of deplaning passengers met by private car.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Flight arrival or departure time in minutes relative to simulation start time</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Domestic, commuter, or international flight. Domestic is default value.</td>
<td>DOM = 1, INT = 1, COM = 1</td>
</tr>
<tr>
<td>8</td>
<td>Aircraft type.</td>
<td>AC</td>
</tr>
<tr>
<td>9</td>
<td>Gate number.</td>
<td>GATE</td>
</tr>
<tr>
<td>10</td>
<td>Numbers of originating or terminating passengers.</td>
<td>PAX - TPAX</td>
</tr>
<tr>
<td>11</td>
<td>Transfer passengers.</td>
<td>TPAX(1)</td>
</tr>
<tr>
<td>12</td>
<td>(a) Bag claim area for arriving flight. (a) BAG</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(b) Number of enplaning passengers waiting at departure lounge for boarding. (b) No input data.</td>
<td></td>
</tr>
</tbody>
</table>

A-1-14
<table>
<thead>
<tr>
<th>Column</th>
<th>FORTRAN Data Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Number of through passengers on flight using terminal facilities.</td>
</tr>
<tr>
<td>14</td>
<td>Total terminating bags on arriving flight.</td>
</tr>
<tr>
<td>15</td>
<td>Total transfer bags on arriving flight.</td>
</tr>
<tr>
<td>16</td>
<td>Number of passengers originating from or proceeding to flights outside of simulated system.</td>
</tr>
</tbody>
</table>
### Table 2.

Contents of Matrices MH4 and ML2

<table>
<thead>
<tr>
<th>MH4 Element</th>
<th>Usage</th>
<th>FORTRAN Input &amp; Preticketed Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,1</td>
<td>Percent of enplaning passengers preticketed for domestic flights.</td>
<td>DOM</td>
</tr>
<tr>
<td>1,2</td>
<td>Percent of preticketed enplaning domestic passengers proceeding from terminal entrance directly to security.</td>
<td>DOMDIR</td>
</tr>
<tr>
<td>2,1</td>
<td>Same as 1,1 for commuter flights.</td>
<td>COM</td>
</tr>
<tr>
<td>2,2</td>
<td>Same as 1,2 for commuter flights.</td>
<td>COMDIR</td>
</tr>
<tr>
<td>3,1</td>
<td>Same as 1,1 for international flights.</td>
<td>INT</td>
</tr>
<tr>
<td>3,2</td>
<td>Same as 1,2 for international flights</td>
<td>INTDIR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ML2 Element</th>
<th>n = 1/2/3 DOM/COM/INT</th>
<th>FORTRAN Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>n,1</td>
<td>Percent of enplaning passengers using auto mode</td>
<td>PVT C AR</td>
</tr>
<tr>
<td>n,2</td>
<td>Percent of enplaning passengers using rental car.</td>
<td>CRENT</td>
</tr>
<tr>
<td>n,3</td>
<td>Percent of enplaning passengers using bus or limousine.</td>
<td>BUS</td>
</tr>
<tr>
<td>n,4</td>
<td>Percent of enplaning passengers using taxi.</td>
<td>TAXI</td>
</tr>
</tbody>
</table>
starting times of transactions representing originating passengers. ARV1F is the distribution of arrival times prior to flight of passengers on flight departing between 0905 and 2400 hours. ARV2F is the distribution used for all other departure times.

The arrival time distribution, TOS1F, is used for simulation of transfer passengers when the system modeled does not include all concourses of an airport under study. The transfer passengers in this case originate outside the system. This function provides a starting time for transactions representing this passenger type.

Variable FLT1V and FLT2V delay the entry of a transaction into the simulation until 150 and 85 minutes respectively. FLT1V is used for transactions representing originating passengers of flights departing between 0905 and 2400. FLT2V is applied to all other originating passenger transactions representing transfers outside the system until 40 minutes before flight departure. A fourth variable, TOS2V is equated to MHI(PHI,16)-1. This is the number of passengers originating outside the system, minus one.

Enplaning Passenger Creation Section

A single transaction representing all departing flights is generated. This has a priority level 10 and contains 11 halfword parameters and 14 byte parameters. This transaction will later be split to represent each flight and subsequently split to represent individual passenger groups. Table 3 lists
### Table 3.

**Passenger Transaction Parameters**

#### Halfword Parameters

<table>
<thead>
<tr>
<th>PH</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Flight table row number</td>
</tr>
<tr>
<td>2</td>
<td>Current location, point number</td>
</tr>
<tr>
<td>3</td>
<td>a. Maximum random number from subroutine BAGS for deplaning passengers.</td>
</tr>
<tr>
<td></td>
<td>b. Passenger storage number in deplaning curb logic.</td>
</tr>
<tr>
<td>4</td>
<td>Address parameter.</td>
</tr>
<tr>
<td>5</td>
<td>Scratch parameter. Initially contains gate number for deplaning passenger. Used to designate GPSS storage and queue numbers during passenger processing.</td>
</tr>
<tr>
<td>6</td>
<td>a. User chain number for bag claim simulation.</td>
</tr>
<tr>
<td></td>
<td>b. Storage number of deplaning curb.</td>
</tr>
<tr>
<td>7</td>
<td>Facility matrix, MH9, row number for current processing operation.</td>
</tr>
<tr>
<td>8</td>
<td>Scratch.</td>
</tr>
<tr>
<td>9</td>
<td>Cumulative walking time.</td>
</tr>
<tr>
<td>10</td>
<td>Transaction sequence number for passenger and greeter matching.</td>
</tr>
<tr>
<td>11</td>
<td>Cumulative passenger waiting time.</td>
</tr>
</tbody>
</table>

#### Byte Parameters

<table>
<thead>
<tr>
<th>PB</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GPSS number of processing function assigned to this transaction.</td>
</tr>
<tr>
<td>2</td>
<td>Pointer indicating current step of processing function.</td>
</tr>
<tr>
<td>3</td>
<td>Flag indicating type of flight (1 = Domestic, 2 = Commuter, 3 = International).</td>
</tr>
<tr>
<td>4</td>
<td>Number of bags.</td>
</tr>
<tr>
<td>5</td>
<td>Number of passengers and visitors in party</td>
</tr>
<tr>
<td>6</td>
<td>Mode of ground transportation ( PVTCAR CURB=1, PVTCAR PARK=2, RENTAL=3, BUS or LIMO=4, TAXI=5)</td>
</tr>
</tbody>
</table>

A-1-18
Table 3 (cont.)

<table>
<thead>
<tr>
<th>PB 7</th>
<th>Flag indicating deplaning or enplaning passenger.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Deplaning</td>
</tr>
<tr>
<td>1</td>
<td>Enplaning</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PB 8</th>
<th>Flag indicating category of deplaning passenger.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Terminating</td>
</tr>
<tr>
<td>2</td>
<td>Transfer</td>
</tr>
<tr>
<td>3</td>
<td>Transmit</td>
</tr>
<tr>
<td>4</td>
<td>Transfer out of system</td>
</tr>
</tbody>
</table>

| PB 9 | Flag to designate ticketed or non-ticketed status. |

| PB 10 | Car rental agency number. |

| PB 11 | Number of facility type currently entered by transaction. |

<table>
<thead>
<tr>
<th>PB 12</th>
<th>Flag to designate meeting location of deplaning passenger and greeter.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gate = 1</td>
</tr>
<tr>
<td></td>
<td>Bag Claim = 2</td>
</tr>
<tr>
<td></td>
<td>Ticketing = 3</td>
</tr>
</tbody>
</table>

| PB 13 | Passengers in party. |

<table>
<thead>
<tr>
<th>PB 14</th>
<th>a. GPSS number of bag per passenger distribution function.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b. Parking lot number.</td>
</tr>
</tbody>
</table>
the information contained in the transaction parameters describing the passenger group.

Byte parameters 2 and 7 are assigned a value of 1, to respectively initialize the process function pointer and mark the transaction as a representation of an enplaning passenger. At block ENPLO, PH1 is first initialized to a value of 1 and on successive passes is incremented by one to represent the flight table row subscript.

A loop beginning at ENPLO and ending with the block SPLIT 1, ENPLO creates a transaction for each departing flight. The element PH1,l of MH1 is tested for a negative value contained in the row following the last flight. The last transaction is terminated for this condition. All previous transactions are tested for a value of 1 in MH1(PH1,1) to determine if the row represents an arriving or departing flight. Arriving flights are ignored in the Auxiliary Program and a branch back to ENPLO is executed. Departing flight transactions continue to the SPLIT 1, ENPLO block. The parent transaction proceeds to the next block and the copy transaction to ENPLO for the next flight.

Flight type and gate number are assigned to PB3 and PH5 from columns 7 and 9 of MH1, respectively. A split block to generate two types of enplaning passengers sends one copy to ENPL3. The first category consists of transfer passengers originating outside of the system. The model tests MH1(PH1,16)
for values less than or equal to zero to determine if any passengers of this class were input to the flight schedule. If none, the parent transaction is terminated. For positive values, PB8 is assigned a value of 4 to mark this transaction as a transfer out of the system and PB13 and PB5 are set to one to indicate a single passenger with no visitors. The time of entry into the simulation is first calculated as 40 minutes before flight departure time by the variable TOS1V. A SPLIT block creates all of the transactions to fill the number, TOS2V, of transfers out of the system for this flight and directs them to the next block. Individual transactions are further assigned a simulation entry time between 20 and 40 minutes before flight time by delaying them a time randomly selected from the function TOS1F. When the simulation clock advances to this entry time, they are transferred to the WRITE block for writing onto the JOBTape.

At ENPL3, transactions representing originating passengers test MH1(PH1,10), for a zero value for this occurrence, a default value of 1 is placed in the element. Otherwise, the program branches to a test of the flight departure time in MH1(PH1,10), to determine if it occurs between 0000 and 0905. For departures outside this interval, the program branches to ENPL1. Flights inside the interval proceed to the next block which holds the transaction until 85 minutes before departure. Transaction parameter PH8 is made equal to the number of originating passengers. The number of passengers in a party is assigned to PB13.
by a random draw of function PPPEF. These are subtracted from the total in PH8. A test is performed to determine if PH8 is zero or negative. A positive value, indicating that more passenger group transactions will be required, transfers the transaction down to a SPLIT block. This creates a copy transaction to represent another passenger group and transfers it back to assign PPPEF to PB13. A zero or negative value of PH8 allows the transaction to drop through to the next block where the value in this parameter is added to PB13 to adjust the last party size. If PB13 becomes negative, the value of this parameter is made equal to 1. The transaction transfers out of the loop to an ADVANCE block which further delays the transaction activity by an amount obtained from a random draw on function ARV2F.

Flights departing between 0905 and 2400 are routed to ENPL1. The creation of passenger group transactions is performed exactly as before except that these are delayed on variable FLT1V and a random draw on function ARV1F.

After delays, all transactions proceed to ENPL2 for a HELPA call to subroutine FORTM. Random numbers produced by generators 4 and 5 and flight type in PB3 are used as operands. The subroutine assigns a preticketed status and ground transportation modal choice to PB9 and PB6, respectively.

The number of passengers in the party is assigned to PB5. If PB6 is 1 or 2, indicating a private vehicle proceeding to curbside or parking lot, well wishers are added to PB5, based upon a random draw on function WWPPF.
All transactions are written in time sequence on JOBTAl at location WRITE. These transactions are then terminated.

Program Control Section

A single transaction with priority level 127 and one halfword parameter is generated to initiate and, subsequently, to terminate the activity of the Auxiliary Program. A HELPA call to subroutine CLINK initiates linking of this program and the supporting FORTRAN subroutine FORTM. The next block is a HELPA call to FORTM which completes the linking. After execution of this statement, any HELPA call to FORTM appearing in the Auxiliary Program will operate with two-way communication between the two programs. FORTM also reads input data and places the information in matrices MH1, MH4, and ML2. Any errors detected by FORTM are indicated by setting PHI of this transaction to a value greater than zero. Upon return from FORTM, PHI is tested for a value greater than zero. For this condition, the transaction is terminated with a termination count equal to PHI and the simulation stops.

A zero value of PHI, when no errors are indicated, transfers the transaction to a SPLIT block. The copy is sent to the program location WRITE to become the first transaction on the JOBTAPE. This is performed to provide an initial time, coincident with the start of the simulation to which all succeeding transaction arrival times are referenced on the JOBTAPE.

A-1-23
The first dummy transaction is terminated immediately after entry into the main program from the JOBTAPE.

The original transaction is delayed 100,000 seconds, a time far exceeding a normal simulation time length. Following this delay, the transaction is again split and the second copy is written as the last transaction on the JOBTAPE. This writing prevents an END OF JOBTAPE message during main program execution.

The original transaction is delayed 1 second to permit the WRITE block to execute before the final terminate block is executed. When this occurs the termination count is incremented by 1.

List function number 1 is used to transmit values between the main program and FORTM. Mnemonics are used as Y values of this function. These mnemonics have been previously assigned numerical values by the program. Corresponding mnemonics are used in the FORTRAN subroutine FORTM after calling subroutine MNLNK. Values represented by these mnemonics may be passed in either direction. The entity symbols CMHO1, CMHO4, CMLO2 and CLKXH were previously assigned values by SYN statements and are passed to FORTM. This list is placed in this program location for potential expansion for future requirements. Any GPSS entity symbolically represented may be used in this function.

A START statement with a run termination count 1 starts the program and normally stops it after the control transaction is written on the JOBTAPE. This block will also cease program operation if errors are determined in the subroutine FORTM.

A-1-24
MAIN PROGRAM

General Description

The Air-port Landside Simulation main program enacts the movement of passengers and visitors through the terminal area. This program generates deplaning passenger and greeter transactions, merges them with enplaning passenger and well-wisher transactions generated by the auxiliary program and models their flow through a sequence of simulated landside processing facilities. Flow, waiting time, queue length and occupancy data are produced for analysis.

The main program contains the following sections: program definitions area, deplaning and enplaning logic sections, facility modules, control section, timer section, transfer flight schedule area, and facility server reallocation section. This document describes those sections in detail and provides a flow chart of GPSS blocks used.
PROGRAM DEFINITIONS AREA

This section contains a description by statements defining limits of GPSS entity numbers, system instructions, SYN and EQU statements to identify facilities, matrix definition statements, function and variable definitions and table definition statements.

Limits on GPSS entity numbers are established by REALLOCATE statements. This program requires entity limits differing from those provided by default values. The program listing contains the values currently used for this simulation model.

System instructions includes an RMULT statement to provide different random number sequences by changing random number generator multipliers. A LOAD statement retains the FORTRAN subroutine, FORTM, and the assembler subroutine BAGS, in core during program execution. The GPSS output module, DAG06, and an assembler program XACNO, are also kept resident. A JOBTAPE statement defines JOBTAl as the file of enplaning passenger transactions created by the auxiliary program, and ENPLO as their entry block.

SYN statements are used to pass argument values symbolically to the FORTRAN subroutine, FORTM. A list function in the timer section contains the entity symbols used in the SYN statements as Y values. A HELP call to FORTM provides a call to the assembler program MNLINK. The arguments of the FORTRAN CALL statement contain variable names which are positionally identified with the list function Y values after NMLINK is executed. The absolute values of the symbols are assigned by the SYN statements of this section.
and are passed to FORTM. These values specify the first entity numbers assigned by EQU statements used to reserve a sequenced set of GPSS entities. The sequential set is used for simulation of all landside facilities of a given type, for example, all security stations. The types of GPSS entities specified are logic switches, user chains, queues and storages. When the program is executed, the number contained in the EQU sequential set size field for a given facility type must equal or exceed the input parameter FACNO for that type.

SYN statements are also used to transfer the number of columns contained in matrices to FORTM. Halfword matrices 1 through 4, 6 through 9, and floating point matrix 2, utilize this feature.

Matrix definition statements specify matrix numbers type and size. Numbers of columns must agree with preceding SYN statements. Contents of halfword matrices 2, 3, five, 6, and 7 are listed in Table 4. Information for halfword matrix 8 is in Table 5 and matrix 9 in Table 6. Matrices MH 11 and MH 12 are one column matrices used to accumulate counts of persons entering and leaving concourses.

Floating point matrix, ML1, is a single row, 127 column matrix used as a table of floating point random numbers ranging from 0.0 to 1.0. This matrix is filled with random numbers by a group of blocks in the timer section.
TABLE 4
CONTENTS OF MATRICES 2, 3, 5, 6, 7

MH2 - Airline Information Table
1 row per carrier

<table>
<thead>
<tr>
<th>Column</th>
<th>FORTRAN Input Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Enplaning curb facility no.</td>
<td>Epcurb</td>
</tr>
<tr>
<td>2 - Per cent preticketed pax using express check-in (0 = No Express Check-in)</td>
<td>Expchk</td>
</tr>
<tr>
<td>3 - Enplaning curb point for bus stop (If different from column 1)</td>
<td>Bustop</td>
</tr>
</tbody>
</table>

MH3 - Table of Points
1 row per carrier

<table>
<thead>
<tr>
<th>Column</th>
<th>FORTRAN Input Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - X coordinate</td>
<td>POINTX or XY(1)</td>
</tr>
<tr>
<td>2 - Y coordinate</td>
<td>POINTY or XY(2)</td>
</tr>
<tr>
<td>3 - Nearest exit point no.</td>
<td></td>
</tr>
<tr>
<td>4 - Nearest entrance point no.</td>
<td></td>
</tr>
</tbody>
</table>

MH5 - Transfer Flight Table
Contains MH1 row of departing flights taking transfer passengers

MH6 - Walking time between points.

MH7 - Used as work area by bag claim routines
1 row for each possible random number, 1-64, generated by "BAGS"
MH8 - Used to access facility data in MH9 (contains same information as FORTRAN 'NFASCM' array)

<table>
<thead>
<tr>
<th>Number of Facilities in Type</th>
<th>Index Number of Facility Type (one less than number of first facility in type in MH9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gate</td>
<td>1,1</td>
</tr>
<tr>
<td>Check-in</td>
<td>2,1</td>
</tr>
<tr>
<td>Security</td>
<td>3,1</td>
</tr>
<tr>
<td>Bagclaim</td>
<td>4,1</td>
</tr>
<tr>
<td>Customs</td>
<td>5,1</td>
</tr>
<tr>
<td>Entrance</td>
<td>6,1</td>
</tr>
<tr>
<td>Exit</td>
<td>7,1</td>
</tr>
<tr>
<td>Enplaning curb</td>
<td>8,1</td>
</tr>
<tr>
<td>Transfer (stairs, etc.)</td>
<td>9,1</td>
</tr>
<tr>
<td>Parking</td>
<td>10,1</td>
</tr>
<tr>
<td>Rentacar</td>
<td>11,1</td>
</tr>
<tr>
<td>Deplaning curb</td>
<td>12,1</td>
</tr>
<tr>
<td>Immigration</td>
<td>13,1</td>
</tr>
<tr>
<td>Tickets &amp; check-in</td>
<td>14,1</td>
</tr>
</tbody>
</table>
TABLE 6

MH9 - Facility Table
1 Row per actual or dummy facility

Column
1 Facility type
2 Facility number within type
3 Location (point Number)
4-6 Used to identify a facility with other model components

<table>
<thead>
<tr>
<th>Facility(1)</th>
<th>Column No.</th>
<th>Facility(2)</th>
<th>FORTRAN Input Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bagclaim</td>
<td>4</td>
<td>Deplaning curb</td>
<td>NDELPC</td>
</tr>
<tr>
<td>Customs</td>
<td>4</td>
<td>Deplaning curb</td>
<td>NDELPC</td>
</tr>
<tr>
<td>Gate</td>
<td>4</td>
<td>Security</td>
<td>NSECUR</td>
</tr>
<tr>
<td>Gate</td>
<td>5</td>
<td>Immigration area</td>
<td>NIMMI</td>
</tr>
<tr>
<td>Immigration</td>
<td>4</td>
<td>Customs</td>
<td>NCUST</td>
</tr>
<tr>
<td>Rentacar</td>
<td>4</td>
<td>Agency code</td>
<td>AGENCY</td>
</tr>
<tr>
<td>Rentacar</td>
<td>5</td>
<td>Parking</td>
<td>NPARKL</td>
</tr>
<tr>
<td>Tickets &amp; Check-in</td>
<td>4</td>
<td>Airline Code</td>
<td>AIRLIN</td>
</tr>
<tr>
<td>Concession</td>
<td>4</td>
<td>Security</td>
<td>NSECUR</td>
</tr>
</tbody>
</table>

A-1-30
The halfword matrices 1 and 4, and floating point matrix 2 have been described previously in Tables 1 and 2 of the Auxiliary Program Description Section.

Functions defined in this section of the program are used for three purposes: (1) retention of program locations representing processing facilities encountered by passenger and visitor transactions, (2) selection of functions containing these locations by random numbers draw; and (3) provision of random variable values to the program by describing cumulative distributions subject to random draw during program operation.

The first group are lists of numerical-valued functions containing symbolic program locations required to be accessed by all transactions of a class. For example, the deplaning domestic passenger is assigned the function DDPIF. Processing of this passenger requires routines to the concourse exit, CNCRO; a rental car counter, RCARO; bag claim, BAGCO; terminal building exit, EXITO; and a ground transportation facility, CGTRO at each of these locations, a transaction parameter is examined to determine if further processing is required. When no processing is simulated the transaction is directed to the next location contained in the function. Otherwise, processing is completed and the transaction is directed to the next program location. The program presently uses 22 functions of this type to describe movement of passengers, well-wishers and greeters through the landside.

A-1-31
Selection functions are either entity functions or discrete attribute-valued functions. Entity functions are used by transfer or transit passenger transactions to select routing functions based upon a random number draw. The X values are treated as arguments of discrete numerical valued functions. The argument values are supplied by a random number generator. Y values are function names.

The other type of selection function used selects ticket counter processing time based upon the airline number of the transaction currently being processed. A discrete attribute valued function performs this task. A list of the selection functions is contained in Table 7.

Cumulative distribution functions used by this program are continuous or discrete. Continuous functions describe landside facility service times, unloading times or empty vehicle parking times. Discrete distributions are used to assign quantities to transaction parameters. For example, passenger and visitor group sizes or number of bags are assigned by random selection from this function type. A list of numerical valued functions is contained in Table 8.

Arithmetic variables are used for assignment of service times to facilities. Values of functions representing the service time distribution are multiplied by the halfword representing the scale factor SCLXH. When service times are applicable to individuals instead of passenger groups, the service time is determined by multiplying the value from the distribution, the scale factor and PBl3, then number of passengers.
TABLE 7

Functions Used for Selection

TADF - Transfer passenger, long-stay/different concourse
TDLSF - Transfer passenger, long-stay/same concourse
TDLSF - Transfer passenger, short-stay/different concourse
TDSSF - Transfer passenger, short-stay/same concourse
TDPXF - Transit passenger
ATKIF - Choose airline ticket/check-in variable
BUNIF - Baggage unload entity function (aircraft type)
### TABLE 8
Cumulative Distribution Functions

<table>
<thead>
<tr>
<th>Continuous Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUN2F to BUN3F</td>
<td>Baggage time to claim area by aircraft type</td>
</tr>
<tr>
<td>CHK1F to CHK4F</td>
<td>Ticket check-in time by airline</td>
</tr>
<tr>
<td>GAT3F to GAT4F</td>
<td>Gates process time by airline</td>
</tr>
<tr>
<td>IMM1F</td>
<td>Immigration process time</td>
</tr>
<tr>
<td>CUS1F</td>
<td>Customs process time</td>
</tr>
<tr>
<td>PAR1F</td>
<td>Parking lot exit service time</td>
</tr>
<tr>
<td>RCA1F</td>
<td>Car rental processing time</td>
</tr>
<tr>
<td>SEC1F</td>
<td>Security service time per person</td>
</tr>
<tr>
<td>CSCKF</td>
<td>Curbside checkin process time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discrete Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPPDF</td>
<td>Passengers per party deplaning</td>
</tr>
<tr>
<td>GRPPPF</td>
<td>Greeters per party (parties with greeters only)</td>
</tr>
<tr>
<td>DBAGF</td>
<td>No. of bags - domestic flight</td>
</tr>
<tr>
<td>RCAZF</td>
<td>Car rental agency selection</td>
</tr>
</tbody>
</table>
plus visitors in the group. A list of service time variables is given in Table 9.

Other arithmetic variables are used to calculate aircraft and vehicle unloading times, numbers of transactions assigned to ground transportation modes, starting times relative to flight departures and GPSS storage numbers for deplaning and enplaning curb, double parking and storage areas. Two arithmetic functions are used for random number generation.

Boolean variables are used for testing conditions at curb-side to determine if congestion due to double parking will cause queueing of traffic attempting to bypass a section. Other Boolean variables test passenger transaction parameters for transfer passengers status and preticketed status.
### TABLE 9

Service Time Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHK1V - CHK4V</td>
<td>Ticket and check-in time by airline</td>
</tr>
<tr>
<td>CUS1V</td>
<td>Customs time</td>
</tr>
<tr>
<td>GAT3V to GAT4V</td>
<td>Gate service time by airline</td>
</tr>
<tr>
<td>IMMLV</td>
<td>Immigration</td>
</tr>
<tr>
<td>PARLV</td>
<td>Parking lot service time</td>
</tr>
<tr>
<td>RCA1V</td>
<td>Car rental checkout time</td>
</tr>
<tr>
<td>SECLV</td>
<td>Security service time (party)</td>
</tr>
<tr>
<td>CIRC1V</td>
<td>Vehicle recirculation time</td>
</tr>
</tbody>
</table>
Control Section

This section provides a method of routing transactions from one program logic section to another. Routing functions are assigned the byte parameter 1 by other program sections such as the enplaning and deplaning passenger logic. The transaction is transferred to the control section which, in turn, transfers it to the next program location.

The first statement of the control section at CTRLO assigns the value of the function contained in PB1 to PH4. The argument of the function is contained in PB2. Byte parameter 2 is assigned a value 1 when the transaction is created. After the transaction is routed through CTRLO, the second block of this section increments PB2 by 1. This will make PB2 point to the next value of the routing function if the transaction is later returned to CTRLO. The third block of this section CTRLL transfers the transaction to the program location contained in PH4. This location is normally the current value of the routing function in PB1.

When a transaction is directed to a location outside the routing function, the transfer is performed without incrementing PB2. Under this condition, the program logic section specifying the transfer, places the name of the next program location in PH4 and directs the transaction to CTRLL. An example of this operation is in the enplaning curb logic. A vehicle transaction unable to locate a parking space has the
address of the recirculation road logic placed in PH4 and is transferred to CTRL2 to effect this routing. After simulating the recirculation process the vehicle is directly transferred to the enplaning curb logic. Successful use of the simulated curbside will then provide a transfer to CTRL0 to continue transaction routing according to the PBl function.

A special purpose block at CTRL8 transfers deplaning passenger transactions using the parking lot from the ground transportation module to the location PARKO. This location is the starting block of a module representing the parking facility payment booth. The transfer is unconditional for transactions entering the CTRL8 block.

Secondary functions of the control section are, (a) the provision of ground transportation facility program locations to represent buses and (b) provision of locations for transaction terminations after simulation of landside processing is completed.

For transactions representing terminating deplaning passengers, the selection function CTRLF routes the transaction to the deplaning curb, parking facility, or rental car parking facility. The mode selection defined in PB6 determines the facility. Program locations to represent these facilities are placed in PH4 at CGTRO and the transfer is executed in the next block. Originating enplaning passenger transactions use the function CTR2F to select the enplaning curb parking facility or rental car return location. Transfers are performed by the value in PH4.
Transactions representing buses are generated at intervals specified by input data. The time between creation of transactions is ABUXH for buses proceeding to the enplaning curb and DBUXH for buses departing the deplaning curb. These are rotated to program locations ENPC6 and DPLC6 respectively.

The section providing transaction termination begins at the block DEP99. The total walking time PAXWT, accumulated on the landside is recorded by a TABULATE block for each transaction routed to DEP99, END99, or TRX99.

This section also tabulates the total simulated waiting time spent on the landside by transactions. A new table is produced for each simulated hour. The hour is designated by the value of the byte savevalue PXTBN. The transaction is entered into the table by a second TABULATE block and then terminated.

In general, deplaning passenger transactions are routed to DEP99, enplaning passenger transactions to ENP99 and transfer passenger transactions to TRX99.
DEPLANING PASSENGER LOGIC

This section of the program generates transactions representing individual flights. It then creates greeter and deplaning passenger transactions through the use of SPLIT blocks. Attributes assigned to transaction parameters by this section include: number of bags, meeting location, modal choice, sequence numbers for passenger-greeter matching, process function, party size, and parking lot number. The deplaning passenger logic simulates the discharge of passengers from the aircraft and performs the matching of those deplaning passenger and greeter transactions assigned to meet at the airplane gate.

An initial GENERATE block provides one transaction of priority level 10. This transaction contains 11 halfword parameters and 14 byte parameters which are available for attribute assignment. The process function pointer, PB2, is initialized to 1. A copy of this transaction is sent to program location XFLTO to initialize a table of transfer flights in halfword matrix five. An ADVANCE block delays the original transaction by one second to permit the transfer table initialization before flight and passenger transaction creation.

Byte parameter 11 is assigned the value 1 to indicate that the transaction begins activity at the gate. A loop beginning at program location DEPL0 creates a transaction for each arriving and departing flight. At DEPL0 the MH1 row counter, PH1, is incremented by 1 each time a transaction passes through the block. Column 1 of each row of MH1 is tested for a negative value. The row after the last flight in
the schedule contains the value -1 in this position and the last transaction is terminated when this is reached. Rows representing flights cause the transaction to branch around the terminate block to an ADVANCE block. At this location, the transaction is delayed until the difference between MH1 (PH1,6), the flight arrival or departure time relative to simulation start, and the absolute clock time is one hour. A SPLIT block completes the loop by sending the flight transaction to the next block and the copy back to DEPL0.

The flight transaction proceeds to a TEST block to determine if an arriving or departing flight is represented. Departing flights proceed to program location GATE9, where the boarding process is simulated. Arriving flights transfer to DEPL5. At this location, the value 1, 2, or 3 is assigned to PB3 to indicate if the flight is a domestic, commuter or international type, respectively.

Byte parameter 3 is tested for a value of 1. Non-domestic flight transactions are transferred to DEPL1. Domestic flight transactions are assigned DDPL1F, the routing function for domestic passengers to PB1, and DBAGF, the bag distribution function for domestic passengers, to PB14. These transactions are then transferred to DEPL3.

At DEPL1, PB3 is tested for a value of 2. Commuter flight transactions proceed to the next block where the commuter passenger routing function, DCPL1F, is assigned to PB1. The commuter bag distribution function, CBAGF, is assigned to PB14 and the transaction transfers to DEPL3.

A-1-41
International flight transactions are transferred from DEPL1 to DEPL2. At DEPL2, the international deplaning passenger routing function DIPLF, is assigned to PBl. Byte parameter 14 is assigned IBAGF, the international passenger bag distribution function. This type of transaction proceeds to the next block at location DEPL3.

At DEPL3, the halfword savevalue SEQ1H, used for assigning sequence numbers to transactions, is tested for a value equalling or exceeding 32,000. Under these conditions, SEQ1H is reset to zero, then the transaction proceeds to the first of three consecutive SPLIT blocks. Otherwise, the transaction is transferred directly to the first SPLIT block.

Three program locations, DEP25, DEP15 and DEP17 are the destinations of the copy transactions from the SPLIT blocks. At DEP25, the transaction is initiated which will represent greeters using both the parking facility and the curbside after meeting passengers inside the terminal building. Greeter transactions to be created for merging with terminating passenger transactions after meeting inside the terminal building and departing the airport directly from the parking facility are started at DEP15. Program location DEP17 is the starting point for transactions to be generated representing greeters meeting passengers at curbside.

The flight transaction proceeds to DEPL7 where the creation of deplaning passenger transactions is initiated. A percentage of these passenger transactions will be matched
later with greeter transactions at locations specified by
the logic of both the following sections and the sections
creating deplaning passengers.

The first copy transaction proceeds to DEP25 where PB6
is given a value of 1 to indicate that the curbside will be
used after parking. The number of greeters in this category
for this arriving flight is determined by the variable FLT2V.
This number is assigned to PH8. The transaction transfers to
location DEP16 to Test PH8 for the existence of this greeter
type on this flight. If PH8 is zero, this copy transaction
is terminated. For non-zero values of PH8, the transaction
bypasses the TERMINATE block and adds the number in PH8 to
MH1 (PH1,5).

The random number table in ML1 is used as the argument
of functions which will assign the numbers of passengers per
party and bags per party to be transacted. The MH1 row number,
PH1, is used to assign the starting column of ML1. Variable
DPL3V is PH1, modulo125 and is assigned to PB10 to designate
starting the ML1 column number for this flight. At DEP 12, PB10
is incremented by one, then tested for a value in excess of 124.
When PB10 is greater than 124, it is reset to 1.

The value of the function PPPDF; the number of
passengers per deplaning party, is subtracted from the total
on the flight. PB10 is increased by 1, then the number of
bags drawn from the function contained in PB14 by using ML1
(1,PB10) as the argument value, is assigned to PB4. The
pointer PB10 is again incremented by 1 and then the random number
drawn from ML1(1, PB10) is tested for a value less than or
equal to GRGXL, the percentage of passengers meeting greeters
at the gate. When this condition occurs, the transaction
is marked for gate meeting by assigning the function GREGF to
PB1 and the value 1 to PB12. This transaction type is then
transferred to DEP14.

Greeter transactions not meeting at the gate are
transferred to a TEST block to determine if the passenger to
be greeted will have checked bags. The number of bags is
contained in PB4 and is tested for zero.

For the zero value, the function GRELF is assigned
to PB1. This function routes the greeter to the enplaning
level for meeting the deplaning passenger in the ticket lobby.

Greeter transactions with a non-zero value in PB4 are
assigned the function GREBF. This directs all remaining greeters
proceeding inside the terminal building to the bag claim
area for meeting.

Halfword parameter 8 is tested for a value greater
than zero to determine if another greeter transaction is required.
When PH8 is greater than zero, a copy transaction is sent back
to DEP12 by a SPLIT block to become the next greeter transaction.
The parent transaction from the SPLIT block is assigned the
parking lot number in PB14 and the number of greeters in PB5
from the distribution function GRPPF. The sequence counter,
SEO1H, is incremented by 1 and the current value is assigned
to PH10.
The transaction is held at an ADVANCE block for a time duration drawn from the function DCA2F, the distribution of arrival times at the landside for greeters. The times of greeter arrival are between one hour prior to flight arrival and 10 minutes after. After departure from this block, the transaction proceeds to CTRLO to begin simulation activity with routines prescribed by the function assigned to PB1.

The second copy transaction proceeds to DEP15. Byte parameter 6 is assigned a value of 2. This indicates that only this parking facility will be used by the greeter vehicle. After greeter and terminating passengers are matched inside the terminal building the group departs the landside directly after leaving the parking facility. The numbers of greeters from this category is calculated from the variable FLT3V and assigned to PH8. At the next block, DEP16, the test is performed for the existence of this greeter type in PH8. All further processing of this transaction is identical to that of the first copy transaction.

The third copy is routed to DEP17. The value 1 is assigned to PB6 to indicate curb usage. This transaction will represent greeters meeting their assigned passengers at this curbside. The number of passengers of this category is determined by the variable FLT4V and is assigned to PH8. A test is made to determine if PH8 is zero. The absence of this greeter type terminates this copy. When greeters of this type
are to be simulated, the number of these is added to the total of terminating passengers with greeters. This number is contained in MH1 (PH1,5). The starting column number of the random number table in MLI is placed in PB10 as before. Halfword parameter 4 is given the program location DCARO as the first and only destination for this transaction.

The table pointer is incremented as before and the party size is drawn at random from PPPDF and subtracted from PH8. The number of bags is also determined and assigned to PB4. A test determines if PH8 is greater than zero. When true, indicating more greeter transactions are required, a copy is sent back to DEP13 to represent the next greeter group. The parent transaction has zero assigned to PB14, GRPPF to PB5 and a sequence number, SEQ14, assigned to PH10. The transaction transfers to CTRL1 which will affect the transfer to DCARO. The greeter arrival time function DCA2F, is located at DCARO.

The original flight transaction is transferred to DEPL7 and will be used to generate all deplaning passengers for the flight. At DEPL7 the transaction is held until flight arrival time by an ADVANCE block, which delays further processing for 3600 seconds. A GATE block holds back succeeding aircraft until this baggage unloading logic, following location BUNLO, resets logic switch DPL1G. This allows deplaning passenger transactions for this flight to execute the assembler sub-routine BAGS without interference from a simultaneously arriving flight. After the GATE block is passed, DPL1G is placed in a
set condition by this transaction. The gate number of the flight in MH1 (PH1,9) is assigned to PH5 and also to PH7. The point number of the gate is assigned to PH2.

The maximum passenger unloading time ACUNL is calculated by multiplying the total number of passengers by 3 seconds and adding 90 seconds. A SELECTLR block places the number of the first chain between the limits CHA1B and CHA2B in a reset condition into PH6. Numerical values of these limits were previously set by EQU statements. A LOGICS statement places this chain in a set condition. The chain represents a bag claim device which will hold the passenger transactions in the bag claim logic until all bags are delivered from the flight.

Three split blocks route copy transactions to DEPL4, DEPL6 and DEPL10 to initialize the generation of transactions representing terminating, transit and transfer out of system passengers, respectively. The parent transaction proceeds to a TEST block to determine if a flight is available in the MH5 (1,1) element of the transfer flight table for accepting transfer passengers. If no flights are available, the number of transfer passengers on the flight represented by this transaction is added to the halfword save value XFRXH and this transaction to represent transfer passengers is terminated.

When MH5(1,1) indicates a flight is available, the transaction proceeds to DEPL8 where MH1(PH1,11) is tested to
determine if the arriving flight has any transfer passengers. If none, the flight transaction is terminated. When transfer passengers are to be simulated for this flight, PB8 is assigned a value 2, as an indicator of a transfer passenger and byte parameters 5 and 13 are assigned the value 1. This assumes all transfer passengers are travelling singly, rather than in a group.

A split block creates the number of transactions required to simulate transfer passengers on the flight. This number is obtained from MH1 (PH1,11). These transactions are tested to determine if the arrival flight is international. When this occurs, the international transfer passenger routing function, TIPlF, is assigned to PBl. A HELPA call to FORTM is used to assign the next flight by random number draw. Transactions not representing international arriving flights are routed directly to this HELPA block. The departing flight MH1 row number is assigned to PH1 at this time. The flight table matrix element MH1 (PH1,7) is tested for a value 3 to determine if the flight is international. When this occurs, the transaction is given a routing function TDP5F. This transaction is then transferred to DEPl1.

The non-international transfer passenger is transferred to a SAVE VALUE block, where the gate of the departing flight is retained as the save value SAVXH. The time difference between the departing flight time and the current clock time is tested to determine if the duration exceeds 45 minutes.
Those time differences exceeding this value are categorized as long term waiting times and the transaction is transferred six blocks to the next test. Transactions with a short stay, less than 45 minutes, test the value of the MH9 matrix elements in column 4 for the gate numbers of the arrival and departure gates to determine if both are on the same concourse. When this occurs, the selection function TDSSF is assigned to PBl and the transaction proceeds to DEPl. Transfer passenger transactions with a short stay and with arrival and departure rates on different concourses are assigned selector function TDSDF and transfer to DEPl.

The passengers with a greater duration than 45 minutes between current absolute clock and departure times are assigned selection functions TDLSF and TDLDF for the same or different concourses, respectively. The selector functions assign routine functions to PBl based upon input percentages and the current value of a random number generator. All transactions proceed to DEPl which executes a transfer based upon the address assigned to PH8 by FORTM.

The transit passenger transactions, represent continuing passengers who exit an aircraft, circulate within the terminal building and return to the same flight. These are marked as transit passengers by assigning 3 to PB8 and designated as single passengers with no visitors, by assigning 1 to PB5 and PB13.
Transit passenger transactions are created by a SPLIT block using MHl(PHl,13) to specify the numbers of copies. The parent transaction is terminated after the SPLIT block. An ADVANCE block simulates the deplaning process by delaying the entry of each transaction by an amount ACU2V, the random variable specifying service times. A HELPA block calls FORTM to assign the MHl row number of the next flight at the same gate. A process function is selected by assigning TDPXF to PB1. The function chosen by this selection function routes the transit passenger through selected landside facilities. If a process function cannot be selected, PB1 equals zero and the transaction is terminated. When PB1 is non-zero, the function executes the assigned routing function and proceeds to CTRLO to begin landside processing.

Transfers out of the system are routed to location DEP10. The number of transactions to be generated is specified by MHl(PHl,16). After departing the aircraft, these are assigned the routing function TOSDF, and transferred to CTRLO to begin processing.

Terminating passengers are routed to DEPL4 where PB8 is assigned the value 1 to designate the transaction as representing the deplaning terminating passenger.

Transaction sequence counter SEQ2H is tested for a value equal to or greater than 32,000. When this condition exists, the value is reset to zero. Otherwise, the transaction is
directed to a series of SPLIT blocks. Copies are directed to DEP18, DEP19, DEP22 and DEP26 to generate passenger groups using private vehicles and to assign sequence numbers for passenger greeter matching. A copy is also directed to a section generating transactions for simulation of passengers using public modes of transportation. The flight transaction is transferred to BUNLO to initiate simulation of the baggage unloading process.

The first copy is directed to DEP18 where PB6 is assigned the value 1 to indicate that the curbside will be used after the greeter-passenger meeting and removal of the vehicle from the parking garage is simulated. The number of transactions from this category is calculated by FLT2V and assigned to PH8. The procedure for determining party size is identical to that previously described for greeters. The random number selections are drawn from the table ML1, using the PH1 row number, modulus 125, as the pointer.

Using the same sequence of numbers for the passenger transaction as for the greeter, a transaction is created for each passenger group from the distribution function PPDF. The number of passengers in the group is assigned to PB13 and subtracted from PH8 as before. This is repeated until PH8 becomes zero or negative. When negative, the last party size is adjusted to force a zero value in PH8.

Subroutine BAGS is called by the passenger transaction and uses the argument FN*PB14 to determine the number of bags
to be assigned to the passenger group transaction in PB4. This is identical to the usage of the bag function performed by the greeter transaction and the same random number from ML1 is used by both transactions. For each terminating passenger transaction, subroutine BAGS also selects one random number for each bag specified in PB4 using one of the GPSS random number generators. The largest random number generated for the transaction is assigned to PH3. This value will be used later in the baggage unloading logic to release the passenger from the chain representing the bag claim device.

Sequence numbers, SEQ1H and SEQ2H assigned to PH10 in greeter and passenger transactions, respectively, are identical for the pair to be matched later. The numbers of greeter and passenger transactions proceeding to the gate, bag claim or lobby for meeting are determined identically. The major differences in the two paired transactions are; (1) assignment of different processing functions to PB1, (2) assignment of passenger group size number to PB13 of the passenger transaction and zero to the same parameter of the greeter transaction, (3) assignment of the greeter group size to PB5 of the greeter transaction, and assignment of passenger group size to PB5 of the passenger transaction.

The second copy is directed to DEP19 where PB6 is assigned the value 2 and the number of transactions needed to proceed directly out of the airport from the parking facility after meeting is calculated from VFLT3V and assigned to PH8. The
same processing as the transaction processing to DEP18 is executed.

The third copy proceeds to DEP22 where the transactions representing terminating passenger groups greeted at the curb are generated. A value of 1 is assigned to PB6. The number of transactions required is determined from FLT4V and assigned to PH8.

All of the transactions generated from the three above copies are transferred to DEP24 where the deplaning and gate meeting processes are simulated.

The fourth copy, routed to DEP26, assigns a value of 2 to PB6 and calculates the number of transactions required to represent passengers without greeters using private auto. The numbers of passengers per party is drawn from the function PPPDR, using the random number table in ML1 as a source for argument values of the function. Subroutine BAGS is also executed with a number from ML1 as the argument of the bag per passenger function in PB14. If the number of bags is zero, the function DLPL1F is assigned to the transaction representing this passenger group. These transactions proceed directly to DEP24 for deplanement.

The final SPLIT block directs a copy transaction to areas creating transactions representing terminating passengers using taxi, limousine or bus for ground transportation. The number of passengers on the flight utilizing ground transpor-
tation modes other than private auto is provided by FLT6V and is assigned to PH8. The function PPPDF is also used to determine party size and is assigned to PB13. A loop determining PH8 by each party size until depletion, creates the required number of transactions. Each transaction executes a HELPA block and is assigned a mode by FORTM using random number selection. The subroutine BAGS is also executed. For passengers other than those using car rental, the number of bags, PB4 is tested for zero. When this condition occurs, the function DLPLF is assigned to PB1. All BAGS transactions are transferred to DEP 24.

At DEP24, all terminating deplaning passenger transactions are delayed by the value ACU2V a random unloading time from a uniform distribution. Byte parameter 12 is tested for a value of 1, to determine if the passenger transaction will attempt to match the corresponding greeter transaction. Those passenger transactions with PB12 not equal to 1 are transferred to CTRL0 to begin landside routine. Those with a value 1 in PB12 are transferred to an UNLINK block to examine this user chain GREGC. The parameter PH10 of the passenger transaction is compared to the same parameter of all the greeter transactions waiting on the chain. When transactions match, the greeter transaction is removed from the chain and transferred to CTRL1. The passenger transaction proceeds to the next block at DEP29. At this location, the
transaction is held until the logic switch PAS3L is placed in a set condition by the greeter transaction. This setting is performed in the gate logic section by the greeter transaction after transfer from CTRL1. The number of greeters is assigned to the savevalue PAS32 and the parking lot number to PAS33. These are obtained from PB5 and PB14 of the greeter transaction respectively in the gate logic section. After PAS3L is set, the number of greeters, PAS32, is added to PB5 of the passenger transaction. The parking lot number, PAS33, is placed in a reset condition for use by the next passenger transaction finding a matching greeter. The active transaction proceeds to CTRLO to continue landside routing.

When the terminating passenger transaction marked for gate meeting does not unlink a greeter, the transaction is transferred to DEP28. An ASSIGN block places the program location DEP29 into PH4 and transaction is linked to GREGC. This will transfer the passenger transaction to this location when unlinked later by the greeter transaction in the gate logic.
Enplaning Passenger Logic

The first block in this section is at location ENPLO. Because this is the B operand of the JOBTAPE statement in the program definition statement section, all transactions created by the Auxiliary Program and written on the JOBTAPE are routed to this block. The first transaction does not represent an enplaning passenger group, but was written on the JOBTAPE to provide an initial time coincident with the start of the simulation to reference the entry times of all originating enplaning passenger transactions. This first dummy transaction is transferred to ENPL9, where a CHANGE block referencing CHNGO redefines the operation performed by the block at ENPLO. All succeeding transactions will transfer to program locations in the enplaning passenger logic based upon the selection function ENPIF. The dummy transaction is then terminated.

Function ENPIF will route the JOBTAPE transactions to ENPL1, ENPL2 or ENPL3 based upon respective values of 1, 2 or 3 in PB3. The parameter was assigned these values in the Auxiliary Program to represent domestic, commuter or international passengers, respectively.

Transactions representing transfers out of the system are arbitrarily assigned a value 1 as the point number designator in PH2 and the routing function TOSEF is assigned to PB1. This transaction is transferred to CTRLO to begin
landsdie processing.

The originating enplaning domestic passenger transaction is assigned the function EDPIF in PBl and a random number from the variable RND2U in PBl0. The latter serves as an argument to DBAGF, the function used in the next block to assign the number of bags to PB4. This transaction then transfers to CTRLO.

Enplaning commuter passenger transactions are assigned their processing function, ECP1F, to PBl at location ENPL2. The number of bags assigned to PB4 from function CBAGF and the transaction is transferred to CTRLO.

International enplaning passengers transfer to ENPL3 from ENPLO. The routing function assigned to PBl is EIP1F and PB4 obtains the number of bags from the function IBAGF.
Facility Modules Section

This section describes the program logic of 17 modules used to simulate landside processing facilities. These operate independently, with few direct linkages from one module to another. Transactions are generally routed from one of these modules to another through the use of the routing function and a transfer to CTRLO of the Control Module or the assignment of the address of another module to PH4 and a transfer to CTRLI, also in the control module.

Two modules, Baggage Unloading Logic and Deplaning Curb (Cars) simulate landside activities not performed by passengers. All others involve passenger simulation.

1. Bag Claim

This module begins at location BAGCO, when a test is performed on PB4 of the entering transaction to determine if the number of simulated bags associated with this transaction is zero. If PB4 is zero, the transaction returns immediately to CTRLO. All other transactions proceed to a HELPA block and return from FORTM with the point number of the bag claim facility assigned to PH2; the value 4 to PB11 and the landside facility number to PH7. The transaction is then advanced by TRUXH, the walking time from the last facility.

Byte parameter 13 is tested for a zero value. When this occurs, indicating that a greeter transaction is being processed, the transaction attempts to unlink the corresponding
passenger transaction from the chain GREBC by matching PH10 of both transactions. If successful, the greeter transaction proceeds to a GATE LR block at BAG 3. Since logic switch PAS4L is normally in a reset condition, the greeter transaction continues and assigns the number of greeters in PB5 to PAS42 and the parking lot number PAS43. Logic switch PAS4L is set to allow the passenger transaction to obtain these two values. If the vehicle used by the greeter was assigned to proceed to the curb, then the greeter transaction is routed to GRTY3 to simulate removal from the parking facility. Otherwise, the greeter transaction is terminated. A greeter transaction unable to unlink a passenger transaction is transferred to BAGC2 where it places BAGC3 in P-4 and then links this transaction on the user chain GREBC.

Passenger transactions are routed to BAGC1. A savevalue recording the occupancy at the point where the bagclaim facility is located is incremented by PB5, the number of passengers in the party. Matrix element MH1(PH1,1) is tested for a value of zero, to determine if all bags have been delivered to the bagclaim for the flight. When this occurs, no time for bag delivery or pickup is simulated. The passenger transaction parameter byte 12 is tested for a value 2, to determine if the simulated greeter and passenger matching process will occur at bagclaim. For PGL2 not equal to 2, the passenger transaction proceeds to BAGC4 for a transfer to CTRLO.
When PB12 is 2, the passenger transaction attempts to unlink a matching greeter transaction from chain GREBC. If successful, the passenger transaction waits for the unlinked greeter transaction to set PAS4L. The values PAS42 and PAS43 are subsequently assigned to PB5 and PB14, respectively, and logic switch PAS4L is reset. The passenger transaction transfers to BAGC4 for immediate transfer to CTRLO.

Passenger transactions unsuccessful at unlinking greeter transactions are sent to BAGC5, and BAGC6 is assigned to PH4 and the passenger transaction is placed on the chain GREDC.

When bag delivery from the aircraft is incomplete, the address BAGC7 is placed in PH4 and the passenger transaction is placed on the chain assigned to PH6 by the deplaning passengers logic. The unlinking from this chain occurs in the Baggage Unloading Logic Module. Upon return from BAGC7, after unlinking from the PH6 baggage chain, the test for greeter matching and subsequent activity described above is performed.

2. Baggage Unloading Logic

This module is accessed by arriving flight transactions generated by the Deplaning Passenger Logic. The first block of this module, at BUNCO, changes the priority of the entering flight transaction from 10 to 5, and, through use of the BUFFER operand, places it below the passenger transactions on the current events chain and restarts the chain. This operation allows the passenger transaction to execute the
assemble subroutine BAGS before the baggage unloading logic operates.

The number of halfword and byte parameters of the flight transaction are changed to 9 and 40 respectively. A HELPA block is enacted to assign MH7 row numbers to transaction parameter bytes. Each element of MH7 represents a count of simulated passenger bags placed in the array by subroutine BAGS. Subprogram FORTM extracts each element beginning with the lowest numbered and computes the cumulative sum of the elements. At the same time FORTM examines the cumulative sum after each element is added. Each time the sum exceeds an integral multiple of the C-operand, it places the MH7 row number in a parameter byte, starting with number 40 and decrementing to number 1, and then increases the C-operand multiplier by one.

An ADVANCE block simulates the aircraft unloading time based upon a random draw from a distribution. Bag delivery is simulated by unlinking all passenger transactions from the PH6 bag claim chain. This process is performed in a loop, with the flight transaction byte counter decremented from 40 to 1.

The PH3 value assigned by BAGS to the passenger transaction is compared to the MH7 row number in the byte parameter of the flight transaction. All transactions with PH3 less than or equal to the value in the byte parameter are unlinked. The flight transaction time is advanced 30 seconds between each
decrement, simulating a large wait for passenger transaction with high PB3 values than those with low. Passenger transactions are routed to BAGC7 by a transfer to CTRL1.

The PH6 chain is released by a LOGIC R block. Matrix savevalue MH1 (PH1,1) is assigned a value 2, indicating that the delivery of all bags is complete. The flight transaction is then terminated.

3. Ticketing and Check-In

This module is entered by enplaning and some deplaning and transfer passenger transactions. Greeter transactions, routed to simulate the meeting of greeters and deplaning passengers without baggage, also utilize this program section.

At CHEKO, the first program location of this module, the Boolean variable CHK1B is used to test transaction parameters PB7 and PB9 and the value produced by random number generation RN7. The test simultaneously determines if PB7 equals 1, indicating an enplaning passenger transaction; if PB9 is zero, indicating a preticketed status; and, if PB7 is less than the input percentage of preticketed passengers proceeding directly to the gate. When CHK1B is true, the transaction bypasses the ticketing and check-in procedure and is routed directly to CTRLO, for transfer to the Security Module.

Transactions requiring processing by the module execute a HELPA block. They are assigned the following parameter
values by FORTM: point number is PH2; program location CHEK2 or CHEK3 is PH4; the GPSS storage or queue number for the check-in facility of the MH1(PH1,3) airline is PH5; the landside facility number is MH9 in PH7; and the process code number 2 or 14 in PB11, to indicate express or full service check-in.

An ADVANCE block simulates the walking time from the previous facility. Byte parameter 13 is tested for a zero value. When PB13 is zero, indicating a greeter transaction, a set of logical operations simulating the greeting process and identical to those performed at Bagclaim, is executed.

Passenger transactions are transferred to CHECK9 and PB12 is tested for a value of 3, the flag indicating a deplaning passenger transaction to be greeted at the enplaning level lobby. These deplaning passenger transactions also simulate the greeting process using program logic identical to that in the Bagclaim Module.

Transactions with PB12 not equal to 3, transfer to CHEK6. At this location another test is performed, this time on PB8, to determine if the transaction represents a terminating passenger. This transaction type is transferred to CTRLO. All other transactions increment the occupancy PH2 savevalue by the value in PB5.

The passenger transaction requiring a simulation of the check-in process joins a queue at the PH5 facility and
the time of entry is marked. When the transaction is ready to enter the service process simulation, the waiting time deviation is placed in PH11 and the transaction transfers to CHEK2 for a preticketed passenger check-in, or to CHEK3 for full service. Service time for each facility is drawn from the applicable service time distribution. For the preticketed passenger transaction only one distribution is used. However, for simulation of full service ticketing, the selection function ATK1F will choose the service time from the distribution applicable to the airline number represented in MH1(PH1, 3). Halfword savevalue CHKXH is assigned the value of the airline number and is used as the argument for the function ATK1F. Upon completion of the simulated ticketing and check-in process, the occupancy count is decremented by PB5. The flow count in MH13 is incremented, and the transaction transfers to CTRLO.

4. Immigration

This module is entered by deplaning international passenger transactions. At IMMIO, a HELPA call to FORTM results in the following parameter assignments: point number of the facility is PH2; GPSS queue and storage number is PH5; landside facility number is PH7 and PH8; and the value 13 in PB11 to denote immigration processing.

The walking time from the previous facility, TPVX!, is simulated by an ADVANCE block. The PH2 occupancy count is incremented by PB5. The transaction enters a QUEUE block if
the storage is full or begins simulated service if it is not full. The simulated service time is obtained from the variable FMM1V. After leaving the PH5 storage, the transaction decrements the occupancy count by PB5 and increments the immigration flow count IMIG by the same value. The module is exited by a transfer to CTRL0.

5. Customs

This module is also entered by international deplaning passenger transactions. The HELPA block at CUSTO returns with the same type of information in PH2, PH5, and PH7 as in the immigration module. PBll is assigned a value 5 to signify customs processing.

All simulation processing is identical to that performed in the immigration module. The service time is provided by the variable CUSlV.

6. Concourse Exit

This module provides a program location for accumulating the count, in MHll, of simulated passengers and visitors exiting concourses and entering the terminal lobby. At CNCRO the HELPA block assigns the point number of the exit to PH2 and the FACNO numbers of the security station for this concourse to PH5. The walking time from the gate to this exit is simulated. After incrementing the matrix element MHll (PH1, PH5) by the value PB5, to increase the passenger-visitor concourse count, the transaction is transferred to CTRL0.
7. Deplaning Curb (Cars)

In this module, vehicles proceeding to the curb to greet terminating passengers are simulated. Assignments to curb-side or double parking are made by this module, based upon current space availability. A limited queue at each curbside section is established if storage representing curbside or double parking is filled. Congestion of vehicular traffic resulting from lane blockage due to double parking and queueing is simulated. The program logic inspects only the curbside section assigned to the bagclaim area of the terminating passenger for space availability. When no space is available the vehicle transaction simulates a recirculating process.

Vehicle transactions routed to DCARO represent greeters arriving at the airport for simulation of passenger-greeter meetings at the curbside. These are delayed by a random draw of the function DCAIF which represents the distribution of times of arrival of greeters at the airport landside. These vehicle transactions are joined at DCAR1 by those representing vehicles removed from the parking lot by the greeter transaction after simulation of their matching process within the terminal building. Vehicular counts on the entry road and deplaning curbside approach section are incremented by 1 at DCAR1 and the next block, respectively.
A HELP call to FORTM assigns a GPSS storage number to the transaction in PH6 and the value 1, 2, or 3 is assigned to PH10 to indicate curbside parking, double or queuing, respectively. If no available space is located, a simulated recirculation of the vehicle will be performed. PH6 is then assigned a value zero and PB10 is 4.

The byte parameter 4, as operand in the HELP block, determines if the vehicle will proceed to the enplaning or deplaning curb. When PB4 is zero, indicating that the passenger and greeter transactions are without simulated bags and the greeting process will take place at the enplaning level lobby, FORTM assigns a storage number to PH6 corresponding to enplaning curbside parking, double parking or queuing. When PB4 is non-zero, the assignment is made to a deplaning curb storage. Recirculation is provided for either PB4 condition when required.

Upon return to GPSS, a test on PB4 for a non-zero value is performed. Transactions with PB4 zero values are transferred to DCARO. Because the simulated vehicle may pass other curbside sections to arrive at the assigned section, the program must determine if lane blockage occurs at this section to be bypassed. The first deplaning curb storage number minus 1 is placed on PH8 and PH4 and the first deplaning double parking storage number minus one is PH5.
A variable DEPLS is established and assigned in PH8 for the purpose of comparison with PH6. This will determine if the GPSS storage number of the curbside parking, double parking or queue, contained within the curb section where the transaction is currently located, matches the destination storage of the transaction. Because the value of DEPLS depends upon PB10, comparison of PH8 with PH6 will only be between storage numbers representing identical facility types. Recirculating vehicle transactions do not calculate the value of DEPLS for assignment in PH8. These transactions transfer to DCARB and are joined by the transactions with DEPLS assigned to PH8.

Vehicle transactions transferred to DCARD are assigned a value of 1 less than numbers of the RPSS storages representing the curbside at the first enplaning curb section in PH4 and the double parking area at the first enplaning curbside section in PH5. The variable ENPLS performs the same function as DEPLS but operates on enplaning curb storage and is assigned to PH8. All transactions proceed to DCARB.

The loop for simulating lane blockage and delay is begun at DCARB. PH8 is incremented by 1 each time this block is passed. Non-recirculating vehicle transactions test PH8 against PH6 to determine if the current curbside facility is the destination. When PH8 equals PH6, the transaction proceeds to DCARA to simulate parking. Otherwise, PH4 and PH5 are each incremented by 1. A Boolean variable LNFLD determines if
all lanes are blocked in the current curb section, and holds the vehicles at the TEST block if this occurs. Otherwise, the PH5 storage is tested to determine if any vehicles are double parked. When no double parking occurs, the land delay, LNDLY, is zero and the transaction performs a test to determine if the current curb section transfers to DCARA for the last section or DLARB for passing the next curb section.

When lane blockage occurs, the maximum number of lanes available is decremented by one and assigned to NOLAN. The delay for each vehicle is calculated as 2x3/NOLAN and placed in the variable LNDLY. Each transaction passing the current curb section is delayed by an ADVANCE V8LNDLY block. The same test and routing to DLARA or DCARB is performed.

At DCARA, the count of vehicles as the deplaning curb is incremented by 1 and the entry time into this section is marked in PH8. Byte parameter 10 is tested for values 1, 2, or 3. The value 1 routes the transaction to DCAR5; 2 to DCAR4; and 3 enters the vehicle in the storage representing queuing at the destination curb section. Recirculating vehicle transactions have CIRCO assigned to PH4 and transfer to CTRL1.

Queuing vehicle transactions enter storage PH6, change their priority level to 12, then link on a chain numbered PH6. These will be released when corresponding greeter vehicle and passenger transactions are matched in the deplaning curb.
passenger logic. A transaction is released from DPLIC and routed to DCAR2, when the simulated matching takes place between the passenger and a vehicle at curbside. At the section following DCAR2, the transaction representing a previously double parked vehicle will move into the storage representing the curbside, reset its priority level to 10, relink itself on chain DPLIC and remove one transaction from the queue chain, and route the transaction to DCAR3. Vehicle transactions released from DPLIC by the deplaning curb passenger logic, representing double parked vehicles, release one transaction from the queue and transfer it to DCAR3.

At the program location DCAR3, PB7 is tested for a value zero. A non-zero value indicates that the transaction is a greeter vehicle using the enplaning curb and transfers it to ENPC8. Deplaning curb vehicle transactions continue to the next block and leave the PH6 storage entered above by queueing vehicle transaction. The number of the double parking storage associated with the queue previously departed is placed in PH6. Byte parameter 10 is assigned a value 2, signifying double parking. The transaction proceeds to the next block at DCAR4.

All double parking vehicle transactions proceed through DCAR4, where the priority level is set at 11. These transactions, and the transactions representing vehicles to be parked at curbside, enter the PH6 curb or double parking storage
An attempt to unlink waiting passengers from the chain DPL2C by matching equal values of PH10 in the vehicle and greeter transactions results, if successful, in a transfer to DPLC9. If unsuccessful, the vehicle transaction is linked on chain DPL10.

The model generates a transaction to perform a policing function. This transaction will remove vehicle transactions remaining in double parking or queuing longer than a prescribed time limit. The length of time allowed in these two facilities is specified in an ADVANCE block in DLAR6. After passing DCAR6, the transaction unlinks all transactions representing double parked vehicles on the chain DPL1C if the absolute clock time is greater than the entry time into these two facilities plus 300 seconds. These vehicle transactions are transferred to DCAR8.

The policing transaction places the total number of all curb sections, enplaning and deplaning, in PH1. The APSS storage number, DPQCS, of the first deplaning curb queue is placed in PH2. At DCAR7, the program unlinks the vehicles from the PH2 chain if they have been in the queue longer than 30 seconds. This chain has the same number as the storage representing a queue at a curb section. The program increments PH2 by one and loops back to DCAR7. Because the storage and chains representing queues at curb sections were numbered continuously for deplaning and enplaning curbs by SYN statements, executing this loop PH1 times will remove all queued
vehicle transactions and transfer them to DCAR9. The policing transaction is transferred back to DCAR6 to await the next enforcement time.

Double parked vehicle transactions forced to leave the chain DPLIC and transferred to DCAR8, unlink one vehicle transaction from the queue chain corresponding to the storage representing double parking. The unlinked transaction is routed to DCAR3. The previously double parked vehicle transaction proceeds to the next block DCAR9.

At DCAR9, previously queued and double parked vehicle transactions leave the PH6 storage. The priority level is dropped to 10. For vehicle transactions assigned to enplaning curb sections, the waiting time spent queued or at double parking is added to PH11. This addition is performed for this type of vehicle transaction because only queued vehicle transactions are forced to leave the enplaning curb. Double parked vehicle transactions remain at this curbside without a simulated enforcement process. Furthermore, the vehicle using the deplaning curb has a waiting time dependent upon time of arrival at the landside. Simulated waiting time for this transaction type is not entirely dependent upon landside service processes and is not entered into waiting-time tables.

Vehicle transactions leaving the deplaning curb area are assigned to program location CIRCO to simulate a recirculation process. Vehicles directed away from the enplaning curb area begin recirculation at CIRC1.
8. Deplaning Curb (Pax)

The terminating deplaning passenger transactions are routed to DPLCO for subsequent matching with greeter vehicle transactions, loading into taxis or loading into buses or limos. A HELPA call to FORTM by the transaction returns with the point number of the deplaning curbside section in PH2, the landside facility number in PH7, and the value 12 in PBI1. The transaction is advanced by TRUXH, the walking time from the last facility utilized. Using the selection function DPL1F, a branch to DPLC3, DPLC4 or DPLC5 is executed for transactions assigned to private vehicle, taxi, or bus/limo modes, respectively.

Transactions transferred to DPLC3 attempt to unlink waiting vehicle transactions from the chain DPL1C, even if previous matching between greeter and passenger transactions has taken place within the terminal building. This is because simulated vehicles removed from the parking facility after the terminal building greeting process and subsequent routing to the curbside, perform the same logical operations as greeter vehicle transactions proceeding directly to curbside from the airport boundary. The unlinked vehicle transaction is transferred to DPLC9 and the passenger transaction transfers to DPLC1. Unsuccessful passenger transactions are transferred to a second UNLINK block to attempt unlinking transactions representing greeters before vehicles have recirculated and parked; these are located on chain DPL3C.
Greeter transactions are transferred to DPCG2 and passenger transactions go to DPCG1. Passenger transactions unsuccessful in removing transactions from DPL3C are transferred to DPLC2.

The passenger transactions unlinking a greeter vehicle from DPL1C are held by a GATE block at DPLC1 until the vehicle transaction places the storage number of the curb section half-word savevalue PASS1, the PB10 flag in byte savevalue PASS2 and sets the logic switch PASSL. The passenger then proceeds and places PASS1 in PH6 and PASS2 in PB10. The passenger transaction resets PASSL and simulates the vehicle loading time by advancing the value DPL1V. Byte parameter 10 is tested for a value 1 to determine if the greeter vehicle will leave from the curbside. When this occurs, the storage number of the curbside is placed in TMPXF for calculating a random pull-out time using variable DPL2V. This time value is retained in PH11 for entry in the table of waiting times. The passenger transaction leaves storage PH6 and unlinks one vehicle transaction from DPL1C when the double parking storage adjacent to the curbside area just departed, matches PH6. The unlinked vehicle transaction is routed to DCAR2. The passenger transaction transfers to DPLC8.

Greeter transactions with PB10 not equal to 1, transfer to DPLC7 and leave the PH6 double parking storage. They unlink one vehicle transaction from the queue chain for vehicles awaiting entry to the double parking storage just departed.
These greeter transactions proceed to DPLC8 where they join the passenger transactions from the curbside and increment the count of vehicles on the departing road. All transactions transfer to CTRLO.

At DPLC2, those passenger transactions unsuccessful at unlinking greeter transactions from DPLC1 and DPL3C are assigned the address DPLC1 in PH4 and are linked on the chain DPL2C to await unlinking by which transactions in the deplaning curb (cars) logic.

Greeter vehicle transactions unlinked from DPL1C are transferred to DPLC9. These are held at a GATE block if PASSL has not been placed in a reset condition by a passenger transaction. When able to proceed, the vehicle transaction places the value of PH6 in PASS1 and PB10 in PASS2 to provide information to the matching passenger transaction. The transaction sets logic switch PASSL, to allow the passenger transaction at DPLC1 to proceed and is then terminated.

Transactions using the bus/limo or taxi are transferred to blocks DPLC4 or DPLC5, respectively by the function DPLC1F. At DPLC4, the count of deplaning passengers waiting for a bus or limo, DPCXH, is incremented by 1. At DPLC5, the loading time of the taxi, PDL1V, is simulated by an ADVANCE block. The taxi increments the count of vehicles on the roadway departing the airport by 1. All transactions entering these two sections are transferred to CTRLO.

A transaction representing a bus or limo is generated in the Control Section and transferred to DPLC6. At this and
subsequent blocks, the vehicle transaction increments the count of vehicles on both the arriving and departing roadways. It also removes the passengers waiting for a bus or limo by assigning the halfword savevalue DPCXH a value of zero. This transaction is then terminated.

The last three areas of the Deplaning Curb Section (Pax) simulate activities of greeters intending to meet passenger at curbside but forced to recirculate and park because of congestion at the curb. At location DPCGO the greeter transaction executes a HELPA block. The E-operand is used as a flag to indicate that this transaction represents a greeter acting as a pedestrian and has a value 1.

The transaction assigns the point number of the curb to PH2. The landside facility number of the curbside is placed in PH7 and the value 12 is assigned to PB11. The transaction is advanced by TRVXH, the walking time from garage to curb. The greeter transaction attempts to unlink a passenger transaction from the chain DPL2C. If successful, the passenger transaction transfers to DPLG1 and the greeter transfers to DPCG2. Unsuccessful greeter transactions are placed on chain DPL3C for subsequent unlinking by passenger transactions.

At DPCG1, the passenger transactions are held by a GATE block until switch PAS5L is placed in a set condition by the greeter transaction. The greeter transaction passes through the GATE block at DPCG2; then places the number of greeters in byte savevalue PAS52 and the parking lot number in PAS53. The
greeter transaction places PAS5L in a set condition and is then terminated.

After the passenger transaction proceeds past the GATE block, PAS52 is added to PB5 and PAS53 is assigned PB14. The passenger transaction resets PAS5L. A value 2 is assigned to PB6 to indicate a parking mode. The process function pointer, PB2, is reset to 1 and routing function GRCPF is assigned to PB1. This routes the passenger transaction to the parking facility. The transaction transfers to CTRLO.
9. Enplaning Curb

This program section represents the activities of the originating passenger from the vehicular approach to the enplaning curbside, through the parking and unloading process to entry into the terminal building. Private vehicles and taxis departing the curbside after separation from the passenger groups are simulated. Arrivals of buses or limos at the enplaning curb are also represented. This module contains a section to simulate recirculation of vehicles from either enplaning or deplaning curbsides.

Transactions representing originating passengers are routed to ENPCO if the simulated mode of arrival at the airport is private car or taxi, or to ENPC2 if the mode is bus or limo. At ENPCO and the next block, the count of vehicles, ARDXH, on the airport entrance road and the count of vehicles, ENPXH, proceeding to the enplaning curbside, are both incremented by 1. A HELP call to FORTM is executed, using the airline number and the PB6 mode operates.

For the private car or taxi mode, subprogram FORTM searches for a curbside or double parking space. The curbside associated with the airline specified by the B operand is examined first, and, if necessary, all other curbside sections are inspected in a prespecified order with adjacent sections first and remote sections last. If a space is located, the space number is assigned to PH6. The value 1 or 2 (from P91) to designate curbside or double parking, respectively, is available at the two above.
facility types, the queue associated with the B-operand air-
line curb section is inspected for a space. An available
space results in the assignment of the storage number repre-
senting the queue in PH6 and 3 in PB10.

Transactions representing vehicles parked at curbside
or in a double parking location or in the queue are assigned
the point number of the facility to PH2 and the landside
facility number to PH7. Vehicle transactions unable to locate
any space are forced to recirculate. These are assigned a
value zero to PH5 and PH6 and a value 4 is assigned to
PB10.

After returning from FORTM, the model performs logical
operations to model congestion due to lane blockage. The
operations performed are identical to those in the deplaning
curb logic with two exceptions. First, the number of bags
in PB4 is not inspected because only the enplaning curb
congestion is modeled by this section. Secondly, the time
spent passing through the congestion is recorded in PH11 for
enplaning curb vehicle transactions but not those using the
deplaning curb.

When the vehicle transaction is at the section containing
the storage designated by PH6, the transaction is transferred to
ENPC1. Recirculating vehicle transactions completing simulated
passage of all curbside sections are also directed to ENPC1.

The subsequent logical operations of this curbside section
closely parallel the Deplaning Curb (Cars) Section. Transactions
representing double parked vehicles are routed to ENPC4, where their priority level is made 11, as before, and directly enter the PH6 storage curbside vehicle transaction. Queuing vehicles enter their PH6 storage, change their priority level to 12, as before, and link on the PH6 chain. Recirculating vehicle transactions are transferred to CIRC1.

Vehicles transactions in curbside or double parking storage simulate the unloading time by an ADVANCE block which holds the transaction by the amount ENP1V. A SPLIT block separates the vehicle transaction from the originating passenger transactions. The passenger transaction is tested for a preticketed status. If PB9 is zero, indicating this condition, a percentage of these transactions, defined by halfword savevalue CRBXH, enter an ADVANCE block to simulate curbside check-in. All transactions, including those representing non-preticketed and non-curbside check-in passengers are transferred to CTRLO.

The vehicle transactions are transferred to an ADVANCE block to delay the transaction for simulating empty-car parking time. The vehicle transaction with PB10 equal to 1, simulates a random pull-out time from curbside and then leaves the PH6 storage. This transaction unlinks a vehicle transaction from the queue corresponding to the departed curbside, and transfers it to ENPC8. The parked vehicle transaction is transferred to ENPC9.
Double parked vehicle transactions transfer to ENPC7 and immediately leave the PH6 storage. These also unlink one vehicle transaction from the queue adjacent to the double parking storage, transfer it to ENPC8 and proceed to ENCP9.

At ENPC9, the count of vehicles on the departing road is incremented by 1. If no well-wishers are simulated to enter the terminal building, PB5 equals PB13 and the vehicle transaction is terminated. When PB5 and PB13 are unequal, the vehicle transaction executes a HELPA call to FORTM to determine the parking facility number to be entered. It then increments the entry count of the parking facility and is terminated.

Queued vehicle transactions released from the PH6 chain by previously parked vehicle transactions and routed to ENPC8, test PB7 for a value of 1 to determine if they represent vehicles using the deplaning curbside. Deplaning curb vehicle transactions are transferred to DCAR3. The enplaning curbside vehicle transactions proceed to the next block and add the waiting time spent in queueing to PH11. These transactions leave the PH6 storage which represented the queue, and assign the storage number of the curbside in the same curb section to PH6 and assign one to PB10. The PH6 curbside storage is tested for occupancy and, if not full, the vehicle transaction enters it at ENPC5. When the curbside storage is full, the corresponding double parking storage number is assigned to PH6.
Byte parameter 10 is assigned the value 2, the priority level of the transaction is changed to 11 and the transaction enters the PH6 storage at ENPC5. The vehicle transaction is then terminated. Enplaning passenger transactions assigned to the bus/limo mode are routed to ENPC2. A HELPA call to FORTM returns with the loadside facility number of the bus stop for the airline designated by the B-operand placed in PH6 and the point number of the facility is PH2. The transaction has the program address ENPC3 assigned to PH4 and is linked on the chain EBUSC to await arrival of the simulated bus/limo at the enplaning curbside. After unlinking by the bus/limo transaction, the enplaning passenger transaction is transferred to CTRLO.

The transaction generated by the Control Section representing the bus/limo is routed to ENPC6. At this location the transaction increments the halfword savevalue ARDXH, the count of vehicles on the approach roadway. An ADVANCE block simulates the unloading time of the passengers from the vehicle. This transaction unlinks all passenger transactions placed on EBUSC by the instructions following ENPC2 and transfers them to CTRL1. The count of vehicles on the departing road, DRDXH, is incremented by 1 and the bus/limo transaction is terminated.

The recirculation roadway logic is also contained in this section. Vehicle transactions assigned to the Deplaning Curb (Cars) logic are transferred to CIRCO for recirculation.
An ADVANCE block simulates the vehicle recirculation time and the count of vehicles proceeding from the deplaning curb to the recirculation roadway, RCDXH, is incremented by 1. A test of PBL2 for zero, determines if the transaction was routed to perform the matching process with the terminating passenger transaction only at the curbside. A percentage of these transactions specified by halfword savevalue CPKXH are selected to enter the parking facility. The PB6 mode is assigned a value 2 and the routing function GRECF is assigned to this transaction. The transaction is then transferred to CTRL0.

All other deplaning curb recirculating transactions, those with PBL2 non-zero and those not parking, are routed to DCAR1 to repeat the deplaning curb logic instructions. Recirculating vehicle transactions transferred from the enplaning curb logic to CIRCl are advanced by the value CIRCV to simulate recirculation time. The count of these transactions, RCEXH, is incremented by 1 and the transactions then transferred to ENPCR to return to the enplaning road logic.
10. Entrance

Originating enplaning transactions and greeter transactions with non-zero PB12 values are routed to ENTRO to simulate entry into the terminal building. A HELPA call to FORTM returns with the point number in P42, of the entrance nearest to the facility immediately departed by the transaction. The transaction advances TRVXH, the amount of walking time from the last point to the entrance. The Boolean variable DPDIN is tested for a value of 1 to determine if the transaction will increment the terminal entry count on the deplaning level. When this occurs, the transaction represents a greeter proceeding to bag claim. The halfword savevalue DPLIN is incremented by the value in PB5. The transaction is transferred to increment the total count of all entrances in halfword savevalue ENDUR and is then transferred to CTRLO.

All other transactions are assumed to enter on the enplaning level. These increment the halfword savevalue ENDIN by the amount of PB5 and also increment ENDOR. These transactions also transfer to CTRLO.
11. Exit

This module begins at program location EXITO. A HELP call to FORTM at this location returns with the point number in PH2, to the exit nearest the facility immediately departed from. The walking time from the last facility to the exit is simulated by an ADVANCE block.

The Boolean variable DDOUT is used to determine if this transaction represents a domestic, international or commuter passenger group with baggage. These transactions increment the halfword savevalue DPOUT, the count of passengers and visitors exiting the terminal from the deplaning level, by the value in PB5. This transaction type also increments EXDOR by the value in PB5.

All other transactions are assumed to represent passengers or visitors exiting through the enplaning level. These increment the count EPOUT, for this level, and increment the total exit count, EXDOR, both by the value in PB5. All transactions entering this section transfer to CTRLO.
12. Gate (Enplaning pax)

This module simulates the processing of enplaning passengers at gate counters, matching of greeters with passengers and separation of well-wishers from passengers. A subsection simulates the aircraft boarding process.

Passenger and greeter transactions first enter this section at GATE0. At this location, a HELPA block is executed. This results in assignment of the point number of the gate facility to PH2 and the GPSS storage number of the gate to PH5. The landside facility number, identical to the gate number, is assigned to PH7 and PB11 is given a value of 1.

The walking time from the last facility utilized is simulated by an ADVANCE block. Byte parameter 13 is tested for a value zero, to determine if the transaction represents a greeter. Non-greeters are transferred to GATE7. Greeter transactions attempt to unlink a passenger transaction from the chain GREGC. The transaction removed is routed to DEP29 in the Deplaning Passenger Logic Section. The greeter transaction proceeds to GATE8, and, after passing a GATE block, assigns the number of greeters in PB5 to byte savevalue PAS32 and the parking facility number is PB14 to byte switch PAS3L in a set condition. PB6 is tested for a value 1, to determine if the greeter transaction is required to simulate removal of the vehicle from the parking facility and perform a
subsequent pickup of the terminating passenger transaction at curbside. A value of 1 in PB6, indicating the routing of passengers and greeters to the parking garage, requires only the use of the passenger group transaction, and the greeter transaction is terminated.

Greeter transactions with PB6 unequal to 1 have zero assigned to PB5 and GRT03 assigned to PH4. These transfers to CTRL1 to proceed to the parking facility. Greeter transactions unable to unlink a passenger transaction from GREGC are transferred to GAT10. At this location, they are assigned the address GATE8 in PH4 and link on the chain GREGC for later release by a deplaning passenger transaction in the Deplaning Passenger Logic Section.

At GATE7, enplaning passenger transactions enter a TEST block to determine if well-wishers are included in the simulated passenger group. Transactions without well-wishers represent transfer to GATE3. Those transactions including a representation of well-wishers enter a SPLIT block as the value is PB13 to PB5 and transfer to GATE3. The copy transaction simulates the well-wishers. The number of passengers in PB13 of this transaction is simulated from PB5 and PB13 is made equal to zero. A well-wisher routing function, WWG1F, is assigned to PB1. The pointer PB2 is reset to 1 and the transaction transfers to CTRL0 to begin the routing process out of terminal area.
Passenger transactions performing the gate check-in operations proceed from GATE3 to GATE1 when they enter a QUEUE block. The total number of passengers to reach gate counters, represented by halfword savevalue GATXH, is incremented by PB13. The simulated waiting time in a gate queue is recorded in PH11. The transaction enters the PH5 storage, simulates the service time by an ADVANCE block then leaves the PH5 storage.

A GATE block directs transactions to GATE2 unless the flight transaction has placed the logic switch PH5 in a SET position. This setting operation is performed at the start of the simulated boarding time. Transactions transferred to GATE2 increment MH1(PH1,12) by PB13, then transfer to CTRL0 for subsequent termination. Those transactions arriving at the GATE block after boarding time decrement the occupancy count at the point number and then transfer to CTRL0 for termination.

Flight transactions for departing flights generated in the Deplaning Passenger Logic Section are transferred to GATE9 to simulate the boarding process. Each flight transaction is generated when the absolute clock time is one hour before the scheduled departure time. At GATE9, the transaction is held until absolute clock time equals the flight time minus a boarding interval, BDTXH. The GPSS storage number of the gate is assigned to PH5 and the PH5 logic switch is placed...
in a set position, allowing passenger transactions executing the LEAVE block at GATE6 to proceed directly to CTRLO for termination.

The flight transaction is held at an ADVANCE block until the absolute clock equals flight time. The transaction then resets the PH5 logic switch and resets MH1(PH1,12) to zero, allowing this element to record the numbers of simulated passengers missing the flight. The flight transaction is then terminated.
13. Ground Transportation (Misc.)

This module is accessed by passenger and visitor transactions. Self-driver, deplaning passenger transactions are routed to GRTO0; enplaning passengers proceed to GRT01. Greeter transactions simulating initial entry to the terminal area proceed to GRT02. The greeter transaction designated to proceed from parking to curbside after matching respective passenger transactions at gate, bag claim, security or lobby, transfer to GRT03. Well-wishers, separated from the passenger group at the gate or security, are routed to GRT00.

Deplaning passenger transactions routed to GRTO0 execute a HELPA call to FORTM. For this transaction type, the parking facility number assigned in GPSS to PB14 is used to obtain other parameter values. The values assigned by FORTM are the point number of the parking facility in PH2, the storage number in PH5, the landside facility number in PH7, and the value 10 in PB11. The transaction is advanced by the amount TRUXH, the walking time from the last facility to the parking facility. Following this delay, the transaction transfers to CTRL8 for direct routing to the parking garage exit.

Enplaning passenger transactions entering the terminal area for parking are routed to GRT01. At this program location, a HELPA call to FORTM is executed and returns with the assignments made to the same parameters as deplaning passenger transactions at GRT0.
Greeter transactions entering the terminal area for entrance to parking facilities are routed to GRT02. At this block, and the three following blocks, the assignment to parameters and incrementing of counts are exactly the same as those of the enplaning passenger at the subsection starting at GRT01.

Greeter transactions assigned to remove vehicles from the parking facility after passenger matching for proceeding to curbside are routed from gate, bagclaim, security or lobby to GRT03. At this program location these transactions execute a HELPA call to FORTM. All parameter assignments are identical to those described for transactions entering GRT00.

After return from FORTM, an ADVANCE block simulates the walking time from the last facility to the parking facility. The greeter transaction then simulates a vehicle and enters the PH5 queue, if necessary, at the parking lot exit. The service time at the parking facility payment booth is simulated. The parking lot inventory is decremented by 1 and the count of vehicles proceeding from parking lot to curbside is incremented by 1. The car is routed to the Deplaning Curb Logic Section by assigning DCAR1 to PH4 and transferring the transaction to CTRL1.
MICROCOPY RESOLUTION TEST CHART
NATIONAL INSTITUTE OF STANDARDS 1983
14. Parking (Dept. Pax-Cars)

The deplaning passengers transactions previously routed to GT00 are routed to PARKO, the first program location of this section. At this point, the PH5 queue is entered and the time of entrance is recorded. After departing the queue to enter service at the vehicle parking payment booth, the time spent queuing is recorded in PH11. The service time at the booth is simulated, then the vehicle inventory in the facility is decremented by 1 and the count of vehicles on the departing road is incremented by 1. The transaction is transferred to CTRLO.
15. Rent-A-Car

Deplaning passengers transactions selected for the car rental mode are transferred to RCARO to begin the simulation of processing performed at the car rental counter. Byte parameter 6 is tested for a value 3, a flag designating usage of this mode. Transactions not utilizing car rental are immediately transferred to CTRLO. Transactions proceeding to the next block select a car rental agency from the selection function RCA2F and assign the agency number to PB10. A HELPA call to FORTM determines the agency counter nearest to the facility immediately departed by the passenger transaction. FORTM returns with the point number of this counter in PH2, the storage number in PH5, the landside facility number in PH7 and the value 11 in PB11. The walking time from the last point to the car rental counter is simulated by TRVXH. The transaction enters a QUEUE block, and subsequently the PH5 storage. The waiting time is recorded in PH11 and the processing time is simulated by RCAIV. This transaction proceeds to CTRLO for further processing.

Deplaning passenger transactions using the car rental mode are assigned to RCAR9 by the ground transportation selection function CTRlF. These simulate passengers who have completed processing inside the terminal building and are proceeding to a location to obtain the vehicle. This logic assumes that the passenger picks up the vehicle at an agency
parking facility.

At RCAR9, a HELPA block is executed. FORTM determines the number of the agency facility and returns with the point number of this facility in PH2, the storage number of the facility in PH5, the landside facility number in PH7 and the value 10 in PB11. The travel time to this facility is advanced. The transaction then increments the count of vehicles on the departing road by 1 and transfers to CTRLO.

Enplaning passenger transactions arriving at the airport landside who are proceeding to return rental vehicles are routed to RCAR8. The HELPA call to FORTM returns with the same assignments as for transactions using RCAR9. The arriving roadway count, ARDXH, is incremented by 1, then the transaction is transferred to CTRLO.
16. Security

All transactions simulating entry to concourses and subsequent movement to gate facilities pass through the Security Module. At SECUO, a HELP call is made to determine the security facility for the destination gate. FORTM returns with the security point number assigned to PH2, the storage number in PH5, the landside facility number in PH7 and the value three in PB12. The transactions are advanced by the travel time between security and the last facility used. A test comparing PB5, the number in the party, with PB13, the number of passengers in the party, determines if the transaction represents any simulated well-wishers. When PB5 equals PB13 there are no well-wishers, and the transaction is transferred to SECU3. When PB5 does not equal PB13 well-wishers are present, and the transaction continues to the next sequential statement. A statistical transfer is made, using halfword savevalue WWXGH. If the result of the statistical transfer requires separation of well-wishers and passengers at security, the transaction goes to the next sequential block. Otherwise the transaction is transferred to SECU3. At the next sequential block, the transaction is split once. The copy transaction, representing well-wishers, is transferred to the third sequential block. For the parent transaction, PB5, the number in the party, is made equal to PB13, the number of passengers in the group, thereby removing
well-wishers from the party count. The passenger transaction is then transferred to SECU3.

The well-wisher transaction removes the number of passengers in PB5 by subtracting the number in PB13 from it, then assigns the value zero to PB13. PB1 is assigned WWS1F, the well-wisher leaving from security process function. The process function pointing PB2 is reset to 1, and the well-wisher copy transaction is transferred to CTRLO.

At SECU3, the PH2 halfword savevalue, representing occupancy at the security point, is increased by PB5, the number in the party. At SECU1 the PH5 queue is entered with an entry count of PB5, then the transaction is marked to record waiting time. The PH5 storage representing security service is entered and the waiting time is added to PH11. At SECU2 the passenger transaction is advanced by V$SEC1V, the security check time.

The occupancy count at the point, which is kept in the halfword savevalue whose number is PH2, is decremented by PB5, the number in the party. The matrix savevalue, MH12(SECNN,1) is incremented by PB5. This matrix is the count of simulated passengers and visitors processed through the security facility. SECNN is the number of the security facility.
17. Concessions

A number of simulated transfer and transit passengers, selected by random draw in the Deplaning Passenger Logic, are sent to their section to spend part of the waiting time for their departing flight.

Concessions are located in the lobby and on the concourses, and the routing to either facility is also specified during the selection process. Transactions are routed to LOBCO for lobby concessions and to CONCO for concourse concessions. Program Logic for both facilities is identical.

At LOBCO or CONCO a HELPA FORTM call is made, with a flag set for lobby or concourse concession, respectively. The FORTM subprogram calculates the travel time to the concession and assigns the time spent at concession to PH5. The transaction is advanced by TRUXH, the travel time from the last facility. The savevalue whose number is the same as the point number of the concession is increased by PB5, the number in the party. The transaction is next advanced by PH5, the simulated time spent at the concession. The savevalue, which keeps a count of the congestion at their point, is decremented by PB5 to indicate the transaction has left the concession, and the transaction is then transferred to CTRLO.
Transfer Flights

This module establishes and maintains a table in MH5 containing the MH1 row number of simulated departing flights able to accept transfer passenger transactions. The variable TPAX of the input flight schedule specifies the initial number of transfer passenger transactions accepted by the flight. During the operation of the simulation, the number of transactions is decremented by the transfer passenger logic in FORTM. Flights are added to or deleted from the MH5 table as simulation time progresses. If an insufficient number of transfer passenger transactions were assigned to the flight prior to deletion from the transfer flight table, this module generates transactions to complete the assigned number of transfers.

A single transaction is split from the initial transaction generated by the Deplaning Passenger Logic Section and transferred to XFLTO. At XFLTO the domestic transfer process function, TDPIF, is assigned to PBl. Byte parameter 2, the process function pointer; is assigned a value 2 to place these transfer passengers at security; Byte parameters 5, the total number in this passenger groups; and 13, the number of passengers in the group, are assigned the value 1. A HELPA FORTM call is made, with a zero value D-operand used as a flag, to initialize the transfer flight table in MH5. At this initial pass, FORTM arranges the MH1 row numbers of simulated departing flights accepting transfer passenger transactions in MH5,
in chronological order. The simulated departure
times of these flights must be between 30 minutes and 2
hours, or between the FORTRAN input limits DELETE and ADD
respectively, after the simulation start time. FORTM
returns with the MHL row number of the first flight depar-
ting after the ADD or two hour time limit in PH1. If the
last row of the flight table matrix was reached during this
initialization process, the number of the next succeeding
row of MHL is assigned to PH1.

The next block tests MHL(PH1, 1) for a value less
than zero. If the end of the flight table matrix was reached
in FORTM, the MHL (PH1,1) value is minus 1. When this occurs,
the transaction proceeds to the next block to determine if any
flight accepting transfers was located by FORTM with subsequent
placement of the MHL row numbers in MH5. The MH5 (1,1) element
is tested for a zero value to determine if an MHL row number is
present. When MH5 (1,1) is zero, the transaction is terminated
If MH5(1,1) is non-zero, the transaction transfers to XFLT3
to delete flight numbers from the transfer flight table as
simulated time progresses.

When the returning value of PH1 does not indicate that
the last row of MHL was reached, the transfer flight transaction
transfers to a SPLIT block. The parent transaction proceeds
to the next succeeding block to attempt to add the PH1 value
to the MH5 flight table. The copy transaction transfers to

A-1-99
XFLT3 to attempt to delete the MH5(1,1) element.

When adding the current PHI and subsequent MH1 row number values to MH5, an initial test is made of MH1 (PH1,11) for a value greater than zero to determine if simulated transfer passengers were input for the flight. The transaction proceeds to the next block, at location XFLT1, when MH1(PH1,11) is greater than zero, otherwise it is transferred to XFLT2. An ADVANCE block at XFLT1 delays movement of the transaction until absolute clock time equals flight time minus the ADD time. Following this delay, HELPA FORTM call is executed with a flag value of 2 to implement addition to PHI to the MH5 table. Upon return from FORTM, PHI is increment by 1 at XFLT2 and the test of MH1(PH1,1) for the end of the table is again performed. If the test indicates the end of the table, the transaction is terminated, otherwise it is transferred to XFLT5. At this location, the Boolean variable XFL1B is tested to determine if the MH1 row number represents a departing flight and has been assigned to accept transfer passenger transactions. When XFL1B is true the transaction proceeds to the next block for transfer back to XFLT1. If it is false, the transaction transfers immediately to XFLT2.

The deletion process begins at program location XFLT3. The MH5(1,1) element is tested for a value greater than zero. If transfer flights are available, as indicated by a non-zero value of MH5(1,1) the value of this element is assigned to PHI,
otherwise the transaction is transferred to XFLT9. After
the above assignment to PHI is performed, an ADVANCE block
at XFLT7 delays the transaction until the absolute clock
time equals flight time minus DELETE time. The gate
number of the flight is assigned from MHI(PHI,9) to PH5
and flight type from MHI(PHI,7) to PB3. A HELPA call to
FORTM with a flag value 3 is executed to determine
the point number of the ticket counter corresponding to
the airline number in MHI(PHI,3) of the flight. Transactions
are then generated to create remaining transfer passenger
assignments if the flight was not filled by the Deplaning
Passenger Logic Section. The number of transactions required
to fill the flight is contained in MHI(PHI,11). A SPLIT block
creates these transfer passenger transactions and transfers
them to XFLT8 immediately before deletion of the row number
from MH5.

A HELPA call with a flag value of 1, deletes the MHI row
number contained in PH1 from MH5. Upon return from FORTM, the
PH1 value is incremented by one at XFLT4. The next block,
located at XFLT9, tests MHI(PHI,1) for a negative value to
determine if the end of the flight schedule matrix has been
reached. The negative value terminates the transaction,
otherwise the transaction transfers to XFLT6.
At XFLT6 the Boolean variable XFL1B is again tested. If XFL1B is true the transaction proceeds to the next block for transfers to XFLT7. If false, the transaction transfers to XFLT4 to increment PHI.

Transfer passenger transactions routed to XFLT8 are advanced a random delay time between 0 and 15 minutes before proceeding to the next block. After this delay, they transfer to CTRL0 to proceed from ticket counter point to security.
Change Card Reader

A single transaction, used to perform storage capacity changes is generated, with a priority level of 120. At CHGOO, this transaction performs a HELPA call to FORTM. The initial call by this transaction reads the first change card and assigns the time difference between the current clock and time of the first service change to halfword savevalue CHGXF.

The transaction tests the halfword savevalue NSCXH, the number of storage changes designated by the CHANGE data card, for a value greater than zero. For the initial pass, NSCXH is zero, and the transaction transfers to an ADVANCE block. When the clock time equals the change time, the transaction proceeds to the next block and tests NSCXH for zero. Again, at the initial pass NSCXH is zero and the transaction transfers to CHGOO.

The second and subsequent HELPA calls to FORTM at CHGOO assign GPSS storage numbers to MH7(I+30,1). Subscript I refers to the Ith storage designated for a capacity change on the CHANGE input data card. The total number of storage to be changed is assigned to NSCXH. This information is transferred from the data card read on the previous HELPA call. Thus, the data from the initial FORTM call is assigned on the second call. FORTM then reads the next input CHANGE card, assigns the time interval between the current time
and the next change to CHGXF and returns to the GPSS main program.

The TEST block examines NSCXH for a value greater than zero and the transaction proceeds to the next block for this condition. PH1 is assigned a value of minus 1. The transaction is then split NSCXH times and each copy transaction is sent to block CHG01. Sequence numbers of these transactions are assigned to PH1. When the transaction is split, the value of PH1 is incremented by 1, producing the value zero in PH1 of the parent transaction and sequence numbers beginning with 1 in PH1 of the copy transactions. These sequence numbers will be used to address elements in MH7.

The parent transaction proceeds to the next block where it is delayed by the value CHGXF. This block is also the destination of the transaction if NSCXH is zero before the splitting process. After leaving the ADVANCE block, a TEST block holds the transaction until NSCXH is decremented to zero by the copy transactions. Thus, if NSCXH is greater than zero, matrix MH7 is still used by transactions generated by the previous change, and the parent transaction is held at the TEST block until MH7 is available. The parent transaction then returns to CHG00 to process the next change.

At CHG01, the destination of the sequenced copy transactions, PH2 is assigned the value in PH1. The halfword savevalue SAVXH is assigned the value of the sequence number
in PHl in order to save the pointer to the MH7 row. Half-word parameter 2 has the value 30 added to its current contents. The number of the storage to be changed and the storage capacity information are next assigned to PHl and PH2 respectively, from MH7(PHl,1) and MH7(PHl+30,1). Using the halfword savevalue SAVXH, the pointer to the storage number in MH7, the MH7(SAVXH,1) element, is reset to zero. The capacity information element at MH7(SAVXH+30,1) is also assigned the value zero. Halfword savevalue NSCXH is next decremented by one to indicate that the number of transactions using MH7 has decreased by 1. When the new capacity of the PHl storage equals or exceeds the current contents, PH2 contains the value zero. PH2 is tested for zero and the transaction continues to the next block for this condition. Because the remaining capacity of the PHl storage has been reset in FORTM, to the difference between the new capacity and current contents, the storage may change status from full to not full. Transactions waiting to enter a storage which was full before the FORTM reset are on a delay chain not scanned by GPSS until a transaction in that storage executes a LEAVE block. In order to activate these waiting transactions immediately, the transaction performing this change will attempt to enter and leave the storage. A GATE SNF block allows the transaction to enter the storage if it is not full. It will then execute a LEAVE block for the PHl
storage and be terminated. If the storage is full, transactions will undergo the normal wait, and the changing transaction will be terminated immediately. If the storage is entered, FORTM adjusts the count of transactions to discount the changing transaction. When PH2 is not zero, the new storage capacity is below the current contents and must be lowered. The TEST block transfers the transaction to CHGO2.

At the block labeled CHGO2, the location that the change transaction is transferred to if the storage capacity is to be lowered below the current contents, the change transaction is held at a GATE SNF block until another transaction has left the PHI storage, and therefore the storage is no longer full. After passing the GATE block, a HELPA FORTM, call is made to lower the storage capacity to the current contents. If the new capacity specified on the data card equals the current contents, the FORTRAN program sets a halfword savevalue SLCXH, used as a flag, to 1. The value one is SLCXH is next tested for in the GPSS program. If SLCXH is 1 the storage lowering is complete. The flag SLCXH will be reset to zero and the transaction is terminated. If the current contents exceed the new capacity, the number of available units of storage is set to zero and SLCXH remains zero. Because the flag SLCXH is not set to 1, more lowering of the storage capacity is needed. The change
transaction is transferred back to CHGO2 to wait until another transaction leaves the storage.
Timer Section

A single transaction with priority level 127 and one halfword parameter is generated to initiate and, subsequently, to terminate the activity of the Main Program. An advance of XH$CLKXH is done to define CLKXH as a halfword savevalue. XH$CLKXH has the value zero at simulation start, and consequently no actual advance of the simulation time is done. The clock increment savevalue, INCXH, is set to 60 to represent 60 seconds of simulation time. A HELPC call to subroutine CLINK initiates linking of this program and the supporting FORTRAN subroutine FORTM. The next block is a HELPA call to FORTM which completes the linking. After execution of these statements, any HELPA call to FORTM appearing in the Main Program will operate with two way communication between the two programs. FORTM also reads the input data and places the information in GPSS main program matrices. Any errors detected by FORTM during the initialization and input phase are indicated by setting PHI of this transaction to a value greater than zero. Upon return from FORTM, PHI is tested for a value greater than zero. For this condition, the transaction splits repeatedly and goes to a TERMINATE PHI block in order to guarantee termination.

When PHI is zero, the transaction is transferred to a SPLIT block. The parent transaction is held at a GATE LS block to provide a method of terminating activity when errors accumulated by FORTM exceed a specified number. If this
should occur, logic switch JOBLS is placed in a set condition and this transaction proceeds to a SPLIT block at STOP1. Repeated copies activate a TERMINATE 100 block to force simulation termination.

The copy generated before the GATE LS block is again split. This parent is advanced by the variable V$INClV, which is one less than XH$INCXH, the clock increment time. This compensates for the GPSS simulation starting time which is 1 instead of zero. A HELPA call to FORTM is performed to update the clock time in the FORTRAN subprogram by the amount INCXH. The clock time, CLKXH, is kept in a 24-hour format showing hours and minutes. After an advance by the clock increment XH$INCXH, the transaction is transferred back to the HELPA FORTM, 21 block to again update CLKXH. This procedure of advancing by XH$INCXH and then making a HELPA FORTM, 21 call is continued until the simulation ends.

Halfword PHI of the copy transaction is assigned the value of 127 which will be used as a loop counter. The function RANDF is used to place a random number between zero and 1 in the matrix element M1(l,PH1). PHI is then decremented by 1 by the LOOP block and the next location in the M1 matrix is filled with a random number. This process is continued until all 127 locations in M1 are filled with random numbers.

The priority of this timer transaction is set to 126, which is 1 less than the priority of the transaction.
updating the clock. This will allow the clock to be updated at the last time increment before the run is terminated. The timer transaction is then advanced to the end of run time, and an HELPA FORTM,20 call provides the printing of FORTRAN formatted summary reports. A selected list of the regular GPSS output is also printed out. A TERMINATE block with an A-operand of 1 causes the simulation run to be terminated.

Another single transaction is generated 11,700 seconds after the simulation has started. This transaction entry block is HELPA FORTM,22 to print-out flow and queue length information. The transaction is then advanced by 300 seconds and transfers back to the HELPA FORTM,22 block. This activity is continued for the rest of the simulation so that information is printed out every five minutes of simulated time.

Another GENERATE block generates one transaction 3 hours and 25 minutes after the start of the simulation. This transaction then encounters a TERMINATE 1 block and stops the simulation. This procedure is used in conjunction with the RESET command in order to clear out the simulation statistics accumulated during the start up time. If this feature is not wanted then both the GENERATE block and the RESET block should be commented out.
The final GENERATE block produces a single transaction which increments the value of byte savevalue PXTBN by 1. PXTBH will be used as the number of the table providing hourly frequency distributions of simulated landside waiting time. The time is advanced one hour by an ADVANCE block, and the transaction is transferred back to the block that increments byte savevalue PXTBN by 1. This procedure is continued until the end of the simulation.
APPENDIX A-2
FLOW CHARTS
CONTROL

CTRL8

4, FNS*PS1.PH
ASSIGN

2+1, PB
ASSIGN

Increment process function pointer

CTRL1

TRANSFER
PK4

Handles routing of deplaning pax to parking lots

CTRL3

TRANSFER
PARK3

Route deplaning pax by ground transportation mode

CTRL8

4, FNSCTR1F.PH
ASSIGN

TRANSFER
PK4

Route enplaning pax by ground transportation mode

CTRL1

4, FNSCTR2F.PH
ASSIGN

TRANSFER
PK4

Address of next operation
GENERATE

Generate Transaction for bus/limo service to enplaning and deplaning curb

ASSIGN

Assign a location

SPLIT

Fall through if no arriving bus simulated

XH$ABUXH

TEST

TERMINATE

Interval between arriving buses

COTR2

Advance XH$ABUXH

SPLIT

TRANSFER

ENFC6

COTR3

COTR2

COTR3

A-2-4
Fall through if no deplaning bus simulated

Interval between departing buses

Deplaning pax waiting time

A-2-5
ENP99

TABULATE PAXWT

TABULATE XBSPXBTN

TERMINATE

Enplaning pax waiting time

Enplaning pax waiting time by hour

TRX99

TABULATE PAXWT

TABULATE XBSPXBTN

TERMINATE

Transfer pax waiting time

Transfer pax waiting time by hour
DEPLANING PASSENGER LOGIC

Generate one transaction for deplaning pax logic

Initialize Process Function Pointer

Copy to transfer flight logic

Assure that transfer flight table gets built

Deplaning pax at 'GATE'

Flight Table (MH1) row subscript

A value of -1 in MH1 (PH1,1) indicates end of flight table

A-2-7
A

TERMINATE

B

ADVANCE VSFLT

SPLIT 1

DEPLO

 Delay to 1 hour before scheduled arrival

E

TEST

DEPLS

 Branch for arriving flights

MHL (PH1,1)

4, GATE 9, PM

ASSIGN

 For gate logic to start boarding

TRANSFR

CTRL1

DEPLS

3, MHL(PH1,7), PB

ASSIGN

Type of flight:
Domestic = 1
Commuter = 2
International = 3
Tests for domestic flight
Deplaning domestic route function
No. of BAGS function-Domestic flight

Test for commuter flight
Deplaning commuter route function
No. of BAGS Function-Commuter flight
Deplaning international route function

No. of Bags Function - International Flight

Is transaction Sequence No. GE 32000

% of greeters meeting at curb

% of parking greeters proceeding to curb

To curb with greeters logic

To park with greeters logic

To curb without greeters logic

Reset transaction sequence no.
DEP25

6.1 PB
ASSIGN

No. of parking greeters proceeding to curb

8.VSFLT2V.PH
ASSIGN
TRANSFER
DEP16

Node = Park + curb

DEP15

6.2 PB
ASSIGN

No. of pax to be greeted

8.VSFLT3V.PH
ASSIGN

DEP16
PH3

E
TEST

0

FALL THROUGH IF NO GREETERS

TERMINATE

M SAVE VALUE

PH3 - PH8.WH

10.VSDE3V.PB
ASSIGN

Add to total met

Starting point in Random number table

A-2-11
DEP12

10+.1, PB
ASSIGN

PB10
G
TEST
124

10.1, PB
ASSIGN

B-, FN*PPPDP, PB
ASSIGN

10+.1, PB
ASSIGN

A-2-12
Fall through for gate meeting
Meet at gate
Set flag for meeting at gate
Fall through if no bags
Meet in lobby
Meet in bag claim
Fall through if need more parties

Reset PB14 for later use

No. of greeters

Increment transaction Counter

Transaction sequence no.

Advance to airport
Mode = Curb

No. of pax to be met

Fall through if none

Add to total met

Starting point in random number table

Will go to curb

A-2-15
DEP13

10+,1, PB
ASSIGN

Increment random number table pointer

P810
G TEST 124

Is pointer at end of table

10,1, PB
ASSIGN

Reset pointer to 1

G-PH
124-7
ASSIGN

Subtract no. of pax from total

10+,1, PB
ASSIGN

Increment pointer

4,7-NP14, PB
ASSIGN

Compute no. of bags

PH8
G TEST 0

Fall through if need more parties

SPLIT

DEP13

M
14.0.P8
ASSIGN

5.FM5CRPBP.PB
ASSIGN

SAVE VALUE
SEQ1H+1.XH

10.XHSE21H.PH
ASSIGN

TRANSFER CTRLI

DEPL7

ADVANCE 3600

GATE LR DPLIG

I

Reset P814 for later use

Greeters per party; not used unless recirculating and parking

Increment transaction Counter

Transaction sequence no.

Advance deplaning pax to time of arrival

Guard 'BAGS' from the simultaneous arrival of aircraft
LOGIC

set switch that one flight is being processed

ASSIGN

Gate number

ASSIGN

Last MH9 row

ASSIGN

Point no. of gate

SAVE VALUE

Maximum pax unloading time

SELECT

Pick free bag claim chain and switch

LOGIC

PH6
Copy to terminating pax

Copy to transit pax

Copy to transfer out-of-system

Fall through if no departing flights in transfer table

Add transfer pax to holding save value

Fall through if no transfer pax

A-2-19
K

8.2.PB
ASSIGN

13.1.PB
ASSIGN

5.1.PB
ASSIGN

SPLIT
MHL(PL1.11)

TERMINATE

HELPA
FORTH 17,".RHS,.PB3,.PB5

PB3
TEST
D

DEP30

L,TIPE,.PB
ASSIGN

L

Mark as transfer pax

No. of pax in party

Total no. of people in party

Create transfer pax

Switch, Random number, Pax type, gate no.

Fall through if international pax

International transfer process function

A-2-20
Starting point in random number table

Fall through if international flight

Assign Dom. Con. transfer function

Set pax as not ticketed

Is outgoing flight international

Skip IMSIS, BAGCS, COSTS

Stagger pax off aircraft

A-2-21
Gate no. of departing flight

SAVE VALUE
SAVXN,MH1(PH1,9),XH

VSXFL4V

L 45
TEST
N
Fall through if departure is less than 45 minutes

MH9(PH3,4)

E
TEST
MH9(XH$SAVXN,4)

O
Fall through if same section

1,PNDOSSP,DP
ASSIGN

P
Short-stay/same section
DEPL6
ASSIGN

13.1.PB
ASSIGN

SPLIT

MH1(P:1,1)

TERMINATE

ADVANCE
VSA CUZ V

HELP A
PROM, 17, 2, PB

S,NSTDPFX, PB
ASSIGN

0

MARK as transit

No. of pax in party

Total no. of people in party

Stagger pax off aircraft

Transfer pax section
switch, arriving flight

Select a process function
E 0 fall through if never deplaned

TEST

TERMINATE

TRANSFER

PB1

DEP10

ASSIGN

ASSIGN

ASSIGN

SPLIT

MH1(PH1,16)

TERMINATE

Mark as transfer-out-of-system

No. of pax in party

Total no. of people in party

Create transfers out-of-system
ADVANCE  
V5ACU2V  

1, TOSEP, PB  
ASSIGN  

TRANSFER  
CTRL0  

MARK AS TERMINATING Pax  

DEPL4  
S, I, PB  
ASSIGN  

SAVE VALUE  
SEQ2H, 2, XM  

SPLIT  
1  

TEST  
GE  

J200  

XHSEQ2H  

TEST  

To curb with greeters logic  

To curb with greeters logic  

A-2-26
SPLIT EPS To park with greeter logic
SPLIT To curb without greeter logic
SPLIT To park without greeter logic
SPLIT To all other modes
ASSIGN Assign address parameter
TRANSFER For following assign only
ASSIGN No. of pax in modes other than curb/park
Fall through if none

Starting point in random number table

Increment random number table pointer

Is pointer at end of table

Reset pointer to one

No. of Pax in party

Subtract from total
U

PH8

LE

TEST

0

13=PH8,PH

ASSIGN

TRANSFER

SPLIT

DEPL9

1

5,PB1,PB

ASSIGN

HELP

FORTH,5,0,RN5,PB3

10=P,PB

ASSIGN

HELP

BAGS,PH1,PH*PB14,4,3,PB8

V
Reset PB14 for later use

If rent-a-car, use DDPIF

Fall through if no bags

Change to lobby process function

Mode = curb

No. of parking greeters proceeding to curb

A-2-30
DEP19
6,2, PB
ASSIGN

6, VSFLT1V, PB
ASSIGN

DEP20 PH8
E
TEST
0
TERMINATE

10, VSDFL3V, PB
ASSIGN

DEP21 10+1, PB
ASSIGN

PB10 G 124
TEST

10, 1, PB
ASSIGN

13, FNSPPDF, PB
ASSIGN

Mode = park
No. of pax to be greeted
Fall through if none
Starting point in random number table
Increment random no. pointer
Prevent pointer from exceeding 127
Reset pointer
No. of pax in party
IS-PB13, PH

ASSIGN

12.1, PB

ASSIGN

10+.1, PB

ASSIGN

HELP
BAGS, PH1, PH*PB14, 4, 3, 4, BS maximum random no.

Increment random number pointer

Increment random number pointer

ML1(1, PB10) XLSGRCXL

LE TEST

ML1(1, PB10) XLSGRCXL

LE TEST

PH8

TEST

0

Is last party size too large

Y

Subtract group from total

Meet at bag claim

Generate nc. of bags, maximum random no.

Fall through for gate meeting

Meet at gate

Y

A-2-32
Adjust last party size

Fall through if not international

Fall through if no bags

Change to lobby process function

Meet at ticketing

Reset PB14 for later use
5.PB13.PB
ASSIGN
SAVEVALUE
SEQ2H+,1,XH
ASSIGN
TRANSFER DEP24

DEP22
6.1.PB
ASSIGN
8.VSFLT V.PH
ASSIGN
TEST
0
AA
TERMINATE

Total in party
Increment transaction counter
Transaction sequence no.
Mode = curb
No. of pax to be met
Fall through if none
Starting point in random number table

Increment random number pointer

Prevent pointer from exceeding 127

Reset pointer

Pax in party

Subtract group from total

Increment random number pointer

Generate no. of bags, maximum random no.
PS

PB

Adjust last party size

PB

13+,PB8,PB

ASSIGN

TRANSFER

SPLIT

Fall through if not international

PB3

NE

1

Fall through if no bags

PB4

E

0

Change to lobby process function

PB41F,PB

ASSIGN

14.5,PB

ASSIGN

5,PB13,PB

ASSIGN

CC

Reset PB4 for later use

Total in party

PB

LE

0

Fall through if

international
DP27 10+.PB
   ASSIGN
   DEP10
   PB 10
   PB
   TEST 124
   PB
   ASSIGN
   PB11
   ASSIGN
   12.PN5PPPF,PB
   ASSIGN
   PB12
   ASSIGN
   PB8
   TEST 0
   EE
   PB
   ASSIGN
   TRANSFER
   FF
HELP Generate no. of baqs.

DAGSPHIFNP314,4,3,PBS

maximum random no.

Reset PB14 for later use

Increment random number pointer

Total in party
Fall through if no bags

Change to lobby process function

Stagger pax off aircraft

Fall through if not meeting at gate

Try to unlink greeter

Wait for information from greeter

Add greeters to party
Pick up parking lot no.

Permit next of PASS save values

Will go to DEP29 when unlinked

Wait for greeters
ENPLANING PASSENGER LOGIC

First transaction on JOBTAPE which is a dummy, is sent here.

Fall through if transfer from out of system.

Entrance facility for out of system transfers.

Process function number.

Enplaning domestic pax.

Get random index for bag function.

No. of bags - domestic.
ENPL2

1.ECPIF,PB
ASSIGN

10.RSBN2IV,PB
ASSIGN

4.FNSCBAGF,PB
ASSIGN

TRANSFER 
CTRL0

Enplaning commuter pax
Get random index for bag function
No. of bags - commuter

ENPL3

1.EIPIF,PB
ASSIGN

10.VBN2IV,PB
ASSIGN

4.EMSIBAGF,PB
ASSIGN

TRANSFER 
CTRL0

Enplaning international pax
Get random index for bag function
No. of bags - international

A-2-43
Redirect actual enplaning pax transactions to CHNGS

Route pax according to whether domestic, commuter, or international flight
Return to control if no bags

Location M&l Row

Travel time to bag claim area

Fall through if greeter

Try to unlink pax

Only one at a time

Number of greeters

Parking lot number
LOGIC PSL

Let pax pick up information

Branch if mode is curb

No one in party

Route to greeter leaving parking

Will go to BAGC1 when unlinked

Make greeter wait for passenger
BAGC1

SAVEVALUE

PM2+, PB5, XH

WHI(PH1,1) E 0 BAGC7

TEST

4, BAGC7, PH

ASSIGN

MARK

PH6 FIFO

LINK

BAGC7

LI+, PH1, PH

ASSIGN

SAVEVALUE

PM2-, PB5, XH

PB12 E 2 BAGC4

TEST

CTREBC

10 PH

PB10

UNLINK

BAGC5

CTRL1

BAGC8

Congestion

Branch if all bags at bag claim

Save address parameter

Mark pax for waiting time

Wait for bags (unlink in Baggage unloading section)

Add waiting time to PH11

Congestion

Fall through if to be greeted at bag claim

Try to unlink greeter
Wait for information from greeter

Add greeters to party

Pick up parking lot number

Permit next use of PAS4 savevalues

Will go to BAGC6 when unlinked

Make pax wait for greeter
PRIORITY Allow pax to execute "BAGS"

SPLIT 40PB
9PH 1

TERMINATE

HELPA
FORTM, 2, 10

5, MH1(PH1, 8), PH
ASSIGN

5, FYBUNLF, PH
ASSIGN

LOGIC P
DPLIT

Allow next aircraft transaction to move in deplaning section

Baggage unloading logic

Aircraft type

Unload time friction (by aircraft type)
ADVANCE FN*PH5

5, VSBN4V, PH
ASSIGN

5, MH9(PH5, 3), PH
ASSIGN

4, 40, PH
ASSIGN

BINL2
PB*PH4

TEST

BUNL3

ADVANCE 30

XK

Baggage unloading time
Baggage claim index and area number
Point no. of baggage claim
Loop counter
Branch if all bags unloaded
Bag unloading time at baggage claim
3PH Bags to waiting pax

CTRL1 → PH6 → UNLINK LE → Bags to waiting pax

ALL JPH PHXPH4

LOOP → Loop for next group of bags

BUNL2 → Free chain/switch pair

LOGIC R → Mark all bags at claim area

BUNL3

MSAVE VALUE 1,PH1,1,2,MH

TERMINATE

A-2-51
TICKETING AND CHECKIN

**CHECK**

&check18

**HELPA**

**ADVANCE**

**PB13**

**CTRL1**

**UNLINK**

**CHE11**

**SAVE VALUE**

**SAVE VALUE**

**LL**

Branch if enplaning ticketed, and direct

Location, airline, ticketed, express check random no. of pax

Walking time to ticketing and checkin

Fall through if greeter

Try to unlink pax

Only one at a time

Number of greeters

Parking lot num'3r
Let pax pick up information

Branch if mode is curb

No one in party

Route to greeter leaving parking

Will go to CHEll when unlinked

Wait for greeter

A-2-53
P8126

E all through if to be greeted at ticketing

CTRLE Try to unlink greeter

GRELL Wait for information from greeter

GATE LS Add greeters to party

PAS2L Pick up parking lot no.

LOGIC Permit next use of PAS2 savevalues

TRANSFER Will go to CHEK8 when unlinked

PAS2L Wait for greeter

A-2-54
Branch if terminating
Congestion
Wait for free server
Mark pax for waiting time
Service
Add waiting time to PH11
Branch to CHEK2 or CHEA3. Set in FORTM
Note: Halfword SAVEVALUES may be inserted here to record flow through Express Check-in.
Note: Airline full service counter flow values may be recorded in MM11.

MM

LEAVE

PH2-,PB5,XH

SAVEVALUE

M(PPH1)

TEST

SAVEVALUE

TCKT-,PB11,XH

TRANSFER

CTRLF

Leave service

Congestion

It is airline no. 1

Add to total no. of pax for airline no. 1 at ticket counter
IMMIGRATION

HELPA
FORTH, S, PH2, PH1(PH1) (PH1.9)

Point number, gate number

ADVANCE
XHSTRXH

Walking time to immigration

SAVEVALUE
PH2*, PBS, XH

Congestion

GATE
SNF

PH5

IMM1

If storage full, transfer to IMM1

ENTER

PH5

Enter service

TRANSFER

IMM2

IMM1

QUEUE
PBS

PH5

Enter service

ENTER

PH5

NN

A-2-58
CUSTOMS

CUST#
HELPA FORMT, 4, PH2, PH5

ADVANCE XHSTRVXH
Walking time to customs

SAVEVALUE PH2+, PB5, XH
Congestion

CUST3
GATE SNF PH5
Enter service PH5

TRANSFER CUST4

CUST2
QUEUE PB5

ENTER PH5
Enter service

Location, MH9 row
CONCOURSE EXIT - TERMINATING PAX

CNCR# HELPA
FORMH,25,PH2,MH1(PH1,9)

ADVANCE XHSTRUVXH

"SAVEVALUE
11+,PH5.1,PB5,XH

SAVEVALUE
CONXH+,PB5,XH

TRANSFER
CTRL#
DEPLANING CURB (CARS)

DCARB

ADVANCE FNSDCARH

Arrival distribution for cars

DCARB1

SAVEVALUE

Count of vehicles on arriving road

ARGXH=1,XH

SAVEVALUE

Count of vehicles at deplaning curb

DEPXH=1,XH

HELPA

FORTM,10,MHI(PHI,3),PH1,PB4,0

PB4

NE TEST

Fall through if deplaning

8,VSDBPKD,PH

ASSIGN

One less than first deplaning curb storage no.

8,VSDBPKD,PH

ASSIGN

Save curb storage no. minus one

PB8,PH

ASSIGN

One less than first double parking storage no. for deplaning

PB10

NE 4 TEST

If circulating vehicle, transfer to DCARB

PP
One less than corresponding storage no.

Save curb storage no.

Save curb storage no. minus one

Double parking storage no. for enplaning

If circulating vehicle, transfer to DCARB

One less than corresponding enplaning storage no.

Increment storage no.
I ZIE If circulating vehicle skip next test
TEST
If storage number same as destination storage number, go to DCARA
ME MG If storage number same as destination storage number, go to DCARA
storace number same as TESTA destination storage number, go to DCARA
ASSIGN Increment double parking storage number
QUEUE Enter queue for blocked lane
17Stoo traffi if double parking and no. of passing lanes available is one
DEPART UPH Depart queue for blocked lane
DCARC Fall through if current contents of double park storage is zero
SAV-VALUE
Set no. of passing lanes to maximum available

Save count

SAV-VALUE
Set no. of passing lanes to one less than maximum available

Save count

SAVEVALUE
Save time

SAVEVALUE
Delay vehicle according to no. of lanes available

ADVANCE

VSDLNDLY, XH

TOTL2+1, XH

TOTL1+1, XH

NOLAN, VSNLNM, XH

DCARC

RR

SS
Fall through if deplaning

If circulating vehicle is at last deplaning storage, transfer to DCARA

Move to next deplaning storage

If circulating vehicle is at last enplaning storage, transfer to DCARA

Count of vehicles on deplaning curb

For time limit enforcement
PB10  G  1  OCAR5
      TEST

PB10  G  2  OCAR4
      TEST

PB10  G  3  Branch to enter queue slot
      TEST

PB10  4,CIRC,P1

ASSIGN

TRANSFER

CTRL1  

ENTER

PH6

PRIORITY

UU

Branch to enter curb slot
Branch to enter double parking slot
Send to recirculation and
Enter Queue slot
To assure priority for slot
Get random no.

Link on queue in random order

Leave double parking storage

Unlink one from corresponding queue chain

Change PH6 to curb storage no.

Now at curb

Drop to normal priority
Enter curb storage

Get random no.

Relink in random order

Branch if vehicle unlinked is enplaning

Leave queue storage

Change PH6 to Double parking storage no.

Now using double parking

Set double parking priority
Curb or double parking storage

Match car with pax

Get random no.

Link in random order

'Time Limit Enforcement'

'Police Officer'

Frequency of enforcement

A-2-71
Unlink any vehicles which have been double-parked longer than the limit.

No. of deplaning curbs

Add no. of deplaning curbs

First queue chain

Unlink any vehicles in queue longer than limit

Increment curb or double parking storage no.

Check next curb or double parking storage
DCAR8

DCAR3

1

VSPDOV

UNLINK

Unlink one from corresponding queue chain

LEAVE

Double or queue storage

PH6

10

Drop to normal priority

PRIORITY

PB7

1

If enplaning, fall through

TEST

PB7

11+, M1, PH

Add waiting time to PH11

ASSIGN

PB7

E TEST

Branch if unlinked vehicle is enplaning

CIRCE

TRANSFER

Recirculate

CIRCE

A-2-73
DEPLANING CURB (PAX)

HELPA
PORTM, 9, PH2, PH11, PH7, PH1

ADVANCE
XH37RXXH

Location, facility type, MH9 row, MH1 row

Walking time to deplaning curb

TRANSFER
FN DPL IF

Branch by transportation mode

PRIVATE CAR
DPLC3

DPLC9

DPLC1

UNLINK E

Match pax with car

Branch if match successful

Match with greeter

Branch if match successful

DPLC9

10 PH PH10

DPLC1

DPLC2

DPLC2

DPLC1

DPCG1

A-2-74
DPLC1

GATE LS

Wait for car location information

PASSL

Pickup

6,XH$PASS1.PN

Car location information

ASSIGN

Reset logic switch

10,XH$PASS2.PB

Loading time

ASSIGN

Fall through if leaving curb

LOGIC R

Indirect addressing using variable

PASS2

Mark pax for waiting time

ADVANCE V$DPLL1V

Random pullout delay

PB10 E 1

DPLC7

SAVEVALUE

YY

MARK

ADVANCE V$DPLL2V

A-2-75
Add waiting time to PH11

Unlink 1 from chain with PH6 equal to corresponding double part storage

Unlink one from corresponding queue chain

Count of vehicles on departing road

A-2-76
DPLC2

ASSIGN

Set address parameter

DPLC9

LINK

Pax waiting for cars

GATE LR

Only one car at a time

SAVEVALUE

PASS1, PH6, XM

PQSS car location

SAVEVALUE

PASS2, P810, X13

Info to passenger transaction

LOGIC 5

Let next car pick up information

TERMINATE

A-2-77
BUS/LIMO - PAX

DPLC4

SAVEVALUE

DPCXH+,1, XH

TRANSFER

CTRL$4

Deplaning pax waiting on bus/limo

TAXI

DPLC5

ADVANCE

VFUFLIV

SAVEVALUE

DRDXH+,1, XH

TRANSFER

CTRL$4

Loading time

Count of vehicles on departing road

BUS/LIMO-VEHICLE

DPLC6

SAVEVALUE

ARDXH+,1, XH

Remove pax waiting for bus/limo

SAVEVALUE

DPCXH,8, XH

Count of vehicle on arriving road

SAVEVALUE

DRDXH+,1, XH

Count of vehicles on departing road

TERMINATE

A-2-78
DEPLANING CURB GREETERS (AFTER RECIRCULATE AND PARK)

HELPA
FOTH, 10, MH1(PH13), PH1, PB4, 1

ADVANCE X87EXKN

SAVEVALUE
PH2*, PB5, XH

DPLIC 1
UNLINK
NPH PH10

TRANSFER
DPCGO

DPLIC
LINK
FIFO

GATE LS
PASSL

S*, XBPASS2, PB
ASSIGN

II

Airline, flight, no. of bags, greeter
Walking time for greeter to deplaning curb
Congestion
Match pax at curb
Branch if successful
Otherwise link to special chain
Wait for information from greeter
Add greeters to party
14.XBSPASS3.PB
ASSIGN

Pick up parking lot number

LOGIC
R
PASSL

Permit next use of PASS sevevalue

6.1.PB
ASSIGN

Change mode to parking

2.1.PB
ASSIGN

Process function pointer

1.GRCPE.PB
ASSIGN

New process function

TRANSFER
CTRL$
DPCG2

GATE PASEL Only one at a time

SAVEVALUE Number of greeters
PAS52,P85,KB

SAVEVALUE Parking lot number
PAS53,P814,SB

LOGIC 5 Let sax pick up information

TERMINATE
ENPLANING CURB

PRIVATE CAR/TAXI

ENPCA

SAVEVALUE

ANDX+H/1,XH

SAVEVALUE

ENPXH+1,XH

HELPA

FORTN,11,NO1(PH1,H),PH6

8,VSDBPR1,PN

ASSIGN

4,PH8,PN

ASSIGN

5,VSIPCDS,PH

ASSIGN

PB10

NE

4

ENPCA

TEST

If circulating vehicle, transfer to ENPCA

8,VSENFLS,PH

ASSIGN

AB

Count of vehicles on arriving road

Count of vehicles on arriving road proceeding to enplaning curbside

Airline, mode

One less than first enplaning curb storage no.

Save curb storage no. minus one

One less than first double parking storage no. for enplaning

One less than corresponding first enplaning storage no.
ENPCA
ASSIGN
Increment enplaning storage no.

PBP
TEST
If circulating vehicle, skip next test

PH6
TEST
Is enplaning storage no. same as destination storage no.

ASSIGN
Increment enplaning curb storage no.

ASSIGN
Increment enplaning double parking storage no.

QUEUE
PH4
Enter queue for blocked lane

MARK
Mark pax for waiting time

BVSINFUL
TEST
Stop traffic if double parking
and no. of passing lanes available is one.

A-2-83
Add waiting time to PHI1

Leave queue for blocked lane

If current contents of double parking storage is fall through through

Set no. of passing lanes to maximum available.

Save count

Set no. of passing lanes to one less than maximum available.
Save Count

Save time

Mark pax for waiting time

Delay vehicle according to no. of lanes available

Add waiting time to PH11

If calculating vehicle is at last enplaning curb, transfer to ENPC1

Move to next enplaning storage no.

Count of vehicles on enplaning curb
AF

MARK

For time limit enforcement

PB10 G

Branch to enter curb slot

TEST

PB10 G

Branch to enter double parking slot

TEST

PB10 G

Branch to enter queue slot

TEST

4.CIRCL.PH

Sent to recirculation road

ASSIGN

TRANSFER

CTRL1

ENTER

PH6

12

To assure priority for double parking slot

PRIORITY

AG
AG

5, RH2, PH
ASSIGN

MARK

PH6

LINK

Get random number

Mark pax for waiting time

Link on queue in random order

ENPC8
PB7

E

TEST

DCAR3

Branch if vehicle unlinked
is deplaning

11+, M1, PH
ASSIGN

Add waiting time to PH11

LEAVE

PH6

Leave queue storage

10, 1, PB
ASSIGN

Change PH6 to curb storage no.

RXPH6

E

TEST

ENPC5

Using curb

Fall through if storage filled

AH
ASSIGN

Using double parking

Double parking priority

Enplaning curb unload time - car/taxi

Fall through if preticketed

Fall through if using curb side check-in
Curb side check-in time

Empty car parking time

Fall through if leaving curb

Indirect addressing using variable

Random pullout delay

Curb storage

Unlink one from queue
Double parking storage

Unlink one from queue

Count of vehicles on departing road

Fall through if no well-wishers

Location, mode, enplaning, zero, Lot no. only

Vehicle to lot

Enter car in parking lot
Vehicle count in parking lot

Vehicle count in parking lot

Air

Assign address

Mark pax for waiting time

Link to enplaning bus user chain

Add waiting time to PH11

A-2-91
BUS/LIMO (VEHICLE)

EMPC6

SAVEVALUE
ARDXX=1, XH

ADVANCE VSENP4V

CTRLIN A11

UNLINK

SAVEVALUE
BDPXH=1, XH

TERMINATE

Count of vehicles on arriving road

Bus/Limo load time

Bus leaves curbside

Count of vehicles on departing road

RECIRCULATION ROAD

CIRCS

ADVANCE VSCIRCb

SAVEVALUE BEXXX=1, XH

SAVEVALUE RCDXH=1, XH

Recirculate

Include recirculation in deplaning curb counts

Recirculation to deplaning
Branch N greater who has all ready met

Fall through to enter park or branch to deplaning road

Change mode to parking

Process function to curb
CIRCL

ADVANCE
VSCIRCH

SAVEVALUE

RLEXH+.L.XH

TRANSFER

Recirculate

Recirculate to emplaning

EXPRC
ENTRANCE

ENTRANCE

HELPA
FORTM. 12. PH2

ADVANCE
XHSTRVXH

DPDIM
TEST

DPLIN+, PBS, XH
SAVEVALUE

TRANSFER

SAVEVALUE
ENDIN+, PBS, XH

SAVEVALUE
ENDOR+, PBS, XH

TRANSFER

CTRLB

Location

Waiting time to entrance

A-2-95
EXIT

HELPA
FORTM,3PH2,PH1,PM+PBS.PH7

ADVANCE
XHSTRYXH

DDOUT.
TEST

SAVEVALUE
DPOUT+,PHS.XH

TRANSFER

SAVEVALUE
EPOUT+.PBS.XH

SAVEVALUE
EXOGR+.PBS.H+

TRANSFER
CTRL$

Location, current process:
next address, M99 row

Walking time to exit
GATE (ENPLANING PAX)

Location, gate

Walking time to gate

Fall through if no pax

Match with pax

Only one at a time

Number of greeters

Parking lot number

Let pax park up

information

GATE

HELPA
FORTM. 15, PH2, MHI (PH1.9)

ADVANCE
XHSTRVXH

PBB
E
TEST
GATE 7

CTRL7
IDPH
PH10

UNLINK
GATES

GATE LR

PAS3L

SAVEVALUE
PAS32, PBS, XB

SAVEVALUE
PAS33, PBS4, XB

LOGIC
5

PAS32

AL

A-2-97
Branch H mode is curb

No one in party

Rate to greeter leaving parking

Will go to GATE when unlinked

Wait to meet greeter

Fall through if have well-wishers.
SPLIT

1

ASSIGN

TRANSFER

GATE 3

ASSIGN

13.0.PB

ASSIGN

1. WWGFIP.PB

ASSIGN

3.1.PB

TRANSFER

CTRLS

AM

Split well-wishers off

Drop well-wishers

Subtract at Pax

Set pax to zero

Well-wishers leaving from gate process function

Process function pointer

A-2-99
GATE 3

SAVEVALUE
PH2+, PBS.XH

GATE 1

QUEUE
PHS

SAVEVALUE
GATXH+, PB13.XH

MARK

ENTER
PHS

DEPART
PHS

IN, M. PH

ASSIGN

GATE 4

F
TEST

GATE 5

AN

Congestion
Queue at gate
Passenger count at the gate
Mark pax for waiting
Service
Leave gate queue
Add waiting time to PH11
Braniff pax branch

A-2-100
Gate service time - Braniff

ADVANCE VSGAT3V

Gate 6

LEAVE PHS

Gate set when boarding begins

PH2.PHS.XH

Congestion

SAVEVALUE

GATE 2

TRANSFER CTRL8

Gate service time - Braniff

ADVANCE VSGAT4V

Gate 5

TRANSFER GATE 6

Pax waiting time at lounge to board

Gate 2

M SAVEVALUE

M.PHI.12.1.Ph

AO

A-2-101
START BOARDING OPERATIONS

GATE

ADVANCE V$GATIV

Advance to boarding time

5,V$GATIV,PH

Assign logic switch number

ASSIGN

LOGIC 5

Gate boarding switch

ADVANCE XH$BDTXH

Boarding time

LOGIC R

Gate boarding switch

ASSIGN

5,MH1(PH1,9),PH

Gate number

ASSIGN

5,MH9(PH5,3),PH

Point number

AP
SAVEVALUE
PM5-.MHL(PH1,12)XH

AP

Remove pax boarding from point

TERMINATE
DEPLANING PAX-SELF

GROUND TRANSPORTATION (MISC)

HELPA
Location, mode, deplaning switch

FORTN.16,PB2,286,8

ADVANCE
Walking time

HELPA

GRTJ1
Spare, mode, enplaning switch

FORTN.16,0,PB6,1

SAVEVALUE
Count of vehicles on arriving road

SAVEVALUE
Enter car in parking lot

SAVEVALUE
Vehicle count in parking lot

TRANSFER
CTRL8

SAVEVALUE

ARX+,1,XH

SAVEVALUE

PBXH+,1,XH

SAVEVALUE

PLXH+,1,XH

TRANSFER
CTRL8
GREETERS

GRT#2
HELPA
FORTN,16,$,PB6,$

SAVEVALUE
ARDXH+,1,XH

SAVEVALUE
PFLXH+,1,XH

SAVEVALUE
PLIXH+,1,XH

TRANSFER
CTRLS

GRT#3
HELPA
FORTN,16,PH2,PH6,$

ADVANCE
XHSTRUXH

QUEUE
PH5

AQ
ENTER

DEPART

ADVANCE VSPARIV

LEAVE

SAVEVALUE PKLXH=1,XH

SAVEVALUE PKCXH=1,XR

ASSIGN 4,DCARL,PN

TRANSFER

CTRL1
PARKING (DEPL PAX - CARS)

1. QUEUE
   - Service

2. MARK
   - Mark pax for waiting time

3. ENTER
   - Add waiting time to PH 11

4. DEPART
   - Parking exit service time

5. LEAVE
   - Exit car from parking lot

6. PKLXH=1.XH

7. SAVEVALUE

8. CTRL$ TRANSFER
   - Savevalue
   - Count of vehicle on departing road

A-2-107
Return to control if not rent-a-car

Car rental agency select

Location, agency

Travel time to car rental area

Congestion

Wait for free agent

Service

A-2-108
AR

Add waiting time to PH1

ASSIGN

RCAR2

ADVANCE

Car rental processing time

VSRCA1V

LEAVE

PH5

Service

SAVEVALUE

PH2-, PB5, XH

Congestion

TRANSFER

CTRLS
Deplaning pax in ground transport who have already rented car.

NOTE: Current logic assumes pax picks up car at agency parking lot.

Diagram:

```
RCARS

HELPA
FORTM, 16, PH2, PB6, S, PB1P

ADVANCE
XHSTRVXH

SAVEVALUE
DROXH+, 1, XH

TRANSFER
CTRL#

Location, mode, deplaning
emplaning switch agency
Waiting time
Count of cars on departing
road

A-2-110
```
Enplaning Pax Rent-a-car

NOTE: Current logic assumes rental car is returned to a parking lot (general or agency lot). Processing, if any, is done in the terminal.

Mode, deplaning/enplaning switch, agency

Count of vehicles on arriving road
If current location is point zero, print current events chain.

Location, gate

Walking time to security

Fall through if have well-wishers

Fall through if leaving well-wisher at security

Split well-wishers off

Drop well-wishers
AS

\[ S, PB, PB \]

ASSIGN

\[ 13, 0, PB \]

ASSIGN

\[ 1, \text{WWSIF, PT} \]

ASSIGN

\[ 2, 1, PB \]

ASSIGN

TRANSFER

CTRL$

\text{Subtract out pax}

\text{Set pax to } 0

\text{Well-wisher leaving from security process function}

\text{Process function pointer}

\text{Congestion}

\text{Passenger count at security}

\text{A-2-113}
AT

ENTER PH5

DEPART PBS PH5

11+, MI, PH ASSIGN

SECU2 ADVANCE VSSEC1V

LEAVE PH5

SAVEVALUE PH2-, PB5, XH

SAVEVALUE 12+, VSSECFN, 1, PB5, MH

TRANSFER CTRLS

Service

Add waiting time to PH11

Security check time

Service

Congestion

No. of pax leaving security
CONCESSIONS

Location, flight, random no., clock, switch

Walking time to lobby concession

Congestion

Waste time at concession

Congestion

Location, flight, random no., clock, switch

Walking time to concourse concession

Congestion
ADVANCE PHS

SAVEVALUE
PZ2=PZ3,XH

TRANSFER

Waste time at concession

Congestion

CTRL#
TRANSFER FLIGHTS

XFLT\n
1. TOP{1, PB

ASSIGN

ASSIGN

2. 1. PB

ASSIGN

3. 1. PB

ASSIGN

Pax in party

Set pointer

Total in party

HELPA

FORTM, 10, 0, 0

Initialize transfer flight table, MHS

MHI(PHI, 1) L TEST

Fall through if at end of flight table

MHS(1, 1) E TEST

Fall through if no transfer flights

TERMINATE

A-2-117
Add flight to transfer table MHL(PHL,11)

Branch if no transfer pax for next flight

Delay until time to add to transfer flight table

MHL row no., flag for adding flight

Increment row no. for MHL

Test for end of table
Branch if arriving flight or no transfer pax

Branch if transfer flight table empty

First flight to delete

Delay before deleting flight from transfer table

Gate no.

Type of flight (DOM, COM, INT)

MH row no., flag for ticket counter

Skip concourse for TDPIF function
AX

SPLIT
MH1(PH1,11)

HELPA
FORM,16,PH1,1

XFLT4
1+,1,PH
ASSIGN

XFLT8
MH1(PH1,1)

L
TEST

TERMINATE

XFLT7

BV5XFLIB

E
TEST

TRANSFER

XFLT9

ADVANCE
V5XFL4V

TRANSFER

CTRL8

Create remaining transfer pax

Delete flight from table

Increment MH1 row no.

Test for end of table

Branch if arriving flight or no transfer pax

Stagger pax to security
Collects transactions causing selected errors. PHA will normally give clue to nature of problem.
Perform previous change card and read next change card

If no storage changes, do not split

Parent PHI set to -1

Split 1 transaction to change each storage sequenced in PHI

Wait until next change

If wait was $, wait till NH7 used
2, PH1, PH
ASSIGN

SAVEVALUE
SAVEX, PH1, XH

2+, 30, PH
ASSIGN

1, MH7(PH1, 1)
ASSIGN

2, MH7(PH2, 1), PH
ASSIGN

MSAVEVALUE
7, 7, XH@SAVXH, 1, 0, MH

SAVEVALUE
SAVEX-, 30, XH

MSAVEVALUE
7, XH@SAVXH, 1, 0, MH

AY

Save MH7 row pointer

PM2 points to second portion of MH7

Change PH1 from MH7 row reference to storage subscript (set in FORM)

Change PH2 to required storage capacity (set in FORM)

Zero out MH7
Count down of transactions using MH7

Fall through if no lowering needed

If new capacity higher...

Allow transactions waiting on delay chain to start moving

Wait until someone leaves

Push storage capacity down to current contents
Storage lowering complete

Reset storage lowering completion flag
Initial call to FORTRAN help block to: read flight schedule and geometry data, initialize matrix base address, execute CLINK and MLINK.

- Needed to define "CLINK" halfword
- Set "clock" increment
- Complete linking and read in input data
- Was error detected in FORTRAN program
- Stop simulation
GATE Set if error count over limit

STOP1

SPLIT

GATE LS

SPLIT

TERMINATE 100

SPLIT

ADVANCE VSINCR

HELPA FORMT.31.XSINCR

ADVANCE XSINCRH

TRANSFER

STOP1

Clock increment

Clock update

Clock increment
Loop counter

F,11 ML1 with random numbers

Allows clock to occur first

Advance to end of run

Make formatted reports

Print selected output: Halfword matrices 1 to 5
Halfword matrices, 7 to 11

Clock time

Block counts'

Storage statistics

Queue statistics

User chain statistics

A-2-129
Table statistics

Fullword savevalues

Halfword savevalues

Byte savevalues

Floating point savevalues

Logic switches
GENERATE 300,

HELPF
FORTM.XHSTOT.XHSTEXP.XH$TIXG.XH$STOM

ADVANCE 300

TRANSFER

Generate transaction for snapshot reports

Call snapshot section in FORTM

Five minute wait until next snapshot
GENERATE 11000.4

TERMINATE 1

Timer for changing hourly table for pax waiting time

GENERATE 1..8

SAVEVALUE PXTBN+,1.XX

ADVANCE 3600

TRANSFER

For reset

Generate timer transaction

Increment number of table

Wait one hour

A-2-132
### AUXILIARY PROGRAM

This program generates transactions representing enplaning passenger groups. These transactions are stored on a 'jobtape'.

**RMULT** 743.31, 743.31

**SIMULATE**

**LOAD** FORM

**UNLIST** ABS, used to list the program

### HALFWORD MATRICES

**1**

- **MATRIX** MH,340.16 Flight schedule
- **1 ROW PER FLIGHT PLUS 1 ROW TO TERMINATE DEPL PAX GENERATION**
- **(PAX XAC PHI POINTS TO MH1 ROW)**
- **COL 1** - Arriving or departing flight (0 or 1; -1 --> END OF TABLE)
- **2** - Flight no
- **3** - Airline
- **4** - SCHEDULED ARRIVAL/DEPARTURE TIME
- **5** - DEPL. PAX BEING MET BY PRIVATE CAR
- **6** - TIME FROM START (MINUTES)
- **7** - Domestic, commuter or international flt (1, 2, 3)
- **8** - Aircraft type
- **9** - Gate no
- **10** - Originating/terminating passengers
- **11** - Transfer passengers
- **12** - Bag claim area (ARV FLT)
- **13** - Pax waiting at lounge to board (DEPT FLT)
- **14** - Transit passengers
- **15** - Total terminating bags (ARV FLT)
- **16** - Total transfer bags (ARV FLT)

**2**

- **MATRIX** ML,3.10 Ground transaction modal choice %
- **ROW 1** - Domestic
- **2** - Commuter
- **3** - International
- **COL 1** - Private car (pickup, drop off)
- **2** - Drives self (parking)
- **3** - Rental car
- **4** - Bus/limousine
- **5** - Taxi
- **6** - 10 **SPARE**
* FUNCTIONS

ARV1F FUNCTION RN7.C6 ARV TIME PRIOR TO FLT (0905-0400) 000055000
0.0/0.115,318/0.36,636/0.90,744/1.0,8400 000056000

ARV2F FUNCTION RN7.C6 ARV TIME PRIOR TO FLT (0405-0900) 000057000
0.0/0.02,720/0.16,168/0.42,276/0.93,408/1.0,4500 000058000

TOSIF FUNCTION RN7.C6 ARR TIME (TRANSFERS FROM OUT OF SYSTEM) 000059000
0.0/1.0,1200 000060000

PPPEF FUNCTION RN7.06 PAX/PARTY ENPLANING 000061000
0.32,1/0.71,2/0.91,3/0.97,4/0.99,5/1.0,6 000062000

WWPF FUNCTION RN6.04 WELL-WISHERS/PARTY ENPLANING 000063000
0.9,0/0.96,1/0.99,2/1.0,3 000064000

* VARIABLES

FLTIV VARIABLE 60+61(PH1,6)-8700-300 DELAY TO 150 MIN. BEFORE DEPT 000065000

FLT2V VARIABLE 60+61(PH1,6)-4800-300 DELAY TO 85 MIN. BEFORE DEPT 000066000

TOSIV VARIABLE 60-MHI(PH1,6)-2400 DELAY TO 40 MIN. BEFORE DEPT 000067000

TOSMV VARIABLE MH1(PH1,6)-1 CREATE T.O.O.S. TRANSACTIONS 000068000

SPASS ENGINE TRANSACTION PARAMETERS 000069000

HALFWORD: 1 - FLIGHT TABLE ROW NUMBER 00006A000
2 - LOCATION (POINT NUMBER) 00006B000
3 - MAX. BAG RANDOM NUMBER 00006C000
4 - ADDRESS PARAMETER 00006D000
5 - ** SCRATCH ** 00006E000
6 - USER CHAIN FOR BAG CLAIM (DEPL. PAX ONLY) 00006F000
7 - MH9 ROW FOR LAST FACILITY (NOT IMPLEMENTED IN EXIT LOGIC) 000070000
8 - ** SCRATCH ** 000071000
9 - ** SCRATCH ** FOR RETURN FOR ADDR IN XFER PAX LOGIC 000072000
10 - CUMULATIVE WALKING TIME 000073000

BYTE: 1 - PROCESS FUNCTION NUMBER 000074000
2 - PROCESS FUNCTION POINTER 000075000
3 - TYPE OF FLIGHT (1=DOM, 2=COM, 3=INT) 000076000
4 - NUMBER OF BAGS 000077000
5 - NUMBER IN PARTY (VISITORS + SELF) 000078000
6 - MODE OF GROUND TRANSPORTATION (SEE MATRIX ML2 FOR CODES) 000079000
7 - 0=DEPLANING, 1=ENPLANING 00007A000
8 - 1 = TERMINATING, 2 = TRANSFER 00007B000
9 - 3 = TRANSIT, 4 = TRANSFER OUT OF SYSTEM 00007C000
10 - USED IN DEPLANING FOR BAG FUNCTION 00007D000

ASSIGN 2,1,PB INIT PROCESS FN POINTER 00007E000

A-3-4
ASSIGN 7,1,PB
ENPL0 ASSIGN 1+,1,PH
TEST L MH1(PHI,1),0,++2
TERMINATE TEST E MH1(PHI,1),1,ENPL0
SPLIT 1,ENPL0
ASSIGN 3,MH1(PHI,7),PB
ASSIGN 5,MH1(PHI,9),PH
SPLIT 1,ENPL3
TEST LE MH1(PHI,16),0,++2
TERMINATE ASSIGN 8,4,PB
ASSIGN 13,1,PB
ASSIGN 5,1,PB
ADVANCE VSTOS1V
SPLIT VSTOS2V,++1
ADVANCE FNSTOS1F
TRANSFER WRITE
ENPL3 TEST E MH1(PHI,10),0,++2
MSAVEVALUE 1,PH1,10,1,MH
TEST L MH1(PHI,4),905,ENPL1
ADVANCE VSTOS1V
ASSIGN 8,MH1(PHI,10),PH
ASSIGN 13,FNSPPPPEF,PB
ASSIGN 8,PB13,PH
TEST LE PB9,0,++5
ASSIGN 13,*,PB8,PB
TEST LE PB13,0,++2
ASSIGN 13,1,PB
TRANSFER ++2
SPLIT 1,++7
ADVANCE FNSARV2F
TRANSFER ,ENPL2
ENPL1 ADVANCE VSTOS1V
ASSIGN 8,MH1(PHI,10),PH
ASSIGN 13,FNSPPPPEF,PB
ASSIGN 8-,PB13,PH
TEST LE PB9,0,++5
ASSIGN 13,*,PB8,PB
TEST LE PB13,0,++2
ASSIGN 13,1,PB
TRANSFER ++2
SPLIT 1,++7
ADVANCE FNSARV1F
ENPL2 HELP A FORMT,5,RN4,RN5,PB3
ASSIGN 5,PB13,PB
TEST LE PB9,2,++2
ASSIGN 5+,FNS$PPPPEF,PB
WRITE WRITE JOBTAPE1
TERMINATE
* CREATE DUMMY AAC FOR JOBTAPE
* GENERATE ...1,127,1PH
HELP C LINK,1
HELP FORMT,1,1
TEST G PH1,0,++2
TERMINATE PH1
SPLIT 1,WRITE
ADVANCE 100000
SPLIT 1,WRITE

MARK AS ENPLANING PAX
FLT TABLE (MH1) ROW SUBSCRIPT
MH1(PHI,1)++1 --> TABLE END (++2)
SKIP ARRIVING FLIGHTS
TYPE FLT (1,2,3 = DOM.COM.INT)
GATE NO
TO ORIGINATING PAX LOGIC
FT IF NO T.O.O.S. (++2)
MARK AS TRANSFER-OUT-OF-SYSTEM
ALL TRANSFERS SINGLE PAX
...WITH NO VISITORS
DELAY TO 40 MIN. BEFORE DEPT
FT IF PB13 LE ZERO
SET PB13 TO ONE
OUT OF LOOP (++2)
GO BACK TO CREATE ANOTHER (++-7)
ARR TIME PRIOR TO FLIGHT
0405-0904 DEPARTURES
ORIGINATING PAX
# OF PAX IN PARTY
SUBTRACT FROM TOTAL
FT IF TOTAL <= 0
ADJUST LAST PARTY SIZE
FT IF PB13 LE ZERO
SET PB13 TO ONE
OUT OF LOOP (++2)
GO BACK TO CREATE ANOTHER (++-7)
PREVENT JOBTAPE END MESSAGE
0405-0904 DEPARTURES
ORIGINATING PAX
# OF PAX IN PARTY
SUBTRACT FROM TOTAL
FT IF TOTAL <= 0
ADJUST LAST PARTY SIZE
FT IF PB13 LE ZERO
SET PB13 TO ONE
OUT OF LOOP (++2)
GO BACK TO CREATE ANOTHER (++-7)
PREVENT JOBTAPE END MESSAGE
0405-0904 DEPARTURES
ORIGINATING PAX
# OF PAX IN PARTY
SUBTRACT FROM TOTAL
FT IF TOTAL <= 0
ADJUST LAST PARTY SIZE
FT IF PB13 LE ZERO
SET PB13 TO ONE
OUT OF LOOP (++2)
GO BACK TO CREATE ANOTHER (++-7)
PREVENT JOBTAPE END MESSAGE
0405-0904 DEPARTURES
ORIGINATING PAX
# OF PAX IN PARTY
SUBTRACT FROM TOTAL
FT IF TOTAL <= 0
ADJUST LAST PARTY SIZE
FT IF PB13 LE ZERO
SET PB13 TO ONE
OUT OF LOOP (++2)
GO BACK TO CREATE ANOTHER (++-7)
PREVENT JOBTAPE END MESSAGE
0405-0904 DEPARTURES
ORIGINATING PAX
# OF PAX IN PARTY
SUBTRACT FROM TOTAL
FT IF TOTAL <= 0
ADJUST LAST PARTY SIZE
FT IF PB13 LE ZERO
SET PB13 TO ONE
OUT OF LOOP (++2)
GO BACK TO CREATE ANOTHER (++-7)
PREVENT JOBTAPE END MESSAGE
ADVANCE 1
TERMINATE 1

* 1 FUNCTION PHI,L4 MNEMONIC LINK FUNCTION - SEE FORTRAN CALL
   ,CMH01/,CMH04/,CML02/,CLKXH
   NOXREF
   START 1...1
   NOXREF
   REPORT
   NOXREF

FUNCTION PH1,L4

HMS TITLE 1.FLIGHT SCHEDULE

HMS TITLE 4.%ENPLANING PAX TICKETED

LMS TITLE 2.GROUND TRANSACTION MODAL CHOICE %

* OUTPUT REMOVE * TO PROVIDE ALL STATICS

END

0.468,1/0.833,2/0.917,3/0.987,4/0.987.5/1.0,8
ASSIGN 1,DDP1F,PS
ASSIGN 1,DDP1F,PS

A-3-6
REALLOCATE HWS, 13, BLO, 900, FAC, 0, QUE, 212, TAB, 14, FUN, 63, VAR, 67
REALLOCATE FSV, 25, CHA, 80, PMS, 0, LMS, 2, STO, 212, LOG, 185, BMS, 0
REALLOCATE LSV, 10, BSV, 10, HSV, 170, BVR, 11, XAC, 3600, COM, 220000

***************

MAIN PROGRAM

***************

THIS PROGRAM ENACTS THE MOVEMENT OF PASSENGERS AND VISITORS
THOUGH THE TERMINAL AREA.  THIS PROGRAM GENERATES DEPLANING PASSENGER, MERGING THEM WITH THE ENPLANING PASSENGER FROM THE AUXILIARY PROGRAM.  THE FLOW OF PASSENGER AND VISITORS IS MODELLED THROUGH A SEQUENCE OF SIMULATED LANDSIDE PROCESSING FACILITIES.

RMULT ...

UNLIST ABS USED TO LIST PROGRAM

LOAD FORTM, BAGS, DAG08, XACNO

JOBTAPE JOBTAI, ENPL0

* CHA10 SYN 1
* CHA1B EQU 1(54), L, C
* CHA20 SYN 55
* CHA2B EQU 55 L, C
* CHKQS SYN 1
* CHKQS EQU 1(18), Q, S
* CUSQS EQU 204
* CUSQS EQU 204, Q, S
* CMH01 SYN 16
* CMH01 SYN 64
* CMH02 SYN 3
* CMH02 SYN 64
* CMH03 SYN 4
* CMH03 SYN 64
* CMH04 SYN 2
* CMH04 SYN 64
* CMH06 SYN 110
* CMH06 SYN 64
* CMH07 SYN 1
* CMH07 SYN 64
* CMH08 SYN 2
* CMH08 SYN 64
* CMH09 SYN 6
* CMH09 SYN 64
* CML02 SYN 10

* CURB, DP, AND QUEUE MUST ALL HAVE SAME NUMBER OF ENTITIES, AND
* ENPL CURB ENTITIES MUST IMMEDIATELY FOLLOW DEPL CURB ENTITIES
* INPUT DATA CARDS MUST SHOW A FACILITY FOR ALL ENPLANING AND
* DEPLANING CURB STOREAGES ALLOCATED.  I. E., THE NUMBER OF FACNO'S OF
* ENPLANING AND DEPLANING CURBS MUST MATCH THE NUMBERS IN PARENTHESES
* OF THESE CURB STOREAGES.

* DPCBS SYN 44
* DPCBS EQU 44(4), S
* DPCBS EQU 48
* DPCBS EQU 48(4), S
* DPCBS EQU 52
* DPCBS EQU 52(4), S
* DPCBS EQU 56
* DPCBS EQU 56(4), S
* DPCBS EQU 60
* DPCBS EQU 60(4), C, S
* DPCBS EQU 64
* DPCBS EQU 64(4), C, S
* GAQSL SYN 80
* GAQSL EQU 80(105), Q, S, L
* DEPHX EQU 111, XH
* DEPHX EQU 112, XH
* DEPHX EQU 113, XH
* DEPHX EQU 114, XH
* PLOH EQU 115, XH

A-3-7
<table>
<thead>
<tr>
<th>Matrix</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MH,330,16</td>
<td>Flight Schedule</td>
</tr>
<tr>
<td>2</td>
<td>MH,15,3</td>
<td>Airline Information Table</td>
</tr>
<tr>
<td>3</td>
<td>MH,110,4</td>
<td>Table of Points *** Size ***</td>
</tr>
</tbody>
</table>

**Matrix Description**

- **1** Matrix: MH,330,16 Flight Schedule
  - 1 row per flight plus 1 row to terminate depl pax generation
  - (Pax xac phi points to mh1 row)
  - **Col 1**: -1 --> End of table
  - 0 --> Arriving flight
  - 1 --> Departing flight
  - 2 --> Indicates all bags from arriving flt at bag claim
  - 2 - Flight no
  - 3 - Air line
  - 4 - Scheduled Arrival/Departure time
  - 5 - Depl. Pax being met by private car
  - 6 - Time from start (Minutes)
  - 7 - Domestic, Commuter or international flt (1,2,3)
  - 8 - Aircraft type
  - 9 - Gate no
  - 10 - Originating/Terminating Passengers
  - 11 - Transfer Passengers
  - 12 - Bag claim area (Arrv flt)
  - 13 - Pax waiting at lounge to board (Dept flt)
  - 14 - Total terminating bags (Arrv flt)
  - 15 - Total transfer bags (Arrv flt)
  - 16 - Transfers out of system

- **2** Matrix: MH,15,3 Airline Information Table
  - Size to requirements

- **3** Matrix: MH,110,4 Table of Points *** Size ***
<table>
<thead>
<tr>
<th>Matrix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH.3,2</td>
<td>Percent enplaning Pax ticketed/direct</td>
</tr>
<tr>
<td>MH.101,1</td>
<td>Transfer flight table</td>
</tr>
<tr>
<td>MH.110,110</td>
<td>Walking time between points</td>
</tr>
<tr>
<td>MH.64,1</td>
<td>Used as work area by bag claim routines</td>
</tr>
<tr>
<td>MH.20,2</td>
<td>Used to access facility data in MHB</td>
</tr>
<tr>
<td>MH.230.6</td>
<td>Facility table; size to needs</td>
</tr>
<tr>
<td>ML.1,127</td>
<td>Reusable random number table</td>
</tr>
<tr>
<td>ML.3,10</td>
<td>Ground transaction modal choice</td>
</tr>
</tbody>
</table>

### Matrix MH.3,2
- **Col 1**: % Preticketed
- **Col 2**: % of preticketed Pax who go direct to gates via security
- **Row 1**: Domestic
- **Row 2**: Commuter
- **Row 3**: International

### Matrix MH.101,1
- Contains MHB row of departing flights taking Xfer Pax

### Matrix MH.110,110
- 1 row for each possible random number 1-64 generated by "bags" |

### Matrix MH.64,1
- Used as work area by bag claim routines |

### Matrix MH.20,2
- Used to access facility data in MHB |

### Matrix MH.230.6
- 1 row per actual or dummy facility |
- Col 1: Facility type (see MHB row description for codes) |
- Col 2: Facility number within type |
- Col 3: Location (point number) |
- Col 4-6: Used to identify facility with other model components |
- Bagclaim - 4: Deplcurb facno (arv flts) |
- Customs - 4: Deplcurb facno (arv flts) |
- Gate - 4: Security point facno (dep flts) |
- 5: Immigration area facno (arv flts) |
- Immigrat - 4: Associated customs facno (arv flts) |
- Rentacar - 4: Agency code (see fnsrcas2f) |
- 5: Parking facno (if other than 1, special lot) |
- 10: Parking |
- 11: Rentacar |
- 12: Deplcurb (deplaning curb) |
- 13: Immigration (health, etc) |
- 14: Tickets&Checkin |
- 15: 20 = spare |

### Matrix MH.7,1
- A counter for Pax leaving concourse |

### Matrix MH.7,1
- A counter for Pax leaving security |

### Matrix MH.15,1
- For airlines |

### Floating Matrices
- **Row 1**: Domestic |
- **Row 2**: Commuter |
- **Row 3**: International
<table>
<thead>
<tr>
<th>Function Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL 1 - PRIVATE CAR - CURB AND PARK PERCENTAGE</td>
<td>00149C00</td>
</tr>
<tr>
<td>2 - RENTAL CAR CUM. % PRIVATE</td>
<td>00150000</td>
</tr>
<tr>
<td>3 - BUS/LIMO CUM. % PRIVATE</td>
<td>00151C00</td>
</tr>
<tr>
<td>4 - TAXI CUM. % PRIVATE</td>
<td>00152C00</td>
</tr>
<tr>
<td>6 - 10 ** SPARE **</td>
<td>00153C00</td>
</tr>
</tbody>
</table>

**FUNCTIONS**

**DEPLANING PAX PROCESS FUNCTIONS**

- DEPLANING DOMESTIC PROCESS FN 1
- DEPLANING COMMUTER PROCESS FN 1
- DEPLANING INTERNATIONAL PROCESS FN 1
- DEPLANING LOBBY-BOUND PROCESS FN 1

**ENPLANING PAX PROCESS FUNCTIONS**

- DOMESTIC ENPLANING PAX - PROCESS FN 1
- COMMUTER ENPLANING PAX - PROCESS FN 1
- INTERNATIONAL ENPLANING PAX - PROCESS FN 1

**TRANSFER PAX PROCESS FUNCTIONS**

- INTERNATIONAL TRANSFER PAX - PROCESS FN 1
- DOMESTIC/COMMUTE TRANSFER PAX - PROCESS FN 2
- DOMESTIC/COMMUTE TRANSFER PAX - PROCESS FN 3
- DOMESTIC/COMMUTE TRANSFER PAX - PROCESS FN 4
- DOMESTIC/COMMUTE TRANSFER PAX - PROCESS FN 5

**FUNCTION SELECTOR**

- LONG-STAY/DIFF CONCOURSE
- LONG-STAY/SAME CONCOURSE
- SHORT-STAY/DIFF CONCOURSE
- TRANSIT PAX
- TRANSFER-OUT-OF-SYSTEM PROCESS FN (DEPL)
DWELL TIME FUNCTIONS

ENP2F FUNCTION RN7,C7 VEHICLE UNLOAD TIME – ENPLANING CURB 0.0,0.54.15/.83.45/.86.60/.94.90/1.150

ENP3F FUNCTION RN7,C7 EMPTY VEHICLE PARK TIME – ENPLANING CURB 0.0,0.2,30/.45/.60/.72.90/.96.150/1.0,180

OTHER FUNCTIONS

PPPDF FUNCTION VSRANDV,D6 PAX/PARTY – DEPLANING 0.,0/.2,30/.4,45/.54.60/.72.90/.96,150/1.0,180

GRPPF FUNCTION RN6,D5 GREETERS/PARTY (PARTIES W/GREETERS ONLY) 0.,0/.2,30/.4,45/.54.60/.72.90/.96,150/1.0,180

OCAIF FUNCTION RN7,C6 ARRIVAL DISTRIBUTIONS-CARS MEETING PAX 0.0,0/.575.1800/.66.2700/.75.3600/.81,450/1.0,6300

DCA2F FUNCTION RN7.C6 ARRIVAL DISTRIBUTIONS - GREETERS 0.0,0/.575.1800/.66.2700/.75.3600/.81,450/1.0,6300

DBAGF FUNCTION VSRANDV,D6 NO. OF SAGS - DOMESTIC FLIGHT 0.0,0/.2,30/.52.1/.79.2/.92.3/.96.4/.98,5/1.0,6

CBAGF FUNCTION VSRANDV,D7 NO. OF BAGS - COMMUTER FLIGHT 0.0,0/.2,30/.52.1/.79.2/.92.3/.96.4/.98,5/1.0,6

IBAGF FUNCTION VSRANDV,D9 NO. OF BAGS - INTERNATIONAL FLIGHT 0.0,0/.2,30/.57.0/.63.1/.83.2/.92.3/.96.4/.98,5/1.0,6

RCA2F FUNCTION RN7,D4 CAR RENTAL AGENCY SELECTION 0.0,0/.40.1/.75,2/.96,3/1.0,4

VARIABLES

SERVICE TIME VARIABLES

NOTE: WHEN PASSENGERS SCALED, MULTIPLY BY XHSSCLXH.

CHKV VARIABLE FNSCHK1F*XHSSCLXH EXPRESS CHECKIN

CK2V VARIABLE FNSCHK2F*XHSSCLXH FULL SERVICE TICKETING TIME

CUSIV VARIABLE FNSCUSIF*XHSSCLXH+PB13 CUSTOMS TIME

GAT3V VARIABLE FNSGAT3F*XHSSCLXH+PB13 GATE SERVICE TIME

IMM1V VARIABLE FNSIMM1F*XHSSCLXH+PB13 IMMIGRATION TIME

PARTY VARIABLE FNSFAR1F*XHSSCLXH PARKING LOT SERVICE TIME

RCAlV VARIABLE FNSPCA1F*XHSSCLXH CAR RENTAL CHECKOUT TIME

SCLVO VARIABLE XHSSCLXH DUMMY VAR – DEFINES XHSSCLXH

SECIV VARIABLE B*XHSSCLXH+PBS SECURITY SERVICE TIME (PARTY)

CIRCIV VARIABLE 300 RECIRCULATION 1TIME

OTHER VARIABLES

ACUNV VARIABLE VSACUV=3+90 MAX UNLOADING TIME – PAX

ACU1V VARIABLE MH1(PH1,10)+MH1(PH1,11)+MH1(PH1,12)+MH1(PH1,13)+MH1(PH1,16) TOTAL PAX ON AIRCRAFT

ACU2V VARIABLE XLSACUNL+RN6/1000+.5 STAGGER PAX OFF A/C

BUNAV VARIABLE MNH(4,2)+MNH(PH1,12) BAGCLAIM INDEX + AREA NO

DPLIV VARIABLE 30+PB4+45 DEP/CURB LOAD TIME CARS/TAXI

DPL2V VARIABLE RN3+SA+XFSMPX=60/1000000 RANDOM PULLOUT TIME <60 SEC

DPL3V VARIABLE PH10125 MOD FUNCTION

EMP1V VARIABLE FNSENP2F ENPL CURB VEHICLE DWELL TIME

0,VSCHK2V/9,VSCHK2V/10,VSCHK2V/11,VSCHK2V

A-3-12
ENPV VARIABLE 180 CURB LOAD/UNLOAD TIME - BUS/LIMD 00328000
FLTV VARIABLE 60*PHI(1,6)-AC1-3599 DELAY TO 1 HR BEFORE SCH ARR 00327000
FL2V VARIABLE MH(1,10)XLSGRTXL(1.1-XLSCGTXL)XLSPCBL 00328000
* NO. OF PARKING GREETING PROCEEDING TO CURB 00328100
* GHRTXL = GREETED, GTXL = GREETED AT CURB 00328200
* PCBL = GREETED PARK AND CURB 00328300
FL3V VARIABLE MH(1,10)XLSGRTXL(1.1-XLSCGTXL)XLSPCBL 00328400
* NO. OF GREETERS PROCEEDING TO PARKING FACILITY 00328500
* NO. OF GREETERS PROCEEDING TO CURB ONLY 00330010
* NO. OF TERMINATING PAX, SELF DRIVEN 00330110
FL6V VARIABLE MH(1,10)X(1.-M12(PB3,1)) 00328200
* NO. OF TERMINATING PAX USING OTHER MENCES 00330210
GATIV VARIABLE 3600-XHSIRXH START BOARDING PRIOR TO DEPARTURE 00311000
GATZ VARIABLE GA3SL=MH(1,9)-1 GATE BOARDING SWITCH 00332000
INCIV VARIABLE XHS'CNKH-1 ALLOW FOR START TIME = 1 - TIMER 00333000
XFLIV VARIABLE 60*PHI(1,6)-AC1-XHS'SFAH DELAY TO ADD FLT TO XFRFLT 00334000
XFLIV VARIABLE 60*PHI(1,6)-AC1-XHS'SFAH DELAY TO OLT FLT FM XFRFLT 00335000
XFL3V VARIABLE XHS'SFAXH+MN7/2000 STAGGER XFER PAX TO SECURITY 00336000
XFL4V VARIABLE (60+MH(1,6)-AC1)/60 TIME UNTIL FLT (MIN) 00337000
DPCDV VARIABLE PH6-DPCBS2+DPDPs DEPLANING CURB-DUPLICATE PARK 00338000
DPCDV VARIABLE PH6-DPCBS2+DPDPs DEPLANING QUEUE-DIUPLE PARK 00339000
DPCDV VARIABLE PH6-DPCBS2+DPDPs DEPLANING QUEUE-DUPLICATE PARK 00340000
DPCDV VARIABLE PH6-DPCBS2+DPDPs DEPLANING DOUBLE PARK-QUEUE 00341000
* ABOVE VARIABLES USED FOR ENPLANING CURB ALSO 00342000
EPCDV VARIABLE PH6-EPCBS+EPQCS ENPLANING CURB-QUEUE 00343000
EPCDV VARIABLE PH6-EPCBS+EPQCS ENPLANING CURB-QUEUE 00344000
DENFV VARIABLE 300 DEPL CURB ENFORCEMENT EVERY 5 MIN00345000
DLIMV VARIABLE PH6+300 DEPL CURB TIME LIMIT 5 MIN 00346000
RANDV VARIABLE ML1(PB3,10) USED AS ARGUMENT FOR MLI 00347000
NHOV VARIABLE FNPBANDO+126.O+1.0 YIELDS RANDOM INDEX FOR MLI 00348000
TMODV VARIABLE AC1#32767 MOD FUNCTION FOR ABS CLOCK TIME 00349000
* VARIABLES AND BOOLEAN VAR. FOR ENPLANING AND DEPLANING VEHICLE DELAY 00349000
* 00349100
* 00349200
* 00349300
* 00349400
* 00349500
* 00349600
* 00349700
* 00349800
* 00349900
* 00350000
* 00350100
* 00350200
* 00350300
* 00350400
* 00350500
* 00350600
* 00350700
* 00350800
* 00350900
* 00351000
* 00351100
* 00351200
* 00351300
* 00351400
* 00351500
* 00351600
* 00351700
* 00351800
* 00351900
* 00352000
* 00352100
* 00352200
* 00352300
* 00352400
* 00352500
* 00352600
* 00352700
* 00352800
* 00352900
* 00353000
* 00353100
* 00353200
* 00353300
* 00353400
* 00353500
* 00353600
* 00353700
* 00353800
* 00353900
* 00354000
* 00354100
* 00354200
* 00354300
* 00354400
* 00354500
* 00354600
* 00354700
* 00354800
* 00354900
* 00355000
* 00355100
* 00355200
* 00355300
* 00355400
* 00355500
* 00355600
* 00355700
* 00355800
* 00355900
* 00356000
* 00356100
* 00356200
* 00356300
* 00356400
* 00356500
* 00356600
* 00356700
* 00356800
* 00356900
* 00357000
* 00357100
* 00357200
* 00357300
* 00357400
* 00357500
* 00357600
* 00357700
* 00357800
* 00357900
* 00358000
* 00358100
* 00358200
* 00358300
* 00358400
* 00358500
* 00358600
* 00358700
* 00358800
* 00358900
* 00359000
* 00359100
* 00359200
* 00359300
* 00359400
* 00359500
* 00359600
* 00359700
* 00359800
* 00359900
* 00360000
* 00360100
* 00360200
* 00360300
* 00360400
* 00360500
* 00360600
* 00360700
* 00360800
* 00360900
* 00361000
* 00361100
* 00361200
* 00361300
* 00361400
* 00361500
DDOUT VARIABLE PB1'E'2
ENDIN VARIABLE (PB1'E'6+PB1'E'27+PB1'E'26)
OPRIN VARIABLE PB1'E'28
SEGN VARIABLE PH5+1-SECQS
TCKTN VARIABLE PH5+1-TICOS
* BOOLEAN VARIABLES
  PH5+1-SECQS OBTAIN CONCOURSE NUMBER FROM PH5
  PH5+1-TICOS OBTAIN A/L NO FROM PH5
  PB1'E'28
  PB1'E'2

SECNN VARIABLE PH5+1-SECQS
  PH5+1-TICOS OBTAIN CONCOURSE NUMBER FROM PH5
  PH5+1-TICOS OBTAIN A/L NO FROM PH5

XFL1B VARIABLE (MH1(PH1,1)'E'1)+(MH1(PH1,1)'G'0) DEPT+AND+TPAX>0
  NOT ENPL .OR. NOT TICKETED .OR. NOT DIRECT
  NOT ENPL .OR. NOT TICKETED .OR. NOT DIRECT
  NOT ENPL .OR. NOT TICKETED .OR. NOT DIRECT
  PB1'E'28

OLIMB VARIABLE (VSTMODV'G'VSDLIMV)*(PSIO'G'I)

T A B L E S
HOUR1 TABLE PH11,100,100,20
HOUR2 TABLE PH11,100,100,20
HOUR3 TABLE PH11,100,100,20
HOUR4 TABLE PH11,100,100,20
HOUR5 TABLE PH11,100,100,20
HOUR6 TABLE PH11,100,100,20
HOUR7 TABLE PH11,100,100,20
HOUR8 TABLE PH11,100,100,20
HOUR9 TABLE PH11,100,100,20
HOUR10 TABLE PH11,100,100,20
HOUR11 TABLE PH11,100,100,20
HOUR12 TABLE PH11,100,100,20
PAXWT TABLE PH9,100,100,20

P A S S E N G E R  T R A N S A C T I O N  P A R A M E T E R S
  BYTE:
  1 - PROCESS FUNCTION NUMBER
  2 - PROCESS FUNCTION POINTER
  3 - TYPE OF FLIGHT (1=DOM, 2=COM, 3=INT)
  4 - NUMBER OF BAGS
  5 - NUMBER IN PARTY (VISITORS + SELF)
  6 - MODE OF GROUND TRANSPORTATION
    (SEE MATRIX ML2 FOR CODES)
  7 - 0=DEPLANING, 1=ENPLANING
  8 - 1 = TERMINATING, 2 = TRANSFER
  9 - BAG CLAIM AREA LOG SWITCH (DEPL. PAX ONLY)

  HALFWORD:
  1 - FLIGHT TABLE ROW NUMBER
  2 - LOCATION (POINT NUMBER)
  3 - MAX. BAG RANDOM NUMBER
  4 - ADDRESS PARAMETER
  5 - ** SCRATCH **
     INITIALLY CARRIES GATE NO FOR DEPL PAX
     FORTM USES A - RETURN FOR STD, QUE, NOS, ETC.
  6 - USER CHAIN FOR BAG CLAIM (DEPL. PAX ONLY)
     CAR STORAGE NO IN DEPLANING Curb Logic
  7 - MH9 ROW FOR LAST FACILITY
     (NOT IMPLEMENTED IN EXIT LOGIC)
  8 - ** SCRATCH **
     FORTM USES AS RETURN FOR ADDR IN XFER PAX LOGIC
  9 - CUMULATIVE WALKING TIME
  10 - XAC SEQUENCE NUMBER FOR DEPL PAX MEETING
  11 - CUMULATIVE PAX WAITING TIME

  A-3-14
...CONTROLS... USED TO ROUTE TRANSACTIONS
THIS SECTION ROUTES TRANSACTIONS FROM ONE LOGIC SECTION TO ANOTHER
ALSO CONTAINS DGTR* CODE = DEPLANING GROUND TRANSPORT.
*00400000
CTRL0 ASSIGN 4,FN=PB1,PH ADDRESS OF NEXT OPERATION 00402000
ASSIGN 2+1,PH INCREMENT PROCESS FN POINTER 00403000
CTRL1 TRANSFER ,PH4 00404000
HANDLES ROUTING OF DEPL PAX TO PARKING LOTS. 00405000
CTRL8 TRANSFER ,PARK0 00406000
THE FOLLOWING SECTIONS OF "CONTROL" HANDLE PAX IN GROUND TRANSPORT. 00407000
CGTR0 ASSIGN 4,FN=CTR1,F,PH ROUTE DEPL PAX BY GA TX MODE 00411000
CGTR1 ASSIGN 4,FN=CTR2,F,PH ROUTE ENPL PAX BY GA TX MODE 00412000
THIS SECTION OF "CONTROL" GENERATES THE BUS/LIMO SERVICE TO THE
ENPLANING & DEPLANING CURBS. THE CURRENT MODEL SIMULATES A SINGLE 00418000
STOP AT EACH, THOUGH THE ENPLANING CURB LOGIC OF THE FORTRAN ROUTINE 00419000
"FORTR" SUPPORTS A MULTIPLE STOP SIMULATION WHEN DESIRED. COMPLEX 00420000
BUS SCHEDULES CURRENTLY REQUIRE INDIVIDUAL PROGRAMMING. 00421000
GENERATE ...,1,2PH 00422000
ASSIGN 2,1,PH 00424000
SPLIT 1,CGTR3 00425000
TEST E XH5ABUXH,0,**2 FT IF NO ARV BUS SIMULATED 00426000
TERMINATE 00427000
CGTR2 ADVANCE XH5ABUXH INTERVAL BETWEEN ARRIVING BUSES 00429000
SPLIT 1,CGTR2 00429000
TRANSFER ,DPLC8 00432000
TEST E XH5OBUXH,0,**2 FT IF NO DEP BUS SIMULATED 00433000
TERMINATE 00434000
CGTR4 ADVANCE XH5OBUXH INTERVAL BETWEEN DEPARTING BUSES 00435000
SPLIT 1,CGTR4 00436000
TRANSFER ,DPLC8 00437000
PASSENGER TERMINATION 00438000
DEP99 TABULATE PAXWT PASSENGER WALKING TIME 00439000
TABULATE XBSPXTBN PASSENGER WAITING TIME 00439100
TERMINATE 00439200
ENP99 TABULATE PAXWT PASSENGER WALKING TIME 00440000
TABULATE XBSPXTBN PASSENGER WAITING TIME 00440100
TERMINATE 00440200
TRX99 TABULATE PAXWT PASSENGER WALKING TIME 00441000
TABULATE XBSPXTBN PASSENGER WAITING TIME 00441100

A-3-15
TERMINATE

END OF CONTROL

DEPLANING PASSENGER LOGIC

GENERATE 1, 10, 11PH, 14PB
ASSIGN 2, PB
SPLIT 1, XPLT0
ADVANCE 1
ASSIGN 11, PB
DEPLAN ASSIGN 11, PH, PB
TEST L MH1(PHI, 1), 0, 0, ++2
TERMIMATE
ADVANCE XPLT11
SPLIT 1, DEPLO
TEST E MH1(PHI, 1), DEPL0
ASSIGN 4, GATE9, PH
TRANSFER .CTRL1
DEPLB ASSIGN 3, MH1(PHI, 1), PB
TYPE FLT (1, 2, 3 = DOM, COM, INT)

CREATE VISITORS HERE

TEST E PB3, 1, DEPL1
ASSIGN 1, DEPL1F, PB
ASSIGN 14, DBAGF, PB
DEPL1 TEST E PB2, 2, DEPL2
ASSIGN 1, DEPL1F, PB
ASSIGN 14, CBAGF, PB
DEPL2 ASSIGN 1, DEPL1F, PB
ASSIGN 14, LBAGF, PB
DEPL3 TEST GE XHSEEHM32000, ++2
SAVEVALUE SEQH, 0, XH
SPLIT 1, DEP25
SPLIT 1, DEP15
SPLIT 1, DEP17
TRANSFER .DEPL7

Curb with Greeters Logic

DEP25 ASSIGN 6, 1, PB
ASSIGN 8, XPLT2V, PH
TRANSFER .DEP16

Park with Greeters Logic

DEP15 ASSIGN 6, 2, PB
ASSIGN 8, XPLT2V, PH
DEP16 TEST E PH8, 0, ++2
TERMIMATE
MSAVEVALUE 1+, PH1, 1, PH8, PH
ASSIGN 10, VSPL3V, PB
DEP12 ASSIGN 10, 1, PB
TEST G PB10, 124, ++2
ASSIGN 10, PB
ASSIGN 8, FNSPPDF, PH
ASSIGN 10, 1, PB
ASSIGN 4, FPBP14, PB

Transfer Curb with Greeters Logic

DEP25 ASSIGN 6, 1, PB
ASSIGN 8, XPLT2V, PH
TRANSFER .DEP16

Park with Greeters Logic

DEP15 ASSIGN 6, 2, PB
ASSIGN 8, XPLT2V, PH
DEP16 TEST E PH8, 0, ++2
TERMIMATE
MSAVEVALUE 1+, PH1, 1, PH8, PH
ASSIGN 10, VSPL3V, PB
DEP12 ASSIGN 10, 1, PB
TEST G PB10, 124, ++2
ASSIGN 10, PB
ASSIGN 8, FNSPPDF, PH
ASSIGN 10, 1, PB
ASSIGN 4, FPBP14, PB

A-3-16
ASSIGN 10+.1.PB  INCR POINTER 00502000
TEST LE  ML111.PB10),XLSGRGKL.+4 FT FOR GATE MEETING (**4) 00503000
ASSIGN 1,GRGF.PB  MEET AT GATE 00504000
ASSIGN 12,.1.PB 00505000
TRANSFER .DEP14 00506000
TEST E  PB4,.0.+3  FT FOR NO BAGS (**3) 00507000
ASSIGN 1,GRGF.PB  MEET IN LOBBY 00508000
TRANSFER .DEP14 00509000
ASSIGN 1,GRGF.PB  MEET IN BAG CLAIM 00510000
DEP14 TEST G  PB4,.0.+2  FT IF NEED MORE PARTIES (**2) 00511000
SPLIT 1,DEP12 00512000
ASSIGN 14,.0.PB  RESET PB14 FOR LATER USE # OF GREETERS 00513000
ASSIGN 5,FSNSGPPP,PB  INCR XAC COUNTER 00514000
SAVEVALUE XE01H+1.XH  XAC SEQUENCE # 00515000
ADVANCE FSNSCDA2F TO AIRPORT 00516000
TRANSFER ,CTRL0 00517000

* CURB WITH GREETERS LOGIC (MEET AT CURB) 00518000
* 00519000
DEP17 ASSIGN 6,.1.PB  MODE = CURB 00520000
ASSIGN 8,V5FT4V.PH  NO. OF PAX TO BE MET 00521000
TEST E  PHB,.0.+2  FT IF NONE 00522000
TERMINATE 00523000
SAVEVALUE 15+,.PH1,.5,PB,WH  ADD TO TOTAL MET 00524000
ASSIGN 10,V5DPL3V.PB  STARTING POINT IN R.N. TABLE 00525000
ASSIGN 4,DCARO.PH  WILL GO TO CURB 00526000
DEP13 TEST G  PB10,.12+.+2  INCREMENT R.N. TABLE POINTER (**2) 00527000
ASSIGN 10,.1.PB  TURN CORNER 00528000
ASSIGN 8-,FNPPDPF,PB  SUBTRACT # OF PAX FROM TOTAL 00529000
ASSIGN 4,FSNPB14,PB  INCR POINTER 00530000
ASSIGN 10,.1.PB  COMPUTE # OF BAGS 00531000
ASSIGN 4,FN=PB14,PB  FT IF NEED MORE PARTIES (**2) 00532000
SPLIT 1,DEP13 00533000
ASSIGN 14,.0.PB  RESET PB14 FOR LATER USE NOT USED UNLESS RECIRC&PARK 00534000
ASSIGN 5,FSNSGPPP,PB  INCR XAC COUNTER 00535000
SAVEVALUE XE01H+1.XH  XAC SEQUENCE # 00536000
ASSIGN 10,XHSSEQ1H,PB  TO AIRPORT 00537000
TRANSFER ,CTRL1 00538000

* NOTE: ADVANCE TIME MUST BE DETERMINISTIC 00539000
DEP17 ADVANCE 3600 00540000
GATE LR  DPL1G  GUARD "BAGS" FM SIMUL ARRY A/C 00541000
LOGIC S DLPLG 00542000
ASSIGN 5,MH1(PH1,.9),PH  GATE NO 00543000
ASSIGN 7,PHS.PH  LAST MH9 ROW 00544000
ASSIGN 2,MH9(PH,.3),PH  POINT NO OF GATE 00545000
SAVEVALUE ACUNL,VSACUNV,XL  MAX PASSENGER UNLOADING TIME 00546000
SELECT LR 6PH,CHAI8,CHA2B  PICK FREE BAG CLAIM CHAIN-SWITCH 00547000
LOGIC S PH6  MARK THIS CHAIN IN USE 00548000
SPLIT 1,DEPL4  COPY TO TERMINATING PAX 00549000
SPLIT 1,DEPL6  COPY TO TRANSIT PAX 00550000
SPLIT 1,DEP10  COPY TO TRANSFER-OUT-OF-SYSTEM 00551000
TEST E  MH91110,DEP18  FALL THRU IF NO DEP FLTS IN XFER 00552000
SAVEVALUE XFRXH+MHI(PH1,.11),XH  ADD XFER PAX TO "HOLDING MH" 00553000
DEP18 TEST E  MH1(PH1,.11),0,+2  FT IF NO PAX (**2) 00554000
TERMINATE 00555000
ASSIGN 8.2,PB  MARK AS TRANSFER PAX 00556000
ASSIGN 13.1,PB  PAX IN PARTY 00557000

A-3-17
* MODES OTHER THAN CURB AND PARK

ASSIGN 6.2.PB  ASSIGN MODE
ASSIGN 8.VSFL7V.PH NO. OF PAX IN MODES OTHER THAN CURB/PARK
TEST E PHB.0.+2  FT IF NONE
TERMINATE

DEPL9 ASSIGN 10.+1.PB  STARTING POINT IN R.N. TABLE
ASSIGN 10.1.PB  INCREMENT R.N. TABLE POINTER
ASSIGN 10.1.PB  TURN CORNER
ASSIGN 8.PB13.PH  PXA IN PARTY
ASSIGN 8.PB13.PH  SUBTRACT FROM TOTAL
ASSIGN 10+1.PH9.PB  ADJUST LAST PARTY SIZE
TRANSFER ++2  OUT OF LOOP
SPLIT 1.DEPL9  GO BACK TO CREATE ANOTHER
HELP 10.PB13.PB  TOTAL IN PARTY
HELP 10.PB13.PB  INCR POINTER
HELP 10.PB13.PB  GEN # OF BAGS, MAX RN
ASSIGN 14.0.PB  RESET PB14 FOR LATER USE
ASSIGN 13+1.PH9.PB  CHANGE TO LOBBY PROC FN
TRANSFER ,DEP24

* CURB WITH GREETERS

DEP18 ASSIGN 6.1.PB  MODE = CURB
ASSIGN 8.VSFL72V.PH NO. OF PARKING GREETERS PROCEEDING TO CURB
TRANSFER ,DEP20

* PARK WITH GREETERS

DEP19 ASSIGN 6.2.PB  MODE = PARK
ASSIGN 8.VSFL73V.PH NO. OF PAX TO BE GREETED
DEP20 TEST E PHB.0.+2  FT IF NONE
TERMINATE
ASSIGN 10.VSFL7V.PB  STARTING POINT IN R.N. TABLE
DEP21 ASSIGN 10.+1.PB  INCREMENT R.N. POINTER
TEST G PB10.124.+2  PREVENT POINTER FROM EXCEEDING 127(=)200675C00
ASSIGN 10.1.PB  RESET POINTER
ASSIGN 13.PH9.PPD.PB  PXA IN PARTY
ASSIGN 8.PB13.PH  SUBTRACT GROUP FROM TOTAL
ASSIGN 12.2.PB  MEET AT BAGCLAIM
ASSIGN 10.+1.PB  INCREMENT R.N. POINTER
HELP 10.PB13.PB  INCREMENT R.N. POINTER
HELP 10.PB13.PB  INCREMENT R.N. POINTER
HELP 10.PB13.PB  INCREMENT R.N. POINTER
TEST LE MLI(1,PB10),XLSSGXL.PB OF GATE MEETING (=)200665C00
ASSIGN 12.1.PB  MEET AT GATE
ASSIGN 12.1.PB  MEET AT GATE
TEST LE PHB.0.+2  IS LAST PARTY SIZE TOO LARGE(=)200670C00
ASSIGN 13.PH9,PB  ADJUST LAST PARTY SIZE
TRANSFER ++2  (=)200672C00
SPLIT 1.DEP21
TEST E PB3.3.+4  FT IF NOT INTERNATIONAL
TEST E PB4.0.+3  FT IF NO BAGS
ASSIGN 1.DLP1F.PB  CHANGE TO LOBBY PROC FN
ASSIGN 12.3.PB  MEET AT TIXETING
ASSIGN 12.3.PB  MEET AT TIXETING
ASSIGN 5.PB13.PB  TOTAL IN PARTY
SAVEVALUE SEQ2H+,1,XH  INCR XAC COUNTER

A-3-19
ASSIGN 10.XHSSEQ2H,PH XAC SEQUENCE # 00677000
TRANSFER ,DEP24 00678000

* Curb without Greeters

DEP22 ASSIGN 6,1,PB MODE = CURB 00682000
ASSIGN 8,V$FLT4V,PH NO. OF PAX TO BE MET 00683000
TEST E PH8.0,++2 FT IF NONE 00694000
TERMINATE 10,V$DPL3V,PB STARTING POINT IN R.N. TABLE 00686000
DEP23 ASSIGN 10+1,PB INCREMENT R.N. POINTER 00687000
TEST G PB10.124,++2 PREVENT POINTER FROM EXCEEDING 127(++) 00688000
ASSIGN 10.1,PB RESET POINTER 00699000
ASSIGN 13,FNSPPPDF,PB PAX IN PARTY 00690000
ASSIGN 6-,PB13,PB SUBTRACT GROUP FROM TOTAL 00691000
ASSIGN 10+,1,PB INCREMENT R.N. POINTER 00692000
HELP BAGS,PH1,FN=PB14,4,3,PB8 GEN # OF BAGS. MAX RN 00693000
TEST LE PH8.0,++3 IS LAST PARTY SIZE TOO LARGE(++) 00696000
ASSIGN 13+,PH8,PB ADJUST LAST PARTY SIZE 00697000
TRANSFER ,++2 00698000
SPLIT 1,DEP23 00699000
ASSIGN 14,0,PB TOTAL IN PARTY 00700000
ASSIGN 5,PB13,PB TOTAL IN PARTY 00701000
SAVEVALUE SEQ2H+,1,XH INCR XAC COUNTER 00702000
ASSIGN 10,XHSSEQ2H,PH XAC SEQUENCE # 00703000
TRANSFER ,DEP24 00704000

* Park without Greeters

DEP26 ASSIGN 6,2,PB MODE = PARK 00706000
ASSIGN 8,V$FLT5V,PH NO. OF PAX TO BE CREATED 00707000
TEST E PH8.0,++2 FT IF NONE 00708000
TERMINATE 10,V$DPL3V,PB STARTING POINT IN R.N. TABLE 00711000
DEP27 ASSIGN 10+1,PB INCREMENT R.N. POINTER 00712000
TEST G PB10.124,++2 PREVENT POINTER FROM EXCEEDING 127(++) 00713000
ASSIGN 10.1,PB RESET POINTER 00714000
ASSIGN 13,FNSPPPDF,PB PAX IN PARTY 00715000
ASSIGN 6-,PB13,PB SUBTRACT GROUP FROM TOTAL 00716000
TEST LE PH8.0,++3 IS LAST PARTY SIZE TOO LARGE(++) 00717000
ASSIGN 13+,PH8,PB ADJUST LAST PARTY SIZE 00718000
TRANSFER ,++2 00719000
SPLIT 1,DEP27 00720000
ASSIGN 14,0,PB TOTAL IN PARTY 00721000
ASSIGN 5,PB13,PB TOTAL IN PARTY 00722000
HELP BAGS,PH1,FN=PB14,4,3,PB8 GEN # OF BAGS. MAX RN 00724000
ASSIGN 14,0,PB RESET PB14 FOR LATER USE 00725000
TEST E PB4.0,++2 FT IF NO BAGS 00726000
ASSIGN 1,DLP12,PH CHANGE TO LOBBY PROC FN 00727000

DEP24 ADVANCE V$ACU02V STAGGER PAX OFF AIRCRAFT 00728000
TEST NE PB12.1,++2 FT IF NOT MEETING AT GATE(++) 00730000
TRANSFER ,CTRL0 00731000
UNLINK GRECC,CTRL1,1,10PH,PH10,DEP2B TRY TO UNLINK GREETER 00732000
DEP29 GATE LS PAS3L WAIT FOR INFO FROM GREETER 00733000
ASSIGN 5+,X$BPAS32,PH ADD GREETERS TO PARTY 00734000
ASSIGN 14,X$BPAS33,PH PICK UP PARKING LOT NUMBER 00735000
LOGIC R PAS3L PERMIT NEXT USE OF PAS3 $VALUES 00736000
TRANSFER ,CTRL0 00737000
DEP20 ASSIGN 4,DEP29,PH WILL GO TO DEP29 WHEN UNLINKED 00738000
LINK GRECC,FIFO WAIT FOR GREETERS 00739000

A-3-20
**BAGC5** ASSIGN 4, BAGC6, PH  WILL GO TO BAGC6 WHEN UNLINKED 00010000
  LINK GREBC, FIFO 00020000
  • 00030000
  **BAGC4** TRANSFER , CTRL0 00040000
  • 00050000
  • 00060000
  **BAGGAGE UNLOADING LOGIC** 00070000
  • 00080000
  **BUNLO PRIORITY** 5, BUFFER  ALLOW PAX XAC TO EXECUTE "BAGS" 00090000
  SPLIT 1, ++2, 9PH, 40PB  (++2) 00010000
  TERMINATE 00011000
  HELPA FORMT, 2, 10  BAGGAGE UNLOADING LOGIC 00012000
  ASSIGN 5, MH1 (PH1, 8), PH  A/C TYPE 00013000
  ASSIGN 5, FNSBUN1F, PH  UNLOAD TIME FUN F(A/C TYPE) 00014000
  LOGIC R 00015000
  • 00016000
  ADVANCE FN+PHS  BAGGAGE UNLOADING TIME 00017000
  • 00018000
  • 00019000
  ASSIGN 5, V$BUNAV4, PH  BAGCLAIM INDEX + AREA NO 00021000
  ASSIGN 5, MH9 (PHS, 3), PH  POINT NO OF BAG CLAIM 00022000
  ASSIGN 4, 40, PH  LOOP 00023000
  BUNL2 TEST G PB=PH4, 0, BUNL3  BR IF ALL BAGS UNLOADED 00024000
  ADVANCE 30  LOOP COUNTER 00025000
  UNLINK LE PH6, CTRL1, ALL, 3PH, PB=PH4  BAGS TO WAITING PAX 00026000
  LOOP 4PH, BUNL2 00027000
  BUNL3 LOGIC R PH6  FREE CHAIN/SWITCH PAIR 00028000
  MSAVEVALUE 1, PH1, 1, 2, MH  MARK ALL BAGS AT CLAIM AREA 00029000
  TERMINATE 00030000
  • 00031000
  • 00032000
  **TICKETING & CHECKIN** 00033000
  • 00034000
  CHEKO TEST E BV$CHK1B, 1, CTRL0  BR IF ENPL, TICKETED AND DIRECT 00035000
  HELPA FORMT, 13, PH2, MH1 (PH1, 3), PB9, RN4, PB13 LOC, AIRLINE, TICKETED, EXPCHK RN, # PAX 00036000
  • 00037000
  ADVANCE XHSTRVXH 00038000
  TEST E PB13, 0, CHEK9  FT IF GREETER 00039000
  UNLINK GRELC, CTRL1, 1, 10PH, PH10, CHE10 TRY TO UNLINK PAX 00040000
  CHE11 GATE LR PAS2L  ONLY ONE AT A TIME 00041000
  SAVEVALUE PAS22, PB5, XB  NUMBER OF GREETERS 00042000
  SAVEVALUE PAS23, PB14, XB  PARKING LOT NUMBER 00043000
  LOGIC S PAS2L  LET PAX PICK UP INFO 00044000
  TEST NE PB6, 1, ++2  BR IF MODE IS CURB 00045000
  TERMINATE 00046000
  ASSIGN 5, 0, PB  NO-ONE IN PARTY 00047000
  ASSIGN 4, GRT03, PH  ROUTE TO GREETER LEAVING PARKING 00048000
  TRANSFER , CTRL1 00049000
  • 00050000
  • 00051000
  CHE10 ASSIGN 4, CHE11, PH  WILL GO TO CHE11 WHEN UNLINKED 00052000
  LINK GRELC, FIFO 00053000
  • 00054000
  CHE12 ASSIGN 4, CHE11, PH  WILL GO TO CHE11 WHEN UNLINKED 00051000
  LINK GRELC, FIFO 00052000
  • 00053000
  CHEK8 TEST E PB12, 3, CHEK8  FT IF TO BE GREETED AT TICKETING 00054000
  UNLINK GRELC, CTRL1, 1, 10PH, PH10, CHEK7 TRY TO UNLINK GREETER 00055000
  CHEK8 GATE LS PAS2L  WAIT FOR INFO FROM GREETER 00056000
  ASSIGN 5, KBSPA522, PB  ADD GREETERS TO PARTY 00057000
  ASSIGN 14, KBSPA523, PB  PICK UP PARKING LOT NUMBER 00058000
  LOGIC R PAS2L  PERMIT NEXT USE OF PAS2 S’VALUES 00059000
  TRANSFER , CTRL0 00060000
  • 00061000
  CHEK7 ASSIGN 4, CHEK8, PH  WILL GO TO CHEK8 WHEN UNLINKED 00062000
  LINK GRELC, FIFO 00063000

A-3-22
CHEK1 TEST NE PBB.1,CTRLO BR IF TERMINATING 00864000
SAVEVALUE PH2+.PBB,XH CONGESTION 00865000

CHEK1 QUEUE PH5,PBB WAIT FOR FREE SEVER 00866000
MARK PH5 MARK PAX FOR WAITING TIME 00867000
ENTER PH5 SERVICE 00868000
DEPART PH5,PBB ADD WAITING TIME TO PH1 00869000
ASSIGN 11+,M1,PH BR TO CHEK2 OR CHEK3;SET IN FORT. 00870000
TRANSFER ,PH4 00871000
CHEK2 ADVANCE VSCHK1V CHECKIN TIME FOR TICKETED PAX 00872000
LEAVE PH5 SERVICE 00873000
SAVEVALUE PH2-,PBB,XH CONGESTION 00874000
TEST E MH1(PH1.3),1,++3 00875000
SAVEVALUE EALQ+,PBB13,XH 00876000
TRANSFER ,CHE12 00877000
CHEK2 ADVANCE VSCHK1V 00878000
SAVEVALUE PH2-,PBB,XH CONGESTION 00879000
TEST E MH1(PH1.3),6,++2 00880000
SAVEVALUE UNTXH+,PBB13,XH 00881000
TRANSFER ,CHE12 00882000
CHEK3 ADVANCE CHKXH,MH1(PH1.3),XH 00883000
SAVEVALUE PH2-,PBB,XH CONGESTION 00884000
TEST E MH1(PH1.3),6,++3 00885000
SAVEVALUE AMAX+,PBB13,XH 00886000
TRANSFER ,CHE12 00887000
CHE12 TRANSFER ,CTRLO 00888000
CHEK2 ADVANCE VSCHK1V 00889000
SAVEVALUE PH2-,PBB,XH CONGESTION 00890000
TEST E MH1(PH1.3),6,++2 00891000
SAVEVALUE UNTXH+,PBB13,XH 00892000
TRANSFER ,CTRLO 00893000

...IMMIGRATION...

IMMI0 HELP A FORTN,8,PH2,MH1(PH1.9) PT. NO.,GATE NO. 00894000
ADVANCE XHSTRVXH 00895000
SAVEVALUE PH2+,PBB,XH CONGESTION 00896000
GATE SNF PH5,IMM1 00897000
ENTER PH5 SERVICE 00898000
TRANSFER ,IMM12 00899000
IMMI1 QUEUE PH5,PBB 0089A000
ENTER PH5 0089B000
DEPART PH5,PBB 0089C000
IMMI2 ADVANCE VS1IMM1V IMMIGRATION PROCESS TIME 0089D000
LEAVE PH5 0089E000
SAVEVALUE PH2-,PBB,XH CONGESTION 0089F000
SAVEVALUE IMIG+,PBB13,XH 008A0000
TRANSFER ,CTRLO 008A1000

...CUSTOMS...

CUST0 HELP A FORTN,4,PH2,PH8 LOC,M9 ROW 008A2000
ADVANCE XHSTRVXH 008A3000
SAVEVALUE PH2+,PBB,XH CONGESTION 008A4000
CUST3 GATE SNF PH5,CUST2 008A5000
ENTER PH5 008A6000
TRANSFER ,CUST4 008A7000
CUST2 QUEUE PH5,PBB 008A8000
ENTER PH5 008A9000
DEPART PH5,PBB 008AA000
CUST4 ADVANCE VSCUS1V CUSTOMS PROCESS TIME 008AB000
SAVEVALUE PH5 SERVICE 008AC000

A-3-23
SAVEVALUE PH2-,PB5, XH            CONGESTION          088260
SAVEVALUE CSTM+,PB13, XH          088220
TRANSFER ,CTRL0                        088270

...CONCOURSE EXIT - TERMINATING PAX     088230

CMCRO HELPA  FORTM,25,PH2, M4***(PH1,9)  LOC, GATE     088890
ADVANCE XHSTVRXH  WALKING TM TO CONCOURSE FROM GATE 088900
MSAVEVALUE 11+,PH5,1,PB5, XH NO OF PAX LEAVING CONCOURSE 088910
SAVEVALUE CONXH+,PBS, XH DUMMY TO KEEP PUNCT 1 FOR MNLINK HAPPY 088920
TRANSFER ,CTRL0                        088930

...DEPLANING CURB (CARS)     088940

DCARO ADVANCE FNSDCA1F ARRIVAL DISTRIBUTION FOR CARS 088950
SAVEVALUE ARDH+,1, XH COUNT OF VEH ON ARRIVING ROAD 088960

DCAR1 SAVEVALUE DEPXH+,1, XH 088970
HELPA  FORTM,10,MH1(PH1,3),PH1,PB4,0 AIRLINE,FLT, # OF BAGS 090000
TEST NE PB4,0,DCARO  FALL THROUGH IF DEPL 090010
ASSIGN 8,VSDB#K,PH 1 LESS THAN 1ST DEPL CURB STO NO 090010
ASSIGN 4,PHB, XH SAVE CURB STO NO MINUS 1 0900105
ASSIGN 5,VSEDPCDB,PH 1 LESS THAN 1ST DEPL CURB STO NO FOR DEPL 0900110
TEST NE PB10,4,DCARO IF CIR VEH, TRANSFER TO DCARO 0900120
ASSIGN 8,VSDDEPL,PH 1 LESS THAN CORRESPONDING STO NO 0900140
TRANSFER ,DCARO 0900150

DCARO ASSIGN 8,VSDB#K1,PH 1 LESS THAN ENPL CURB STO NO 0900155
ASSIGN 4,PHB,PH SAVE CURB STO NO MINUS 1 0900163
ASSIGN 5,VSEDPCDB,PH DBL PARKING STO NO FOR ENPL 0900165
TEST NE PB4,0,DCARO IF CIR VEH, TRANSFER TO DCARO 0900170
ASSIGN 8,VSEDENPL,PH 1 LESS THAN CORRESPONDING ENPL STO NO 0900190
DCARO ASSIGN 8+,1,PH INCREMENT STO NO 0900220
TEST NE PB10,4,**2 IF CIR VEH, SKIP NEXT TEST 0900230
TEST NE PB6,PHB,DCARO IF STO NO SAME AS DESTINATION 0900240
ASSIGN 4+,1,PH INCREMENT CURB STO NO 0900250
ASSIGN 5+,1,PH INCREMENT DBL PARKING STO NO 0900255
QUEUE PH4 ENTER QUEUE FOR BLOCKED LANE 0900257
TEST E BVSLNFD,0 STOP TRAFFIC IF DBL PARKING & NO OF PASSING 0900260
DEPART PH4 DEPART QUEUE FOR BLOCKED LANE 0900275
TEST E S-PH5,0,DCARO FT IF CURRENT CONTENTS OF DBL PARK 0900280
STO IS ZERO 0900290

SAVEVALUE NOLAN,VSXNLMX,XH SET NO OF PASSING LANES TO MAX Avail 0900310
SAVEVALUE TOTL1+,1, XH SAVE COUNT 0900310
TRANSFER ,**+ 0900240

DCARO SAVEVALUE NOLAN,VSXNLM1,XH SET NO OF PASSING LANES TO 1 LESS 0900230
SAVEVALUE TOTL1+,1, XH THAN MAX AVAILABLE 0900240
SAVEVALUE TOTL3+,VXNLNDY,XH SAVE TIME 0900250
ADVANCE VXNLNDY DELAY VEH ACCORDING TO NO OF LANES Avail 0900270
TEST NE PB4,0,**3 FALL THROUGH IF DEPL 0900275
TEST E BVSCIRVD,0,DCARO IF CIR VEH IS AT LAST 0900280
DEPL STO, TRANSFER TO DCARO 0900290
TRANSFER ,DCARO MOVE TO NEXT DEPL STO 0900290
TEST E BVSCIRVD,0,DCARO IF CIR VEH IS AT LAST 0900290
ENPL STO, TRANSFER TO DCARO 0900290
TRANSFER ,DCARO MOVE TO NEXT ENPL STO 0900290
DCARO SAVEVALUE ADPLC+,1, XH COUNT OF VEH. ON DEPL CURB 0900290

A-3-24
<table>
<thead>
<tr>
<th>Assignment</th>
<th>Description</th>
<th>Time Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN 8</td>
<td>VS1MOV,PH</td>
<td>FOR TIME LIMIT ENFORCEMENT 00902220</td>
</tr>
<tr>
<td>TEST G</td>
<td>PB10.1,DCAR5</td>
<td>BR TO ENTER CURB SLOT 00903000</td>
</tr>
<tr>
<td>TEST G</td>
<td>PB10.2,DCAR4</td>
<td>BR TO ENTER DP SLOT 00904000</td>
</tr>
<tr>
<td>TEST G</td>
<td>PB10.3,++3</td>
<td>BR TO ENTER QUEUE SLOT (++) 00905000</td>
</tr>
<tr>
<td>ASSIGN 4</td>
<td>CIRCO,PH</td>
<td>SEND TO RECIRCULATION ROAD 00906000</td>
</tr>
<tr>
<td>ENTER CTRL1</td>
<td>PH5</td>
<td>TO ASSURE PRIORITY FOR DP SLOT 00907000</td>
</tr>
<tr>
<td>PRIORITY 12</td>
<td>PH5,PH</td>
<td>LINK ON QUEUE IN RANDOM ORDER 00908000</td>
</tr>
<tr>
<td>ASSIGN 5</td>
<td>RN2,PH</td>
<td>00909000</td>
</tr>
<tr>
<td>DCAR2 LEAVE</td>
<td>PH0</td>
<td>LEAVE DP STORAGE 00910000</td>
</tr>
<tr>
<td>UNLINK VSOPQY,DCAR3.1</td>
<td></td>
<td>UNLINK 1 FROM CORR. QUEUE CHAIN 00911000</td>
</tr>
<tr>
<td>ASSIGN 6</td>
<td>VSOPDCV,PH</td>
<td>CHANGE PH6 TO CURB SLOT # 00912000</td>
</tr>
<tr>
<td>ASSIGN 10.1,PH</td>
<td></td>
<td>NOW AT CURB 00913000</td>
</tr>
<tr>
<td>PRIORITY 10</td>
<td>PH6</td>
<td>DROP TO NORMAL PRIORITY 00914000</td>
</tr>
<tr>
<td>ENTER PH5</td>
<td></td>
<td>ENTER CURB STORE 00915000</td>
</tr>
<tr>
<td>ASSIGN 5</td>
<td>RN2,PH</td>
<td>RELINK IN RANDOM ORDER 00916000</td>
</tr>
<tr>
<td>LINK DPLIC,PH</td>
<td></td>
<td>00917000</td>
</tr>
<tr>
<td>* DCAR3 TEST E</td>
<td>PB7.0,ENPCB</td>
<td>BR IF VEH UNLINKED IS ENPLANING 00918000</td>
</tr>
<tr>
<td>LEAVE PH5</td>
<td></td>
<td>LEAVE QUEUE STORAGE 00919000</td>
</tr>
<tr>
<td>ASSIGN 6</td>
<td>VSOPDCV,PH</td>
<td>CHANGE PH6 TO DP STORE # 00920000</td>
</tr>
<tr>
<td>ASSIGN 10.2,PH</td>
<td></td>
<td>NOW USING DP 00921000</td>
</tr>
<tr>
<td>DCAR4 PRIORITY 11</td>
<td>PH6</td>
<td>SET DP PRIORITY 00922000</td>
</tr>
<tr>
<td>DCAR5 ENTER PH5</td>
<td></td>
<td>CURB OR DP STORE 00923000</td>
</tr>
<tr>
<td>UNLINK E DPLIC,CTRL1,1,10PH,PH10,++2</td>
<td></td>
<td>MATCH CAR WITH PAX (++) 00924000</td>
</tr>
<tr>
<td>TRANSFER DPLIC9</td>
<td></td>
<td>00925000</td>
</tr>
<tr>
<td>ASSIGN 5</td>
<td>RN2,PH</td>
<td>LINK IN RANDOM ORDER 00926000</td>
</tr>
<tr>
<td>* ...TIME LIMIT ENFORCEMENT</td>
<td></td>
<td>00927000</td>
</tr>
<tr>
<td>GENERATE</td>
<td>...1,9,2PH</td>
<td>&quot;POLICE OFFICER&quot; 00928000</td>
</tr>
<tr>
<td>DCAR6 ADVANCE VS1ENFY</td>
<td></td>
<td>FREQUENCY OF ENFORCEMENT 00929000</td>
</tr>
<tr>
<td>UNLINK DPLIC,DCAR8,ALL,BVSOLIMB</td>
<td></td>
<td>UNLINK ANY VEHICLES WHICH HAVE BEEN DOUBLE-PARKED LONGER THAN LIMIT 00930000</td>
</tr>
<tr>
<td>ASSIGN 1</td>
<td>MHB(12.1),PH</td>
<td># OF DEPLANING CURBS 00931000</td>
</tr>
<tr>
<td>ASSIGN 1+</td>
<td>MHB(8.1),PH</td>
<td>ADD # OF ENPLANING CURBS 00932000</td>
</tr>
<tr>
<td>ASSIGN 2</td>
<td>DPOCS</td>
<td>FIRST QUEUE CHAIN 00933000</td>
</tr>
<tr>
<td>DCAR7 UNLINK PH2,DCAR9,ALL,BVSOLIMB</td>
<td></td>
<td>UNLINK ANY VEHICLES IN QUEUE LONGER THAN LIMIT 00934000</td>
</tr>
<tr>
<td>ASSIGN 2+</td>
<td>1PH,DCAR7</td>
<td>00935000</td>
</tr>
<tr>
<td>LOOP PH5,DCAR7</td>
<td></td>
<td>00936000</td>
</tr>
<tr>
<td>TRANSFER DCAR6</td>
<td></td>
<td>00937000</td>
</tr>
<tr>
<td>DCAR8 UNLINK VSOPQY,DCAR3.1</td>
<td></td>
<td>UNLINK 1 FROM CORR. QUEUE CHAIN 00938000</td>
</tr>
<tr>
<td>DCAR9 LEAVE PH6</td>
<td></td>
<td>DP OR QUEUE STORAGE 00939000</td>
</tr>
<tr>
<td>PRIORITY 10</td>
<td></td>
<td>DROP TO NORMAL PRIORITY 00940000</td>
</tr>
<tr>
<td>TEST E PB7.1,++2</td>
<td></td>
<td>IF ENPLANING, FALL THROUGH 00941000</td>
</tr>
<tr>
<td>ASSIGN 11+</td>
<td>M1,PH</td>
<td>ADD WAITING TIME TO PH11 00942000</td>
</tr>
<tr>
<td>TEST E PB7.0,CIRCO</td>
<td></td>
<td>BR IF UNLINKED VEH IS ENPLANING 00943000</td>
</tr>
<tr>
<td>TRANSFER CIRCO</td>
<td></td>
<td>RECIRCULATE 00944000</td>
</tr>
<tr>
<td>* DPLIC HELP</td>
<td>FORTM,0,PH2,PH11,PH7.PH1 LOC,FAC1,PH9ROW,PH1ROW</td>
<td>00945000</td>
</tr>
<tr>
<td>* ADVANCE XHSTVXH</td>
<td></td>
<td>00946000</td>
</tr>
<tr>
<td>* SAVEVALUE PH2+,PB5.XH</td>
<td></td>
<td>CONGESTION 00947000</td>
</tr>
<tr>
<td>TRANSFER PHN,DPLIC</td>
<td></td>
<td>BRANCH BY TRANSP MODE 00948000</td>
</tr>
</tbody>
</table>

A-3-25
ADVANCE XH$TRVXH, XH CONGESTION 01020000
SAVEVALUE PH2+,PB5,XH BR IF SUCCESSFUL 01022000
UNLINK E DPL2C,DPCG1,1,10PH,PH10,++2 MATCH PAX AT CURB 01023000
TRANSFER ,DPCG2 OTHERWISE LINK TO SPECIAL CHAIN 01024000
LINK DPL3C,FIFO 01025000

DPCG1 GATE LS PASSL WAIT FOR INFO FROM GREETER 01026000
ASSIGN 5+XBSPASS2,PB ADD GREETERS TO PARTY 01027000
ASSIGN 14.XBSPASS3,PB PICK UP PARKING LOT NUMBER 01028000
LOGIC R PASSL PERMIT NEXT USE OF PASS S'VALUES 01029000
ASSIGN 6.2,PB CHANGE MODE TO PARKING 01030000
ASSIGN 2.1,PB NEW PROC. FN. 01031000
TRANSFER ,CTRL0 01032000

DPCG2 GATE LR PASSL ONLY ONE AT A TIME 01033000
SAVEVALUE PASS2.PBSXB NUMBER OF GREETERS 01034000
SAVEVALUE PAS53.PB14 PARKING LOT NUMBER 01035000
LOGIC S PASSL LET PAX PICK UP INFO 01036000
TERMINATE 01037000

...ENPL ANG CURB 01038000

ENPC0 SAVEVALUE ARDXH+,1,XH COUNT OF VEH ON ARRIVING ROAD 01040000
ENPCR SAVEVALUE ENPXH+,1,XH HELP PL FORM,11,MH1(PH1,3).PB AIRLINE.MODE 01041000
ASSIGN 8.VESEPCK,PH 1 LESS THAN 1ST ENPL CURB STO NO 01042000
ASSIGN 4.PHB,PB SAVE CURB STO NO MINUS 1 01043000
ASSIGN 5.VESEPCC,PH 1 LESS THAN 1ST DBL PKING STO NO FOR ENPL 01044000
TEST NE PB10,4,ENPCA IF CIR VEH, TRANSFER TO ENPCA 01045000
ASSIGN 8.VESEPCL,PH 1 LESS THAN 1ST ENPL STO NO 01046000
TRANSFER NE PB10,4,++2 IF CIR VEH SKIP NEXT TEST 01047000
TEST NE PH6.PH8.ENPC1 IS ENPL STO NO SAME AS DESTINATION STO NO 01048000
ASSIGN 11+.M2,PH ADD WAITING TIME TO PH1 01049000
TRANSFER NE PB10,4,++2 IF CIR VEH SKIP NEXT TEST 01046000
TEST NE PH6.PH8.ENPC1 IS ENPL STO NO SAME AS DESTINATION STO NO 01047000
ASSIGN 11+.M1,PH ADD WAITING TIME TO PH1 01048000
ASSIGN 5+PH4,PB ENTER QUEUE FOR BLOCKED LANE 01049000
MARK PAX FOR WAITING TIME 01050000
TEST E BVS$INFUL,0 STOP TRAFFIC IF DBL PKING & NO OF PASSING 01051000
ASSIGN 11+.M1,PH ADD WAITING TIME TO PH1 01047000
ASSIGN 5+PH4,PB ENTER QUEUE FOR BLOCKED LANE 01048000
TEST E S+PH5,0,ENPCB IF CURRENT CONTENTS OF DBL PKING STO 01049000
MARK PAX FOR WAITING TIME 01050000
ASSIGN 5+PH4,PB ENTER QUEUE FOR BLOCKED LANE 01048000
TEST E S+PH5,0,ENPCB IF CURRENT CONTENTS OF DBL PKING STO 01049000
SAVEVALUE NOLAN,VSNLNMX,XH SET NO OF PASSING LANES TO MAX AVAIL 01047000
SAVEVALUE TOL4+,1,XH SAVE COUNT 01048000
TRANSFER ++3 01049000
ENPCB SAVEVALUE NOLAN,VSNLNM1,XH SET NO OF PASSING LANES TO 1 LESS 01047000
SAVEVALUE TOL5+,1,XH SAVE COUNT 01048000
TRANSFER ++3 01049000
ENPCB SAVEVALUE NOLAN,VSNLNM1,XH SET NO OF PASSING LANES TO 1 LESS 01047000
SAVEVALUE TOL5+,1,XH SAVE COUNT 01048000
MARK 01049000
ASSIGN 11+.M1,PH ADD WAITING TIME TO PH1 01050000
TEST E BVS$INFUL,0,ENPC1 IF CIR VEH IS AT LAST ENPL CUR STO 01051000
TRANSFER TO ENPC1 01052000
TRANSFER ,ENPC1 MOVE TO NEXT ENPL STO NO 01047000
ENPC1 SAVEVALUE AENPC+,1,XH COUNT OF VEH ON ENPL CURB 01048000
ASSIGN 8.VSTMODV,PH  FOR TIME LIMIT ENFORCEMENT 01049000
TEST G PB10,7,ENPC5  BR TO ENTER CURB SLOT 01050000
TEST G PB10,2,ENPC4  BR TO ENTER DP SLOT 01051000
ASSIGN 4.CIRC1,PH  BR TO ENTER QUEUE SLOT 01052000
TRANSFER ,CTRL1  SEND TO RECIRCULATION ROAD 01053000
ENTER PH6  TO ASSURE PRIORITY FOR DP SLOT 01054000
PRIORITY 12  01055000
ASSIGN 5.RN2,PH  MARK PAX FOR WAITING TIME 01056000
TRANSFER ,CTRL1  MARK MARK PAX FOR WAITING TIME 01057000
ENTER PH6  LINK ON QUEUE IN RANDOM ORDER 01058000
ASSIGN 6.VSEPQCVPH  01059000
CHANGE PF16 TO CURB STORE 01060000
TRANSFER ,CTRL1  01061000
ASSIGN 6,VSEPQCVPH  LEAVE PH6  01062000
CHANGE PH6 TO CURB STORE 01063000
ASSIGN 10,PB  LEAVE QUEUE STORAGE 01064000
ASSIGN 2.PB  USING CURB  01065000
ASSIGN 11+,M1,PH  USING CURB SIDE CHECK IN 01066000
TRANSFER ,CTRL1  ADVANCE VSENPIF CURB SIDE CHECK IN TIME 01067000
TEST E PB9,0,3  TERMINATE 01068000
IF PRETICKETED  01069000
TRANSFER ,CTRL1  ADVANCE VSENPIF EMPTY CAR PARKING TIME 01070000
TEST E PB9,0,3  TERMINATE 01071000
IF PRETICKETED  01072000
TRANSFER ,CTRL1  ADVANCE VSENPIF CAR PARKING TIME 01073000
TEST E PB9,0,3  TERMINATE 01074000
IF PRETICKETED  01075000
TRANSFER ,CTRL1  ADVANCE VSENPIF VEH COUNT IN PARKING LOT 01076000
TERMINATE 01077000
TERMINATE 01078000
TERMINATE 01079000
TERMINATE 01080000
TERMINATE 01081000
TERMINATE 01082000
TERMINATE 01083000
TERMINATE 01084000
TERMINATE 01085000
TERMINATE 01086000
TERMINATE 01087000
TERMINATE 01088000
TERMINATE 01089000
TERMINATE 01090000
TERMINATE 01091000
TERMINATE 01092000
TERMINATE 01093000
TERMINATE 01094000
TERMINATE 01095000
TERMINATE 01096000
TERMINATE 01097000
TERMINATE 01098000
TERMINATE 01099000
TERMINATE 0110000
TERMINATE 01101000
SPLIT 1.++3
ASSIGN 5,P813,PB
TRANSFER .GATE3
ASSIGN 5,-,P813,PB
ASSIGN 13,0,PB
ASSIGN 1,WQ01,PB
ASSIGN 2,1,PB
TRANSFER ,CTRL0
GATE3 SAVEVALUE PH2+,PB5,XH
GATE1 QUEUE PH5,PB5
SAVEVALUE GTXH+,P813,XH
MARK ENTER PH5
DEPART PH5,PB5
ASSIGN 11+,M1,PH
ADVANCE VSGAT3V
GATE6 LEAVE PH5
GATE LS PH5,GATE7
SAVEVALUE PH2-,PB5,XH
TRANSFER ,CTRL0
GATE2 5 SAVEVALUE 1+,PH1,12,PB5,MM
TRANSFER ,CTRL0
* STARTBOARDING OPERATIONS
* GATE9 ADVANCE VSGATIV
ASSIGN 5,VSGAT2V,PH
LOGIC S PH5
ADVANCE XM80TXH
LOGIC R PH5
ASSIGN 5,M1H(PH1,9),PH
ASSIGN 5,M1H(PH5,3),PH
SAVEVALUE PH2-,M1H(PH1,12),XH
SAVEVALUE 1,PH1,12,0,MM
TERMINATE
*...GROUND TRANSPORTATION (MISC.)
* GRT00 - DEPLANING PAX - SELF
* GRT01 - ENPLANING PAX - SELF
* DEPLANING PAX - SELF
* GRT00 TEST E M1H(PH1,3),10,++2 TEST FOR SHUTTLE PAX (++2)
 ASSIGN 14,3,PH HELP D ORM16G,PH2,PB8,0
 ADVANCE XM80TXH TRANSFER .CTRL3
* ENPLANING PAX - SELF
* GRT01 TEST E M1H(PH1,3),10,++2 TEST FOR SHUTTLE PAX (++2)
 ASSIGN 14,3,PH HELP D ORM16G,PH2,PB8,1
 SAVEVALUE ARDXH+,1,XH COUNT OF VEH ON ARRIVING ROAD
 SAVEVALUE PXLXH+,1,XH ENTER CAR IN PARKING LOT
 SAVEVALUE PLXH+,1,XH VEH. COUNT IN PAX. LOT
 TRANSFER .CTRL3
* GREETERS
**GRT02 TEST E**

MH1(PH1,3),10,++2

TEST FOR SHUTTLE PAX (++2)

ASSIGN

14,3,PH

ASSIGN TO PARK FACIL 3

HELPA

FORTM,16,0,PB8,0

LOC,MODE,DEPL

SAVEVALUE

AROXH+,1,XH

COUNT OF VEH ON ARRIVING ROAD

SAVEVALUE

PKLXH+,1,XH

ENTER CAR IN PARKING LOT

SAVEVALUE

PLIXH+,1,XH

VEH COUNT IN PARKING LOT

TRANSFER ,CTRL0

**GREETERS TAKING CAR FROM PARKING TO CURB**

**GRT03 HELP A**

FORTM,16,PH2,PB8,0

LOC,MODE,DEPL

ADVANCE

XHSTRVXH

QUEUE

PH5

ENTER

PH5

DEPART

PH5

ADVANCE

VSPAR1V

LEAVE

PH5

SAVEVALUE

PKLXH-,1,XH

EXIT CAR FROM PARKING LOT

SAVEVALUE

RERXH+,1,XH

RECIRC. ROADWAY COUNT

SAVEVALUE

PKCMH+,1,XH

PARKING TO CURB RECIRCULATION

ASSIGN

4,OCAR1,PH

ROUTE TO DEPLANING CURB

TRANSFER ,CTRL1

**...PARKING (DEPL PAX - CARS)**

PARK9

QUEUE

PH5

MARK

SERVICE

MARK PAX FOR WAITING TIME

ENTER

PH5

DEPART

PH5

ASSIGN

1+,MI,PH

ADD WAITING TIME TO PH11

ADVANCE

VSPAR1V

PARKING EXIT SERVICE TIME

LEAVE

PH5

SAVEVALUE

PKLXH-,1,XH

EXIT CAR FROM PARKING LOT

SAVEVALUE

DROXH-,1,XH

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL0

**...RENT A CAR**

**FOLLOWING FOR DEPL PAX RENTING A CAR**

RCAR0 TEST E

PB8,3,CTRL0

RETURN TO CONTROL IF NOT RENTACAR

ASSIGN

10,PHSRCA2F,PH

CAR RENTAL AGENCY SELECT

HELP A

FORTM,6,PH2,PB10

LOC.AGENCY

ADVANCE

XHSTRVXH

TRAVEL TIME TO CAR RENTAL AREA

SAVEVALUE

PH2+,PB5,XH

CONGESTION

RCAR1 QUEUE

PH5,PB5

WAIT FOR FREE AGENT

MARK

PH5

SERVICE

ENTER

PH5

DEPART

PH5,PB5

ASSIGN

11+,MI,PH

ADD WAITING TIME TO PH11

RCAR2 ADVANCE

VSRCA1V

CAR RENTAL PROCESSING TIME

LEAVE

PH5

SERVICE

SAVEVALUE

PH2+,PB5,XH

CONGESTION

TRANSFER ,CTRL0

**DEPL PAX IN GROUND TRANSPORT WHO HAVE ALREADY RENTED CAR.**

**NOTE: CURRENT LOGIC ASSUMES PAX PICKS UP CAR AT AGENCY PARKING LOT.**

01208000

01209000

01209100

01210000

01211000

01212000

01213000

01214000

01215000

01216000

01220000

01221000

01222000

01223000

01224000

01224100

01225000

01226000

01227000

01228000

01229000

01230000

01231000

01232000

01233000

01234000

01235000

01236000

01237000

01238000

01239000

01240000

01241000

01242000

01243000

01244000

01245000

01246000

01247000

01248000

01249000

01250000

01251000

01252000

01253000

01254000

01255000

01256000

01257000

01258000

01259000

01260000

01261000

01262000

01264000
ADVERTISE

PA X - RENTACAR

NOTE: CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,

* IF ANY, IS DONE IN THE TERMINAL.

** MODE: DEP/AGENCY

COUNT OF VEH ON DEPARTING ROAD

TRANSFER ,CTRL

*NOTE:

CURRENT LOGIC ASSUMES RENTAL CAR RETURNED TO

**PARKING LOI (GENERAL OR AGENCY LOT). PROCESSING,
ASSIGN 1.7DP1F, PB
ASSIGN 2.2.PB  SKIP CONCOURSE FOR TDP1F FUNCTION
ASSIGN 13.1.PB  PAX IN PARTY
ASSIGN 5.1.PB  TOTAL IN PARTY
HELPA FORTM.18,0,0
TEST L MHS(1,1),0, xFLT3 IF IF AT END OF FLT TABLE
TEST E MHS(1,1),0, xFLT3 FT IF NO XFER FLTS
TERMINATE
SPLIT 1, XFLT3

ADD FLT TO XFR FLT TABLE

TEST G MHS(1,1),0, xFLT2 BR IF NO TPAX FOR NEXT FLT
XFLT1 ADVANCE VSXFL1Y
HELPA FORTM.18, PH1, 2
XFLT2 ASSIGN 1, PH1, 2
HELPA FORTM.18, PH1, 3
SPLIT MHS(1,1), XFLT8
HELPA FORTM.18, PH1, 1
TERMINATE
XFLT8 TEST E BSXFL1B,1, XFLT2 BR IF ARRIV FLT OR NO TPAX
TRANSFER xFLT1

DELETE FLT FROM XFR FLT TABLE

XFLT3 TEST G MHS(1,1),0, XFLT9 BR IF XFER FLT TABLE EMPTY
ASSIGN 1, MHS(1,1)
XFLT7 ADVANCE VSXFL1Y
HELPA FORTM.18, PH1, 9
ASSIGN 5, MHS(1,1), PH1, 9
HELPA FORTM.18, PH1, 3
SPLIT MHS(1,1), XFLT8
HELPA FORTM.18, PH1, 1
TERMINATE
XFLT4 ASSIGN 1, PH1, 1
XFLT9 TEST L MHS(1,1),0, XFLT6 BR TEST FOR END OF TABLE
XFLT6 TEST E BSXFL1B,1, XFLT4 BR IF ARRIV FLT OR NO TPAX
TRANSFER xFLT7
XFLT8 ADVANCE VSXFL1Y
TRANSFER xFLT3

ERROR CHAIN

COLLECTS XACS CAUSING SELECTED ERRORS. PH4 WILL
NORMALY GIVE CLUE TO NATURE OF PROBLEM.
ERROR LINK ERRCH.FIFO

WHEN REMOVING ANY MODULE, UNCOMMENT ENTRY POINT CARD & OTHERS NOTED: 01378000
NOTE: TO REMOVE "CHECKIN", UNCOMMENT CHEKO, CHEK1, AND CHEK2.
TO REMOVE "DEPLCURN", UNCOMMENT DEPCO AND OPL9.
TO REMOVE "GATE (EMPLANING PAX)" UNCOMMENT GATE AND GATE9.
TO REMOVE "RENTACAR", UNCOMMENT RCARO, RCARS AND RCAR9.
CHEKO TRANSFER Ctrl0
CHEK1 TRANSFER Ctrl0
CHEK2 TRANSFER Ctrl0
CUSTO TRANSFER Ctrl0
OCARO TRANSFER Ctrl0
OPLC0 TRANSFER Ctrl0

A-3-33
*TRANSFER CTRLO 01388C00
*TRANSFER CTRLO 01389000
*TRANSFER CTRLO 01390000
*TRANSFER CTRLO 01391000
*TRANSFER CTRLO 01392000
*TRANSFER CTRLO 01393000
*TRANSFER CTRLO 01394000
*TRANSFER CTRLO 01395000
*TRANSFER CTRLO 01396000
*TRANSFER CTRLO 01397000
*TRANSFER CTRLO 01398000
*TRANSFER CTRLO 01399000

**CHANGE CARD READER**

**GENERATE ...1,120,2PH**

**PERFORM PREVIOUS AND READ NEXT**

**CHANGE CARD**

**IF NO STORAGE CHANGES, (**) 01407000**

**DON'T SPLIT**

**SPLIT 1 XAC TO CHANGE EACH**

**STORAGE, SEQUENCED IN PH1**

**WAIT TILL NEXT CHANGE**

**1412C00**

**IF WAIT WAS 0, WAIT TILL MH7 USED 14130000**

**14140000**

**HANG**

**CHANGE UNDER 01402000**

**ASSIGN XHSSAVXH,10.XH**

**SAVEVALUE**

**SAXH,PH1,XH**

**ASSIGN 2+30.PH**

**ASSIGN 1,MH7(PH1,1).PH**

**ASSIGN 2,MH7(PH2,1).PH**

**SAVEVALUE**

**7.XHSSAVXH,10.MH**

**SAVEVALUE**

**SAXH+,30.XH**

**COUNT DOWN OF XACS USING MH7**

**14350000**

**14360000**

**14370000**

**14380000**

**14390000**

**14400000**

**14410000**

**14420000**

**14430000**

**14440000**

**14450000**

**14460000**

**14470000**

**GENERATE ...1,127,12PH**

**ADVANCE XHSSAVXH**

**NEEDED TO DEFINE "CLKXH* AS XH**

**14480000**

**14480000**

**A-3-34**
SAVEVALUE INCHX,60,XH
HELP CCLINK,1
HELP FORTM,1,1
TEST G PH1.0,++3
(STOPO SPLIT 1,STOPO
TERMINATE PH1
SPLIT 1,++4
GATE LS JOBLS
(STOP1 SPLIT 1,STOPO
TERMINATE 100
SPLIT 1,++5
ADVANCE VSINCX1
HELPA FORTM,21,XHSINCXH
ADVANCE XHSINCXH
TRANSFER ,+2
ASSIGN 1,127,PH
MSAVEVALUE 1.1,PH1,FNSRANDF,ML
LOOP 1PH,=1
PRIORITY 126
ADVANCE XFSENOXF
(ADVANCE TO END OF RUN)
HELPA FORTM,20,C1
* PRINT ,.C,C
* PRINT ,.W,C
* PRINT ,.S,C
* PRINT ,.Q,C
* PRINT ,.U,C
* PRINT ,.T,C
* PRINT ,.X,C
* PRINT ,.XH,C
* PRINT ,.XB,C
* PRINT ,.XL,C
* PRINT ,.LG,C
TERMINATE 1
* GENERATE 11700,,,1
HELPA FORTM,22
ADVANCE 300
TRANSFER ,+2
* GENERATE 12300,,,1
TERMINATE 1
* TIMER FOR CHANGING HOURLY TABLE FOR PAX WAITING TIME
GENERATE 11,0
SAVEVALUE PXTBN+,1,XB
ADVANCE 3600
TRANSFER ,+2
* FUNCTION PH1,LB7
Mnemonic Link Function—See FORTRAN Call
.CM01/.CM02/.CM03/.CM04/.CM05/.CM06/.CM07/.CM08/.CM09
CM102
.CARC0/.BACCO/.DPLC0/.CHEK2/.CHEK3/.CTGRO/.ERROR/.SEC0/.CTRL0/.CTRL1
.EPCBS/.GRTG0/.GRTXH/.CPXH/.CRBH/.CTXH/.PCBXL
.JOBLS
* SIMULATE 9
* START 1,NP

A-3-35
* RESET 1. NP
* NOXREF
* REPORT
* HMS TITLE 1. FLIGHT SCHEDULE
* HMS TITLE 2. AIRLINE INFORMATION TABLE
* HMS TITLE 3. TABLE OF POINTS
* HMS TITLE 4. ENPLANING PAX TICKETED
* HMS TITLE 5. TRANSFER FLIGHT TABLE
* HMS TITLE 6. WALKING TIME BETWEEN POINTS
* HMS TITLE 7. RANDOM NUMBERS FOR BAGS
* HMS TITLE 8. USED IN MATRIX 9
* HMS TITLE 9. FACILITY TABLE
* HMS TITLE 11. COUNTER FOR PAX LEAVING CONCOURSE
* HMS TITLE 12. COUNTER FOR PAX LEAVING SECURITY
* HMS TITLE 13. COUNTER FOR PAX LEAVING AIRLINES
* LMS TITLE 1. RANDOM NUMBER TABLE
* LMS TITLE 2. GROUND TRANSACTION MODAL CHOICE
* OUTPUT REMOVE * TO GET ALL STATICS
  
  END