CORSIM Data Dictionary

Version 6.0

Prepared by:

ITT Industries, Inc., Systems Division
ATMS R&D and Systems Engineering Program Team
P O Box 15012
Colorado Springs, CO 80935-5012

Prepared for:

FHWA Office of Operations Research, Development and Technology
Federal Highway Administration
Turner-Fairbank Highway Research Center
6300 Georgetown Pike
McLean, Virginia 22101-2296

December 2006

NOTICE

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.

This document does not constitute a standard, specification, or regulation.

The Federal Government and McTrans are not responsible for implementation decisions (e.g., construction designs, traffic signal timings) made based on the results of analyses performed using the computer programs described herein.

Foreword

This Data Dictionary describes the data used by the CORSIM simulation within the **T**raffic **S**oftware **I**ntegrated **S**ystem (TSIS). The McTrans Center is maintaining TSIS.

For detailed information on how to use the individual components of TSIS, please refer to the User's Guides for those individual components.

Abstract

This document defines all CORSIM data (arrays and scalars) that are stored either in COMMON or in FORTRAN 90 data modules. Many of these data items are exported for access by processes outside the main CORSIM library.

With the release of TSIS 6.0, CORSIM dynamically allocates some of its arrays for efficiency and flexibility. Because COMMON blocks do not support dynamic array allocation, CORSIM uses the FORTRAN 90 data module construct, which does allow dynamic allocation. Each data grouping, described herein, indicates if the data in the group are defined in a module, and how to reference each item in the group when importing the item for external access.

Note that many data items are defined as integers. This design concept permits multiple data items to be packed within one data word. That is, specified bits are allocated to each data item within each data word.

The first three sections are devoted to the major data categories in CORSIM: global data, FRESIM data, and NETSIM data. These sections are further subdivided into groups according to function, e.g. link-specific, vehicle specific, etc. Within each such grouping, the array and scalar names are arranged alphabetically for ease of reference. The descriptions identify arrays as subscripted variables.

Each variable is described in detail. If more than one data item is packed within a computer word, each such item is described and the bit-positions used for that item are defined. If only one data item occupies a computer word, the bit positions are not indicated and it is understood that the data item occupies all 32 bits of the word.

The final section describes the data modules used for the CORSIM trip generator. These modules are used by the trip generator and for processing time-varying entry volumes and turn percentages. They are also used by the CORSIM DTA enhancements. As previously stated, FORTRAN 90 allows the use of data pointers, enabling CORSIM to use linked-list queues and to dynamically allocate arrays, both of which were impossible to do with earlier versions of FORTRAN.

Contents

1 GLOBAL Database

These variables belong to the group of COMMON blocks in "GLOBAL.INC." This group is accessed by all submodels. Variables in this section are organized as follows:

- System Data
- Bus Route Data
- Bus Station Data
- Calibration Data
- Traffic Assignment Data
- Real-Time Traffic Control Data
- PRG Commons Data

1.1 GLOBAL System Database

This section defines all GLOBAL scalars and arrays blocks that do not belong to any of the other functional categories presented in sections 1.2 through 1.6. These arrays are exported.

Name	Common		Description
ANODE	GLR045		Highest internal user node number (last entry of GLOBND that is actually used).
AROUTE	GLR046		Highest bus route number.
BEGTME	GLR048		Clock time in hours and minutes at start of simulation. For example, if simulation started at 4:30 p.m., the value here should be 1630.
BUSLNK(B)	GLR250		Bus path array - contains subnetwork type, link number, and bus route number, B.
		1-14	Link ID
		15-21	Route Number
		22-32	Subnetwork ID
CASINP	GLR156		Character string containing case name. 8 characters.
CNTINT	GLR051		Number of time intervals remaining until next intermediate output from Traflo models.
DMPBEG	GLR802		Time that vehicle data dump is to begin.
DMPFRQ	GLR804		Frequency at which vehicle data is to be dumped.
DMPINT	GLR805		Time in initialization that vehicle data dump stops.

DMPSPN	GLR803		Span of time over which vehicle data is to be dumped.
DPPINT	GDET01		Detector point processing interval. Used for on-line point processing.
ELAPST(I)	GLR095		Elapsed time in seconds after commencement of simulation to (I) start of section I of microscopic intermediate output. Up to 3 sections of intermediate output are allowed (I=3).
ERMSG	GLR700		Array containing flags for error/warning messages generated.
ERRCT	GLR055		Counter of errors found in input stream.
FILDUR	GLR148		Actual duration of fill time (initialization).
FROFFL	GLR551		In FRESIM, code (0,1) if point processing, MOE estimation and/or off-line incident detection (is, is not) desired.
FUEL	GLR103		NETSIM Environmental Code of form XY where,
			X = (0,1) if fuel emission rates (are not, are) to be printed

Y	Simulation	Environmental Measures	Rate Tables Used	Trajectory File Processing
0	Performed	Calculated	Embedded	Not Written
1	Performed	Calculated	Embedded	Written
2	Performed	Calculated	Modified	Not Written
3	Performed	Calculated	Modified	Written
4	Not performed	Calculated	Embedded	Not Written
5	Not performed	Calculated	Modified	Not Written
6	Performed	Not Calculated	Embedded	Not Written
7	Performed	Not Calculated	Embedded	Written

When Y = 4 or 5 simulation will not be performed and trajectories will be read from a file created during a previous simulation.

FUELCD	GLR057		CORFLO Environmental Code (0,1) if calculation of fuel consumption and vehicle emissions (is not, is) suppressed.
FUELFR	GLR550		Fuel option code for FRESIM, 0-7 (same discussion as FUEL for NETSIM).
GCLK	GLR058		Case clock. Contains the time since beginning of case, sec. (Not incremented during fill time).
GLOBID1	GLOBIDS		Global vehicle ID of the first vehicle generated in the previous minute of simulation.
GLOBID2	GLOBIDS		Global vehicle ID of the last vehicle generated in the previous minute of simulation.
GLOBND(I)	GLR012		Maps the global node number, I, with its associated subnetwork-specific node number:
		1-4	Identification code of the subnetwork in which the global node, I, is located:

Code	Subnetwork
3	NETSIM
8	FRESIM

		5-15	The subnetwork-specific node number associated with the global node, I.
			The dimension of GLOBND is NMAX.

GSCAN	GLR603	Highest global vehicle id number currently used.
HDIST	GLR183	Headway distribution flag. 0 = Normal, 1 = Shifted Exponential, 2 = Erlang
INTVEH(I)	GLR260	For CORFLO, maximum allowable number of vehicles entering an interface node per second. The dimension of I is MAXINT. (3 bits per second per node).
LENINT	GLR060	Time interval duration, seconds.
LENPRD	GLR061	Length of current Time Period, seconds.
LNKID	GLR306	Internal link number to be traced
LNTMPR	GLR014	Contains length of each Time Period, for up to 19 Time Periods.
LU5	GLR105	Logical unit number 5, input file.
LU6	GLR106	Logical unit number 6, output file.
LU7	GLR107	Logical unit number 7, copy of input file
LU12	GLR112	Logical unit number 12, temporary file for NETSIM.
LU15	GLR115	Logical unit number 15, O-D matrix for TA.
LU16	GLR116	Logical unit number 16, mapping to path network for TA.
LU17	GLR117	Logical unit number 17, geometry link file for TA.
LU18	GLR118	Logical unit number 18, updated geometry link file for TA.
LU19	GLR119	Logical unit number 19, turn movement data for TA.
LU20	GLR120	Logical unit number 20, minimum path tree file for TA.
LU31	GLR131	Logical unit number 31, NETSIM environmental tables.
LU32	GLR132	Logical unit number 32, NETSIM vehicle trajectories.
LU33	GLR133	Logical unit number 33.
LU35	GLR135	Logical unit number 35, FRESIM detector actuation data.
LU36	GLR136	Logical unit number 36, FRESIM vehicle trajectories.
LU37	GLR137	Logical unit number 37, FRESIM point processing data.
LU38	GLR138	Logical unit number 38, FRESIM MOE estimation data.
LU39	GLR139	Logical unit number 39, FRESIM off-line incident detection results.
LU40	GLR240	Logical unit number 40, NETSIM graphics-geometry data.
LU41	GLR241	Logical unit number 41, NETSIM graphics-vehicle data.
LU42	GLR242	Logical unit number 42, NETSIM graphics-MOE data.
LU43	GLR243	Logical unit number 43, NETSIM graphics-bus MOE data.
LU44	GLR244	Logical unit number 44, NETSIM graphics-spillback data.
LU45	GLR245	Logical unit number 45, NETSIM actuated controller data.
LU48	GLR248	Logical unit number 48, NETSIM fixed time interval data.
LU49	GLR249	Logical unit number 49, NETSIM fixed time control codes.
LU52	GLR252	Logical unit number 52, TA output file.

LU55	GLR255	Logical unit number 55, L1 actuated controller and detector.
LU56	GLR256	Logical unit number 56.
LU61	GLR261	Logical unit number 61,L2 histogram data.
LU71	GLR271	Logical unit number 71, FREFLO MOE data file.
LU82	GLR282	Logical unit number 82, NETSIM O-D output data.
LU84	GLR984	Logical unit number 84, TRAF input errors.
LU90	GLR985	Logical unit number 90, FRESIM graphics-geometry file.
LU91	GLR986	Logical unit number 91, FRESIM graphics-vehicle data file.
MAXBRT	GLR001	Maximum allowable bus route number.
MAXGLK	GLR002	Maximum number of global links, i.e., the number of links that can exist in the entire "global" network.
MAXGVH	GLR602	Maximum number of global vehicles.
MAXMAN	GLR003	Maximum allowable number of bus maneuvers that can be performed on all routes. (Dimension of MANUVR array).
MAXINT	GLR262	Maximum allowable size of interface array, INTVEH
MAXSPT	GLR561	Maximum allowable number of interface nodes
MAXSTA	GLR005	Maximum allowable bus station number.
MAXSUB	GLR004	Maximum allowable number of subnetworks (dimension of YEQLB and YSUB arrays).
MAXTP	GLR006	Maximum allowable number of Time Periods (dimension of LNTMPR array).
MSGCT	GLR065	Input warning message counter.
MXBSLK	GLR251	Maximum number of buses loaded in BUSLNK array
NETCD	GLR062	Subnetwork type code: **Code** — **Subnetwork** 3 — NETSIM 8 — FRESIM
NEXTRN	GLR066	Next case code(0,1) if another case (doesn't,does) follow.
NMAX	GLR011	Maximum allowable internal node number (user node number).
NRUN	GLR067	Run identification number.
NSLICE	GLR068	Number of time slices per time interval.
OUTFRQ(I)	GLR096	Frequency of microscopic intermediate output, during section I in sec. There can be up to 3 sections of intermediate output (I=3). Output begins at time ELAPST(I).
OUTSPN(I)	GLR097	Span in seconds of time during which Section I of microscopic intermediate output will be produced, starting at time ELAPST(I). Up to 3 sections of intermediate output may be specified.
PUTINT	GLR070	Number of time intervals between successive intermediate output for macroscopic models.
PUTSTD	GLR071	Number of Time Intervals between successive standard output.

SIGTRN	GLR152	Code for NETSIM signal transition (0, 1, 2, 3) = (no, immediate, two-cycle, three-cycle) transition.
SLIDUR	GLR074	Duration of a time-slice (hundredths-of-a-sec).
SNPFRQ	GLR144	Frequency at which vehicle and link data is written to the link snapshot/animation data files (secs.).
TICNT	GLR076	Time Interval counter.
TMINIT	GLR077	Fill time in minutes. A negative entry indicates program will exit if equilibrium not attained. (This value is rounded to the nearest integer multiple of a Time Interval.)
TPCLK	GLR078	Time Period clock, sec. Clock is set to zero at beginning of each Time Period and is incremented by LENINT at the end of each Time Interval.
TPCNT	GLR079	Time Period counter: Counter will be zero during first pass over input stream. During Time Period processing, it will contain the Time Period number.
TRCSD1	GLR980	Tracing flag for seeds, type 1.
TRCSD2	GLR981	Tracing flag for seeds, type 2.
TRCSD3	GLR982	Tracing flag for seeds, type 3.
TRCDBG	GLR983	Debugging flag used all over TRAF.
TRNCD	GLR102	Code (0,1) if movement-specific output (is not, is) requested for NETSIM subnetwork.
TTLGVH	GLR601	Total # of vehicles on all subnetworks.
TTLSUB	GLR083	Total number of subnetworks being simulated.
TYPERN	GLR086	Code: 1 = if model is to perform simulation only. 2 = if model is to perform traffic assignment only. 3 = if model is to perform both simulation and traffic assignment. The above code is negative if the model is to perform diagnostics on the input data and not simulate or perform traffic assignment.
UNITIN	GLR099	Code (0,1) if data is input in (English, Metric) units.
UNITOT	GLR100	Code (0,1,2,3) if output is requested in (same units as input, English, Metric, Both units).
W23	INPFLAGS	Logical flag indicating that time-varying turn percentages are being used for the NETSIM network.
W26	INPFLAGS	Logical flag indicating that time-varying turn percentages are being used for the FRESIM network.
WODOUT	GLR702	Flag (T,F) if Origin/Destination output (is, is not) requested.
WTPSOD	GLR701	Flag (T,F) if time period specific O/D output (is, is not) requested.
XC7000(I)	GLR145	Node coordinates for all interface nodes numbered 7000-7999. Coordinate data for interface node IN is located in array elements I and I+1 where I=(IN-7000)*2 as follows: **Word I** X coordinate of interface node IN, feet.

		Word I+1 Y coordinate of interface node IN, feet.
XC8000(I)	GLR146	Node coordinate data for all entry/exit nodes numbered 8000-8999. Coordinate data for entry/exit node IN is located in array elements I and I+1 where, I=(IN-8000)*2 as follows: **Word I** X coordinate of entry/exit node IN, feet. **Word I+1** Y coordinate of entry/exit node IN, feet.
XGCOOR(I)	GLR147	Global node specific array containing node coordinates for internal nodes. Coordinate data for node IN is located in array elements I and I+1 where I=(IN-1)*2. **Word I** X coordinate of node IN, feet. **Word I+1** Y coordinate of node IN, feet.
XPHSCD(I)	GLR037	This array contains the divisor used to unpack the phase code when phase I is active.

I	XPHSCD(I)
1	1
2	32
3	1024
4	32768
5	1048576
6	33554432

XSEED	GLR087	Random number seed.
XSEED2	GLR150	Random number seed for generating traffic stream in NETSIM subnetwork.
XSEEDF	GLR712	FRESIM random number seed.
XSPINT(J)	GLR013	The percentage of a time interval during which an interface node experiences spillback. Contains the spillback percentage for interface node 7XXX, J = (7XXX - 6999). During the course of a Time Interval, the number of seconds of spillback may be temporarily stored in the array while processing Entry Interface nodes. At the end of the Time Interval the spillback duration in seconds is then converted to a percentage of the Time Interval.
XVETRP	GLR088	Cumulative number of trips completed on the network (incremented each time a vehicle is discharged through an Exit link or a sink node).
VEHID	GLR305	Internal vehicle number to be traced
WFPASS	GLR987	Logical flag indicating first pass through the input.
YACTND	GLR705	Flag (T) if actuated phase duration output should be printed.
YCASE	GLR089	Case termination flag. Is set to .TRUE. if an error occurs during simulation so no further simulation for this case will be attempted.

YEOI	GLR090	Flag (FALSE., .TRUE.) if end-of-input (has not, has) been reached during input processing.
YEQLB(I)	GLR016	Subnetwork-specific equilibrium flags. Flag is TRUE after equilibrium has been attained. YEQLB(1) = Flag for FRESIM YEQLB(3) = Flag for NETSIM
YFIRST	GLR301	Flag (T, F) if this (is, is not) the first error checking pass over input data.
YGRAPH	GLR142	Flag (.T., .F.) if user (does, doesn't) want graphics output.
YINIT	GLR091	Initialization flag. Set to TRUE during initialization and set to FALSE after initialization has been completed.
YLAHEY	GLR307	Protected-mode version flag.
YMSDOS	GLR300	MS-DOS version flag.
YPRINT	GLR092	Flag (.T., .F.) if input data (is, isn't) to be printed.
YRTALG	GLR093	Real time algorithm flag. Set to TRUE if real time algorithm is to be used.
YTRACE	GLR302	Trace flag for debugging information.
YSUB(I)	GLR015	Subnetwork existence flag. Flag is TRUE if subnetwork data has been read.
YVSVOT	GLR801	Flag (T, F) if vehicle type specific output (is, is not) to be printed.
ZAVGBO	GLR123	Average bus occupancy for all subnetworks for all time periods.
ZBOSUM(I)	GLR124	Sum of bus occupancies for each subnetwork, I.
ZCNVRT(6)	GLR101	Array of factors to convert English units to Metric and vice-versa as follows:

Element	Converts from	To	Value
1	feet	meters	0.3048
2	miles	kilometers	1.60935
3	gallons	liters	3.78541
4	meters	feet	3.28084
5	kilometers	miles	0.621369
6	liters	gallons	0.264172

ZMILE	GLR010	Network sum of vehicle-miles.
ZMOVE	GLR017	Network sum of free-flow vehicle-hours.
ZSSTEP	GLR552	Time step in FRESIM.
ZTIME(I)	GLR019	Network sum of vehicle-hours, for subnetwork I.

1.2 GLOBAL Bus Route Database

The arrays defined in this section contain data associated with each bus route, IBR, in the analysis network. These arrays are exported.

Name	Common	Description
ABUSTT(IBR)	GLR054	Average travel time per bus on route, IBR.

BINDEX(IBR)	GLR025		Index in the MANUVR array which locates the first maneuver of a bus on route IBR.
BOFSET(IBR)	GLR703		Offset (sec) for first bus emitted on route, IBR.
BSTRPS(IBR)	GLR020		Total number of trips completed on route, IBR.
EMHDWY(IBR)	GLR023		Emission headway of buses on route, IBR, sec. Set to zero if route IBR is not specified.
EMTIME(IBR)	GLR024		Time remaining until next bus emission on route, IBR, sec.
ENTLNK(IBR)	GLR022	1-4	Subnetwork number where route, IBR, begins.
		5-15	Subnetwork link number where route, IBR, begins.
MANUVR (J)	GLR026		Contains the sequence of maneuvers performed by a bus on a given route. The index, J, identifies the next maneuver to be performed. This index is set equal to BINDEX(IBR) when a bus on route, IBR, is emitted, and is incremented by one every time the bus enters a new link or after it leaves all bus stations except the last one on a link. MANUVR is packed as follows:
			This entry is set to negative only if the following entry, MANUVR(J+1), specifies that there is another bus station on this route, on the current link.
		1-12	Number of next bus station where bus must stop on this link. Set to zero if there are no bus stations on the current link. If the code contained in bits 13-15 is a 5 then bits 1-12 will contain the bus route number, IBR.
		13-15	Turning code at the downstream node of current link:

Code	Turn Movement
0	Left
1	Thru
2	Right
3	Left Diagonal
4	Right Diagonal
5	Exit or Interface Node

XBSTRV(IBR)	GLR021		Total travel time of buses on route, IBR, sec.
XBSEED(IBR)	GLR151		Random Number Seed.

The highest bus route number specified is AROUTE (i.e., IBR =1, 2, ... AROUTE)

1.3 GLOBAL Bus Station Database

The following array defines the characteristics and current status of all bus stations on the analysis network. These arrays are exported.

Name	Common		Description
STASHN(K)	GLR027		Contains data for all stations on the network. Data for each station, IBS, is stored in 10 sequential data elements,I. Element I for station, IBS, can be obtained from STASHN(K), where
			K=10 * (IBS-1) + I
1		1-3	Lane blocked when bus is at station (Zero if a protected station).

	4-15	Distance between downstream end (front) of station and upstream node of link, feet.
2	1-3	Station type used to obtain the desired statistical distribution of dwell time.
	4-6	Capacity of station, buses.
	7-15	Mean dwell time, sec.
3		Number, IV, of first bus vehicle in station, IBS. During input processing, this entry is packed in the form 10x + y, where x is the global link number and y is the subnetwork type.
4		Number, IV, of second bus vehicle in station, IBS. During input processing, this entry contains the user's upstream node number for the link where this station is located.
5		Number, IV, of third bus vehicle in station, IBS. During input processing, this entry contains the user's upstream node number for the link where this station is located.
6		Number, IV, of fourth bus vehicle in station, IBS. During input processing, this entry contains the distance between the front end of the station and the stopline (feet).
7		Number, IV, of fifth bus vehicle in station, IBS. During input processing, this entry contains the length of a protected station on the NETSIM subnetwork which serves as a right turn pocket.
8		Number, IV, of sixth bus vehicle in station, IBS.
9		Total dwell time of buses using station, IBS, in sec.
10		Percent of buses scheduled to service passengers at this station, which will stop at this station. (The other buses bypass the station due to lack of demand).

1.4 GLOBAL Network Calibration Database

This section presents those arrays and scalars which contain empirical data used to calibrate traffic operations. These arrays are exported.

Name	Common	Description
ACCGAP	GLR044	Acceptable gap for left-turning vehicles, tenths- of-a-second. This value is set to 45.
ARTESP	GLR047	Moving vehicles unimpeded by others must slow as they approach an intersection if they are to negotiate a turning maneuver. The turning speed for right turners, applied, deterministically is 13 feet/second. This value may be changed when the NETSIM model is executed by using card type 140. When card type 140 is used, the new value of ARTESP is applied in all relevant models.
BUSLDR	GLR049	To reflect the more sluggish operating characteristics of buses relative to autos, headways obtained from the HDWPCT array are multiplied by the percent stored in BUSLDR. Currently, this value is 160.

DISIRT	GLR053	Surcharge to discharge headway for through vehicles, to obtain the value of right-turn discharge headway, tenths-of-a-second. DISIRT is set to 4.
DWLPCT	GLR035	The dwell time assigned to a bus servicing passengers at a station is obtained by multiplying the mean dwell time specified for the station , by a factor. A random number, K, between 1 and 10, inclusive, is used to enter a distribution of dwell time factors (in percent) which is stored in the dwell array. There are 6 such distributions in the DWLPCT array, one for each of up to 6 possible bus station "types". The index J to the DWLPCT array is calculated as follows:

$$J = 10 (I-1) + K$$

where K is a random number between 1 and 10 and I is the station "type" specified on card type 185.

The following distributions are embedded in the model for the 6 possible station types:

<u>Percent of Mean Dwell Time</u>

K = I	1	2	3	4	5	6	7	8	9	10
1	40	60	70	80	90	100	120	130	140	170
2	24	48	59	75	85	94	111	126	155	223
3	30	47	65	77	90	103	116	137	157	178
4	0	29	59	75	92	108	125	148	170	194
5	0	18	36	70	104	125	144	156	167	180
6	0	0	0	48	96	120	144	171	198	223

If the user is exercising the NETSIM model and wishes to alter or replace these distributions, he may enter such data on card type 150. Although card type 150 may only be used with NETSIM, changes to these distributions are applied in <u>all</u> subnetworks.

HDWPCT(J)	GLR033	Vehicle Queue Discharge Headways. The HDWPCT array contains factors needed to determine the proper headway for a vehicle on a link of a particular "type" as defined on card type 11. Elements of HDWPCT may be changed when the NETSIM model is executed by using card type 149. Up to 4 link types may be accommodated.

As each queued vehicle moves up to the stop line, it is assigned a delay until discharge, in tenths of a second, reflecting queue discharge headways. This headway is obtained by multiplying the mean queue headway specified for the link, by a percentage. This percentage is extracted from a decile distribution which applies to that "type" of link. The vehicle's driver characteristic is used as an index for reflecting the proper element in the distribution. The index J to HDWPCT is calculated as: J = 10(I-1) = K, where i denotes the "type" of link and K is the vehicle's driver characteristic (a number 1 through 10). There are no internal default distributions for HDWPCT where I = 3 and 4. These may be created by the user through the use of card type 149. If NETSIM card type 149 is used to alter or add distributions for additional link types, these link types may also be used for links in a LEVEL I subnetwork via card type 11.

Distribution of Discharge Headway Percentages

K:	1	2	3	4	5	6	7	8	9	10
Type 1:	170	120	120	110	100	100	90	70	70	10
Type 2:	180	140	120	110	100	90	80	70	60	50

Note: The minimum discharge headway permitted by the program is 1.2 seconds. If the computed value obtained by multiplying an entry of a distribution by the specified link-specific mean value of discharge headway is less than 1.2, then the program will set the value to 1.2 seconds.

LEFTSP GLR059

Moving vehicles unimpeded by others must slow as they approach an intersection if they are to negotiate a turning maneuver. The speed for left turners, applied deterministically, is 22 feet/second.

This value may be changed when the NETSIM model is executed by using card type 140. When card type 140 is used, the new value of LEFTSP is applied in all relevant models.

LSTME(J) GLR032

The first vehicle in queue when the signal turns to GREEN experiences a (start-up) lost time. Lost time, in tenths of a second, is computed by referencing a decile distribution defined by the link "type", as specified on card type 11. The LSTME array may contain up to 4 such distributions, one for each of 4 link "types". Elements of the LSTME array contain percentage values applied to the specified mean lost time; these distributions may be changed when the NETSIM model is executed by using card type 149.

Distribution of Lost-Time Percentages

Driver Characteristic =	1	2	3	4	5	6	7	8	9	10
Type 1:	218	140	125	118	102	86	78	63	47	23
Type 2:	258	190	143	114	95	76	57	38	29	0

There are no internal default values specified for link types 2 and 4. If the user is exercising the NETSIM model and wishes to specify these distributions for additional link types (beyond 2), he must enter this data in card type 49. This card type may also be used to alter the default values for link types 1 and 2.

If card type 19 is not input as described above, then the user must specify one of these two link types on card type 11 for NETSIM links.

SECHWY GLR073

When a vehicle is second in queue when the signal turns GREEN, the headway obtained from array HDWPCT is incremented by the surcharge SECHWY. This value is 5 (tenths of a second).

STOPLT(I) GLR040

Array of service rates, vehicles per hour, for vehicles discharging from an approach controlled by a STOP sign, by executing a left-turn or through maneuver. The index, I, of the STOPLT array is calculated as I = min [IVOL/200+1,10], where IVOL is the total hourly volume on the [uncontrolled] cross street approach(es). Vehicles discharging from an approach controlled by a STOP sign, by executing a left-turn or through maneuver. The index, I, of the STOPLT array is calculated as I = min [IVOL/200+1,10], where IVOL is the total hourly volume on the [uncontrolled] cross street approach(es). Vehicles discharging from an approach controlled by a STOP sign, by executing a left-turn or through

		maneuver. The index, I, of the STOPLT array is calculated as I = min [IVOL/200+1,10], where IVOL is the total hourly volume on the [uncontrolled] cross street approach(es). Index, I: 1 2 3 4 5 6 7 8 9 10 Rate: 900 660 520 410 320 260 200 150 120 100
STOPRT(I)	GLR041	Array of service rates, vehicles per hour, for vehicles discharging from an approach controlled by a STOP sign, by executing a right-turn maneuver. The index, I, of the STOPLT array is calculated as I = min (ITHRU/200+1,7), where ITHRU is the hourly through volume on the inside lane of the [uncontrolled] near cross approach. Index, I: 1 2 3 4 5 6 7 Service Rate: 900 470 370 290 230 185 145
SYIELT(I)	GLR042	Array of service rates, vehicle per hour, for vehicles discharging from an approach controlled by a YIELD sign by executing a left-turn or through maneuver. The index, I, of the SYIELT array is calculated as I = min [IVOL/200+1,10] where IVOL is the total hourly volume on the [uncontrolled] cross street approach(es). Index, I: 1 2 3 4 5 6 7 8 9 10 Rate: 1200 880 690 545 425 345 265 200 160 130
SYIERT(I)	GLR043	Array of service rates, vehicles per hour, for vehicles discharging from an approach controlled by a YIELD sign by executing a right-turn maneuver. The index, I, of the SYIERT array is calculated as I = min [ITHRU/200+1,7], where ITHRU is the hourly through volume on the inside lane of the [uncontrolled] near cross approach. Index, I: 1 2 3 4 5 6 7 Service Rate: 1200 630 490 390 305 245 190
THRHWY	GLR075	When a vehicle is third in queue when the signal turns GREEN, the headway obtained from array, HDWPCT is incremented by the surcharge, THRHWY. this value is 2 (tenth-of-a-second).
TRKLDR	GLR081	To reflect the more sluggish operating characteristics of trucks, relative to autos, headways obtained from HDWPCT array are multiplied by the percent stored in TRKLDR. Currently this value is 120.
UFPCT	SIN231	As each vehicle enters a link, it is assigned a free-flow speed. This assignment is obtained by multiplying a percentage by the specified mean free- flow speed for that link. This percentage is obtained from a decile distribution where index I, is the driver characteristic code. The distribution is stored in the UFPCT array: **Distribution of Free-Flow Speed Percentages** I = 1 2 3 4 5 6 7 8 9 10 75 81 91 94 97 100 107 111 117 127 These values may be altered when the NETSIM model is executed by using card type 147. If the values are altered, via card type 147, they are applied to both the NETSIM and LEVEL I models. Note: The maximum value permitted by the program is 127 ft/sec. If the calculated value obtained by multiplying an element of this

		distribution by the specified link-specific mean value of free-flow speed exceeds 127, then the program will set the value to 127.
VEHLNG	GLR034	Effective length of vehicle type, J, which includes a 3-foot inter-vehicle spacing.

1.5 GLOBAL Traffic Assignment Database

The following arrays and scalars are used by the Traffic Assignment model. Variables that do not have a common block name listed are defined in the TA_LINKS module. The variables in TA_LINKS are not exported for external use. However, all other variables are exported via the listed common block.

Name	Common	Description
DNODES		Array of destination/volume pairs for an origin node.
IACCUR	GLR405	Line search accuracy for the objective function
IDFNC	GLR402	Type of impedence function 0 for FHWA impedence function 1 for Davision impedence function
IDNODS (I)		Destination node/demand pairs for an origin node, I
IDOBJ	GLR403	Type of optimality 0 for user equilibrium assignment 1 for system optimization assignment
IOFLG	GLR419	2-digit flag for printouts and simulation cards 1st 0 do not print intermediate path assignments 1 print intermediate path assignments 2 print tree constructs of each iteration 3 print detailed trees for each O-D pair 4 print all intermediate results 2nd 0 do not write simulation cards 1 write simulation cards 2 write simulation cards with all turn movements as percentages
IOUT	GLR418	Flag (0/1) do not/do print traffic assignment end results
IOUTIT	GLR417	Maximum number of capacity calibration
IPSLON	GLR404	Threshold for the gap between objective function and bound
ITERC	GLR416	Capacity calibration iteration number
KCFCTR	GLR421	Smoothing factor for capacity calibration
LEN (IL)		Length of geometric-network link, IL
LFRE (IL)		Freeflow travel time of geometric-network link, IL
LIST (IN)		Next path-network node in shortest path tree on current path-network node, IN
LNKP (IN)		Link number leading to specify path-network node, IN, on shortest path tree

MAXITA	GLR420	Maximum number of iterations to be performed by the equilibrium Traffic Assignment logic. This variable is referred by name JMXITA in subroutine TRAFIC.
MGTP (IL)		Mapping of path-network node number on geometric-network link, IL
MPTG (IN)		Mapping of geometric-network link number on path-network node, IN
NDS (IL)		Downstream node number on path-network link, IL
NFPNT (IN)		Forward-star pointer on path-network node, IN
NGDN (IL)		Downstream node of geometric-network link, IL
NGEXT	GLR409	Total number of exit nodes in geometric-network
NGLK	GLR406	Total number of geometric-network links
NGNET	GLR408	Total number of entry nodes in geometric-network
NGSRSK	GLR407	Total number of source/sink nodes in geometric-network
NGUP (IL)		Upstream node of geometric-network link, IL
NINT	GLR410	Total number of internal links in geometric-network
NP(IL)		Temp used to store source/sink node number
NPENTL	GLR413	Total number of entry links in path-network
NPEXTL	GLR414	Total number of exit links in path-network
NPLKS	GLR411	Total number of path-network links
NPNODS	GLR412	Total number of path-network nodes
NPORG	GLR415	Total number of origin nodes in geometric-/path-networks
NLKPNT(I)		Pointer to 1st link no. of the subnetwork
RATGC (IL)		G/C ratio on path-network link, IL
RCOST (IN)		Minimum travel time to path-network node, IN
RFLOW (IL)		Link flow (VPH) of path-network link, IL
RTRAV (IL)		Travel time on path-network link, IL
RTZERO (IL)		Free flow travel time on path-network link, IL
RVCUM (IL)		Total flow (VPH) on path-network link, IL
SATFLW (IL)		Saturation flow (VPH) on path-network link, IL
ZALFA	GLR400	First parameter for the applied impedance function
ZBETA	GLR401	Second parameter for the applied impedance function
ZCAP (IL)		Link capacity (VPH) of path-network link, IL
ZSPED (IL)		Estimate average speed on geometric-network link, IL
ZTRN (IL)		Turning code at downstream node on path-network link, IL
ZTRAVL (IL)		Total travel time (including delay) on path-network link, IL

WTA			Logical flag indicating that Traffic Assignment is being performed.
TA_IMXGLK			Scalar used in dynamically dimensioning arrays
TA_IMXPLK			Scalar used in dynamically dimensioning arrays
TA_IMXPND			Scalar used in dynamically dimensioning arrays
TA_IMXDST			Scalar used in dynamically dimensioning arrays
TA_IMXSUB			Scalar used in dynamically dimensioning arrays
CCBUF	GLR452		Character buffer used to store input card image for one data card at a time during first diagnostic pass. Buffer size is 80 characters.
NEXTP	GLR453		Code (0,1) if another Time Period (does not, does) follow current Time Period.
XBUF (J)	GLR451		Array used to store input card data for card being processed.
NLRT		NETSIM	Number of right turn pockets.
		GLOBAL	(same)
NLLT		NETSIM	Number of left turn pockets.
		GLOBAL	(same)
NLCO		NETSIM	Number of closed lanes.
		GLOBAL	(same)
DISCHA		NETSIM	Mean queue discharge headway (in sec*10).
		GLOBAL	(same)
OPPOFG		NETSIM	Opposing link number.
		GLOBAL	(same)
XCYCLE		NETSIM	Maximum possible cycle length.
		GLOBAL	(same)
IGRNR		NETSIM	Duration of green plus amber time for right turn movement.
		GLOBAL	(same)
IGRNT		NETSIM	Duration of green plus amber time for thru turn movement.
		GLOBAL	(same)
IGRND		NETSIM	Duration of green plus amber time for diagonal turn movement.
		GLOBAL	(same)
IGRNL		NETSIM	Duration of green plus amber time for left turn movement.
		GLOBAL	(same)

1.6 GLOBAL Real-Time Traffic Control Database

The following arrays and scalars are used by the Real-Time Traffic Control logic. These arrays are exported.

Name	Common	Description
CNETCF	GLR182	Character - The pathname where NETSIM reads and writes to the control files.
LU21	GLR888	Logical unit number 21, Real-Time: Time invariant data.
LU22	GLR889	Logical unit number 22, Real-Time: Time Period specific data.
LU23	GLR890	Logical unit number 23, Real-Time: Time Interval specific data.
LU24	GLR891	Logical unit number 24, Real-Time: Second-Specific data.
LU25	GLR892	Logical unit number 25, Real-Time: Second-Specific debugging data.
LU26	GLR893	Logical unit number 26, Real-Time: Second-Specific Flag file.
LU27	GLR894	Logical unit number 27, Real-Time: Second-Specific dumps flag file.
LU28	GLR895	Logical unit number 28, Real-Time: Second-Specific signal data file.
FLREAL(4)	GLR896	Array of flags specifying which files are requested by the user to be created to interface NETSIM with a real-time traffic control algorithm. **Word** / **Content** 1 Zero if NETSIM is not communicating with a real-time control algorithm. Otherwise, this word contains the number of seconds between transfers of information between NETSIM and a real-time control algorithm. 2 Flag (T, F) if NETSIM (should, should not) write a file of Time-Interval specific Measures of Effectiveness. 3 Flag (T, F) if NETSIM (should, should not) write a file of Time-Period specific Measures of Effectiveness. 4 Flag (T, F) if NETSIM (should, should not) write a file of second-by-second debugging data.
XSEED3	GLR897	Base random number seed used to generate emission headways.
ALPHAE	GLR180	User selected value of Alpha for Erlang distribution.
YDIST	GLR181	Flag (T, F) if (normal, shifted exponential/Erlang distribution) is used.

1.7 PRG Commons Database

The following arrays and scalars are PRG common variables. These commons are temporary storage of variables. However, they do not store one item for the entire simulation time. They are have different uses (different definitions) depending on which model is using the variable. These arrays and scalars are exported.

Listed below are the PRG common variable, the common number, and the description of the common in each model that uses the variable.

Name	Common	Model	Description
MGTP	PRG001	GLOBAL	Mapping of geometric links to path links.

GLKPNT(ISUB)	PRG003	GLOBAL	Index to first link of a subnetwork in global link arrays.
MPTG	PRG004	GLOBAL	Mapping of geometric links to path nodes.
		GLOBAL	List of internal centroid numbers.
NFPNT	PRG005	GLOBAL	Temporary storage of forward star pointer for path links.
MXPNOD	PRG018	GLOBAL	Maximum number of path nodes.
MXPLNK	PRG018	GLOBAL	Maximum number of path links.
MXDEST	PRG018	GLOBAL	Maximum number of destinations per origin.
MXGLNK	PRG018	GLOBAL	Maximum number of links in geometric network.
NGLK	PRG020	GLOBAL	Number of global links in network.
NGSRSK	PRG020	GLOBAL	Number of source/sink nodes in network.
NGENT	PRG020	GLOBAL	Number of entry nodes in network.
NGEXT	PRG020	GLOBAL	Number of exit nodes in network.
NGTURN	PRG020	GLOBAL	Number of turn movements in network.
NPLKS	PRG021	GLOBAL	Number of path network links.
NPNODS	PRG021	GLOBAL	Number of path network nodes.
NPENTL	PRG021	GLOBAL	Number of entry links in path network.
NPEXTL	PRG021	GLOBAL	Number of exit links in path network.
NPORG	PRG021	GLOBAL	Number of origin nodes in network.
OUTIT	PRG022	GLOBAL	Maximum number of capacity calibrations.
IOUT	PRG022	GLOBAL	Flag (0,1) if assignment results (will not, will) be printed.
IOFLG	PRG022	GLOBAL	Two digit flag for output: 1st digit: 0 - do not print intermediate path assignments. 1 - do print intermediate path assignments. 2 - print tree contructs of each iteration. 3 - print detailed trees for each O-D pair 4 - print all intermediate results (1,2 and 3). 2nd digit: 0 - do not write simulation cards. 1 – write simulation cards with all turn movements as volumes. 2 - write simulation cards with all turn movements as percentages.
MAXITA	PRG022	GLOBAL	Maximum number of iterations.
CCFCTR	PRG022	GLOBAL	Carry-over capacity calibration factor.
ALFA	PRG023	GLOBAL	First parameter of travel impedance function.

BETA	PRG023	GLOBAL		Second parameter of travel impedance function.
IDFNC	PRG023	GLOBAL		Type of impedance function
				(0=FHWA, 1=Modified Davidson).
IDOBJ	PRG023	GLOBAL		Type of traffic assignment
				(0=user's optimal, 1=system optimal).
EPSLON	PRG023	GLOBAL		Terminating threshold for objective function iterations (in .1%).
ACCUR	PRG023	GLOBAL		Accuracy level for line search of objective junction (in .1%).
ZPI	PRG025	NETSIM		Pi, equal to 3.14159265.
		GLOBAL		(same)

1.8 Global Links Database

The following arrays are dynamically allocated and defined in the GLOBAL_LINKS module. The variables in the module are exported and can be externally referenced as GLOBAL_LINKS_mp_XXX, where XXX represents the listed name of the variable. These arrays are for temporary storage of variables and DO NOT store one item for the entire simulation time. They are have different uses (different definitions) depending on which model is using the variable.

Name	Model		Description
GPRHB(I)	NETSIM		Turn movement prohibition codes.
	GLOBAL		Temporary storage of source/sink nodes.
XGPRMV(I)	ALL		Permitted movement codes.
XTEMP(I)	NETSIM		Permitted movement codes.
	GLOBAL		Link capacities.
	GLOBAL		Link capacities and sign or signal information.
GSPEED(I)	ALL		Free flow speed in ft/sec.
GLANE(I)	GLOBAL	1-3	Number of full lanes.
		4-5	Number of right turn pockets.
		6-7	Number of left turn pockets.
CNTRYD(I)	GLOBAL		List of internal centroid numbers.
	GLOBAL		Temporary list of exit nodes.
UPASS(I)	GLOBAL		Link number of underpass.
GTRKCP(I)			In TA, packed truck and car pool percentages for entry links, I (zero for non-entry links).
		1-7	Percentage of car pools entering at origin; origin must. be an entry node.
		8-15	Percentage of trucks entering at origin; origin must. be an entry node.
GMAP(I)			Global link numbers, I, for links in freeway segments.
NDSPL(I)	GLOBAL		Downstream node of path link and turn type.

IFZPTR(I)	NETSIM		Temporary storage of duration of intervals.
SRCSNK(I)	GLOBAL		Temporary storage of centroids.
GRT(I)	GLOBAL		Right receiving link.
	NETSIM		Right receiving link
	GLOBAL		Temporary storage of destination nodes.
	FRESIM		Temporary storage of destination nodes.
GTHRU(I)	ALL		Through receiving link.
	GLOBAL		Temporary storage of TOT attractions to exit nodes.
GUPNOD(I)	ALL		Upstream node number.
XGLEN(I)	ALL	1-14 15-21	Link length. Right pocket length. Left pocket length.
GDIAGN(I)	GLOBAL		Diagonal receiving link.
	NETSIM		Diagonal receiving link.
	GLOBAL		Temporary storage of origin/volume pairs for a destination.
GDWNOD(I)	ALL		Downstream node number.
	NETSIM	1-3 4-5 6-7	Number of full lanes. Number of right turn pockets. Number of left turn pockets.
	GLOBAL		Number of lanes.
	FRESIM		Number of lanes.
GLEFT(I)	ALL		Left receiving link.
	GLOBAL		Turn movement prohibition codes.
LEVCRV(I)	NETSIM		Curvature code (0,1, 2) if link has (no curvature, clockwise curvature, counter-clockwise curvature).
	GLOBAL		(same)
SCRCHA(I)	GLOBAL		Temporary storage of exit nodes.
	GLOBAL		Temporary storage of link sequence for route.
	GLOBAL		Scratch array used for temporary storage of various variables.
SCRCHB(I)	GLOBAL		Temporary storage of entry and exit nodes.
	GLOBAL		Scratch array used for temporary storage of various variables.
INMAP(I)	NETSIM		Mapping to user specified node numbers.
	FRESIM		(same)
CNTLNK(I)	GLOBAL		Global link numbers of centroids.
	GLOBAL		Temporary storage of sequence of stations for route

GLOBAL	Temporary storage of list of entry and exit nodes.
GLOBAL	Temporary storage for various variables.
GLOBAL	Global link numbers of centroids.

2 FRESIM Database

The FRESIM Database is presented by the following groupings:

- Scalars
- Link Data
- Node Data
- Vehicle Data
- Vehicle Type Data
- Bus Data
- Detector Data
- Fuel Consumption and Vehicle Emission Data
- Detector Data
- Algorithm Data
- Data Station Data
- Detector Data Station Data
- Incident Data
- Interface Data
- Metering Data
- Miscellaneous Data
- Object Data
- Segment Data
- HOV Data

2.1 FRESIM Scalars

This section contains all the scalars in the FRESIM database. These scalars are exported.

Name	Common	Description
BFSCAN	PRI270	Bus identification number scanner.
FDIFER	PRI372	Percent change in subnetwork occupancy during previous time interval.
FEATBL	PRI200	Number of acceleration elements in the environmental tables.
FETTBL	PRI201	Number of vehicle types in the environmental tables.
FEVTBL	PRI202	Number of velocity elements in the environmental tables.
GMAX	PRI424	Maximum number of gore arrivals (100).

INCDET	PRI181	Number of detectors for providing point processing information.
INCEVL	PRI146	Number of time steps between incident detection evaluations.
INCPRD	PRI148	Time between evaluations.
INDALG	PRI231	On-line incident detection algorithm number.
IRALGN	PRI251	Counter for number of lane realignments.
LKAHD	PRI296	Distance the driver will look ahead to respond to geometric changes.
MAXFBS	PRI464	Maximum number of buses allowed in FRESIM.
MAXFLK	PRI033	Maximum allowable links in the subnetwork.
MAXFVH	PRI001	Highest vehicle identification number that can be used in this subnetwork.
MAXGD	PRI416	Maximum number of detectors.
MAXHDY	PRI365	Maximum headway for ramp metering.
MAXNOD	PRI072	Maximum number of nodes in the subnetwork.
MAXRDW	PRI309	Maximum allowable number of roadways per segment.
MINHDY	PRI366	Minimum headway for ramp metering.
MOEEVL	PRI154	Evaluation frequency for MOE estimation and point processing in seconds.
MTRDET	PRI124	Maximum number of detectors serving a metering signal.
MTRLK	PRI123	Maximum number of links with detectors serving a meter.
MXADDP	PRI076	Maximum number of lane add/ drops per link (3).
MXALG	PRI235	Maximum number of incident detection algorithms.
MXALN	PRI292	Total number of alignments not including the on-ramp anticipation.
MXDATS	PRI083	Maximum number of freeway data stations.
MXDTST	PRI115	Maximum number of detector data stations.
MXENTR	PRI257	Maximum number of entry links.
MXEXIT	PRI262	Maximum number of exit links.
MXFBUF	PRI375	Current dimension of all buffer arrays (trajectory and detector buffer arrays).
MXFDET	PRI094	Maximum number of detectors.
MXINCD	PRI135	Maximum number of incidents.
MXMETR	PRI122	Maximum number of metering signals.
MXOBJ	PRI175	Maximum number of geometric objects in each segment.
MXSEG	PRI169	Maximum number of freeway segments.
MXVHID	PRI272	Highest vehicle identification number that is used in the subnetwork.
NUMAX	PRI420	Maximum number of actuations that can be stored. (Used in gap acceptance metering.)

NUMBER	PRI371	Number of actuation/deactuation records stored in the buffer.
OBJCNT	PRI328	Number of objects ahead of the vehicle in the look-ahead distance.
PCFSEP	PRI327	Pitt car following separation constant.
PERCNT	PRI405	Percentage of drivers passing the meter which the indication is no-go. Current value is 0.05.
POLFR	PRI206	Polling frequency for digital detectors (number/second). Default value is 10.
TADDP	PRI207	Total number of lane adds/drops in the subnetwork.
TINCST	PRI254	Total number of stations used for incident detection.
TLFILK	PRI213	Total number of internal links in the subnetwork.
TLFIEN	PRI212	Total number of entry interface links in the subnetwork.
TOTDT	PRI208	Number of detectors defined in the subnetwork.
TOTINC	PRI209	Total number of incidents in subnetwork.
TOTOBJ	PRI256	Total number of entries in the freeway geometry list.
TOTSEG	PRI210	Total number of freeway segments in the subnetwork.
TTDATS	PRI224	Total number of freeway data stations in the subnetwork.
TTDRPW	PRI325	Total number of lane drop warning signs.
TTDTST	PRI223	Total number of detector stations in the subnetwork.
TTINCW	PRI323	Total number of incident warning.
TTLFBS	PRI211	Highest bus identification number defined in the subnetwork.
TTLIEX	PRI246	Total number of entry-interface links in the subnetwork.
TTLFLK	PRI215	Total number of links in the subnetwork.
TTLFND	PRI214	Total number of nodes used in the subnetwork.
TTLFVH	PRI216	Highest vehicle identification number used in the subnetwork.
TTLMET	PRI217	Total number of signal meters in subnetwork.
TTLENL	PRI248	Total number of entry links in the subnetwork.
TTLEXL	PRI247	Total number of exits links in the subnetwork.
TTVEHP	PRI373	Total number of vehicles on the subnetwork during the previous time interval.
VFSCAN	PRI271	Vehicle identification number scanner.
WANT	PRI315	Flag (true, false) if the anticipatory lane changing logic (is, is not) requested.
WCNSRT	PRI431	Flag .T. if the vehicle identification array must be sorted before processing the next vehicle. Set to .T. when vehicle enters via a freeway connector.
WDTMOD	PRI230	Flag (.F., .T.) if detector operation mode is (digital, analog).
WEQUIL	PRI219	Flag (.T., .F.) if equilibrium (was, was not) achieved during last equilibrium evaluation interval.
WINCOF	PRI234	Flag .T. if off-line incident detection is desired.

WINCON	PRI151	Flag .TRUE. if on-line incident detection is requested.
WMOE	PRI158	Flag .T. if MOE estimation is desired.
WPOINT	PRI159	Flag .T. if point processing is desired.
WPPOUT	PRI180	Flag .T. if special point processing output is desired.
YDFSTM	PRI543	Flag true if first time to data dump routine.
YVSVOT	PRI511	Flag TRUE if vehicle type specific output is requested.
ZADVTH	PRI186	Threshold for advantage factor in discretionary lane changing.
ZCLOCK	PRI306	Elapsed time since the beginning of the simulation (secs).
ZDEC	PRI387	Decision making period when approaching a meter.
ZDISCR	PRI192	Calibration parameter to control number of lane changes.
ZDSTRH	PRI316	Speed threshold in random lane changing.
ZEDMPS	PRI542	End position on the segment for data dumping.
ZGRAV	PRI393	Gravity model sensitivity factor.
ZHIETA	PRI321	Hiatus-period. Current value is 3 seconds.
ZLAGAC	PRI190	Lag to acceleration, in seconds. Current value is 0.3.
ZLAGDC	PRI191	Lag to deceleration, in seconds. Embedded value is 0.2.
ZMNHDW	PRI304	Lower bound for headway in lane changing.
ZMNHDY	PRI194	Minimum headway for vehicle emission. (Below this value, a vehicle being emitted will seek another lane, if available.
ZMXCO	PRI283	Maximum deceleration for cooperative drivers. Current value is -8 fpss.
ZMXHDW	PRI303	Upper bound for headway in lane changing.
ZMXNCO	PRI284	Maximum deceleration for non-cooperative driver. Current value is -5 fpss.
ZPDYLD	PRI195	Percentage of drivers desiring to yield right-of-way. Embedded value is 0.4.
ZPTCNT	PRI269	Elapsed time since the last intermediate FRESIM output.
ZSDMPS	PRI541	Starting position for data dump.
ZTHSD	PRI305	Speed threshold factor used for headway comparison in discretionary lane changing (currently set to 7.5 seconds).
ZTSPD	PRI386	Speed below which vehicle is assumed to be in a queue.
ZVEHLN	PRI150	Assumed average vehicle length for incident detection.
INITIM	FOCC03	Time equilibrium was reached.
NTHLD	FOCC09	Number of thresholds for multiple occupancy threshold metering.
GRNTIME	FOCC10	Number of seconds of green time.
HOVPCT	PRI882	The percentage of qualifying HOVs that will use an HOV facility. Applies by default to all HOV facilities in the network.

2.2 FRESIM Link Database

All link-specific arrays presented in this section are one-dimensional with the generic subscript, IL. A few arrays in this group are link-and-lane specific, or link-and-vehicle type specific; these are identified explicitly. These arrays are dynamically allocated and defined in the FRESIM_LINKS module. They are exported and can be externally referenced as FRESIM_LINKS_mp_XXX, where XXX represents the listed name of the variable.

Name	**Description**
ADPCD(IDP)	ID of the IDPth lane add/drop on link IL. Internally (-1, 1) indicates (drop, add) As input on card 32, (2,1) indicates (drop, add), where IDP = (IL-1)*3+IDP.
ADPLN(IDP)	Lane ID of the IDPth lane add/drop on link IL, where IDP = (IL-1)*3+IDP.
ALIGN1(IL)	ID of the through lane on the link immediately downstream of link IL, that receives traffic from lane 1.
ALIGN2(IL)	ID of the lane on link IL which feeds lane 1 of the offramp immediately downstream of link IL, if any.
AUXCD(IL)	Code identifying the IAUXth auxiliary lane on link IL, as: 1- acceleration auxiliary 2 - deceleration auxiliary 3 - full auxiliary, where, IL = (IL-1)*3+IAUX.
AUXID(IL)	Lane number of the IAUXth auxiliary lane on link IL, where, IL = (IL-1)*3+IAUX.
CFNAME(IL)	User-defined name for link IL. 12 characters.
DRPWDX(J)	Where IK is the index to the lane drop warning sign, if, J=1 ID of lane drop; otherwise if J=2, the lane to be avoided due to a lane drop, where J= (IK -1)*2+J.
DWNODC(IL)	User-defined downstream node number for link IL.
DWNODE(IL)	Internal downstream node number for link IL.
FBUSES(IL)	Number of bus discharged.
FBZSTP(IL)	Total number of stops by buses.
FDATST(IL)	ID of data station on link IL.
FDETID(IL)	ID of first detector on link IL.
FINCID(IL)	ID of first incident on link IL.
FSMCNT(IL)	Sum of the content for link IL.
LANREC(IL)	Lane receiving through traffic.
LASTCT(IL)	Number of vehicles behind entry link.
LDSCHG(IL)	Total number of vehicles discharged from the link.
LKCNTN(IL)	Link content in number of vehicles.
LKFRIN(IL)	Index to segment specific arrays.
LKVHIN(IL)	Number of vehicles that have entered the link.

LNBAR(K)	ID of lane barrier on link IL, where K = 2 * (IL – 1) + N. N = 1 for the first barrier and N = 2 for the second barrier.
LNKFLW(IL)	User-specified entry link input volume, in VPH.
LNKINDX1(IL)	Starting point in the sorted vehicle list for this link.
LNKINDX2(IL)	Ending point in the sorted vehicle list for this link.
LNKTYP(IL)	Code identifying type of link IL (0-mainline, 1-ramp; sometimes 0-mainline, 1-onramp, 2-offramp).
MOVLFT(IL)	Number of vehicles moving left at the end of the link.
MOVRIT(IL)	Number of vehicles moving right at the end of the link.
MOVTHR(IL)	Number of vehicles moving through at the end of the link.
NADPLK(IL)	Identification number of add/drop.
NLNCHG(IL)	Counter for the number of lane changes on link IL.
OFFRMP(IL)	ID of the first link of the offramp emanating from link IL.
PAVMNT(IL)	Pavement code for link IL.
THRULK(IL)	ID of the first link downstream of link IL.
THRULN(IL)	Number of through lanes on link IL.
TRKDIR(IL)	Code identifying the direction to which trucks are biased.(0-right, 1-left).
TRUKCD(IL)	Code (0,1,2) for link IL, if trucks are (unrestricted, biased, or restricted).
TRUKLN(IL)	Lane ID to which trucks are biased/restricted. If more than on lane is biased or restricted, this lane is next to the unrestricted lanes.
UPNODC(IL)	User-defined upstream node number for link IL.
UPNODE(IL)	Internal upstream node number for link IL.
VCONT(IL)	Content of link IL at the end of the previous time interval.
VEHBEH(IL)	Highest number of vehicles stored behind the link (applicable to entry link only). Currently used for buses only.
WENTR25(IL)	Logical flag (T,F) if this RT25 (is, is not) a duplicate for this link during this time period..
ZADPOS(IDP)	Distance, in feet, between the IDP^{th} lane add/drop on link IL and the upstream node of link IL, where IDP = (IL-1)*3+IDP.
ZAUXLT(IL)	Length of the $IAUX^{th}$ auxiliary lane on link IL, where IL = (IL-1)*3+IAUX.
ZCURVE(IL)	Radius of curvature on link IL.
ZCFSM(IL)	Car following sensitivity factor.
ZDELAY(IL)	Total delay on link IL in seconds.
ZDISCH(IL)	Mean startup delay on link IL.
ZDPWRN(IDP)	Distance, in feet, between the IDP^{th} lane add/drop on link IL and its most upstream warning sign, where IDP = (IL-1)*3+IDP.
ZEMHDY(IL)	Emission headway for entry links (veh/sec).

ZETPCT(I)	Percentage of vehicles assigned to lane L for entry link IL where I = (IL-1)*5+L.
ZFFLOW(IL)	Free flow speed of link IL, in ft/sec.
ZGRADE(IL)	Grade of link IL in percent.
ZLAREA(IL)	Area of link IL in lane-miles. Includes through lanes and full auxiliary lanes.
ZLENTH(IL)	Length of link IL in feet.
ZLTANT(IL)	On-ramp anticipatory distance (feet).
ZLSTOP(IL)	Stop time incurred on the link. (seconds).
ZMNSPD(IL)	Acceleration lane minimum speed for upstream anticipatory lane changes.
ZPCRPL(IL)	Percentage of carpool vehicles.
ZPOFF(IL)	Percentage of traffic exiting at the downstream end of link IL.
ZEXPCT(IL, ITYP)	Percentage of traffic consisting of vehicle type ITYP exiting at the downstream end of link IL.
ZPTHRU(IL)	Percentage of traffic moving through at the downstream end of link IL.
ZPTRK(IL)	Percentage of trucks emitted from the entry link.
ZSUPEL(IL)	Superelevation of link IL.
ZTEMSN(IL)	Time until the next vehicle emission.
ZTRVL(IL)	Total travel time of vehicles on link IL, in seconds.
ZVDIST(IL)	Total distance traveled on link IL, in feet.
ZVMSSPEED(IL)	VMS speed. Can be set by using export function PutVMSSpeed.
ZVTRVL(IL,ITYP)	Total travel time by vehicle performance type and link.
ZWRNOF(IL)	Distance, in feet, to the most upstream offramp warning sign for the offramp at the end of link IL.
ZBSDLY(IL)	Delay time for buses.
ZBSDST(IL)	Distance traveled by buses.
ZBSSTP(IL)	Stop time for buses.
ZBTRVL(IL)	Travel time by buses.
ZDIST(I)	Distance traveled by car performance types on link IL, where I = (IPTYP-1)*200+IL.
ZEBUF(I)	Fuel consumption and emission data stored for each link and vehicle type. J = RESOURCE CODE 1 FUEL 2 HC 3 CO 4 NOX I = ((J – 1) + 4 * (ITYP – 1)) * TTLFLK + IL
ZFUELT(I)	Fuel consumption by link and vehicle type.

	I = (ITYP - 1) * TTLFLK + IL
NHOVLN(IL)	Number of lanes in the HOV facility on link IL.
HOVLAN(IL, 3)	Lane IDs for the lanes in the HOV facility.
HOVLOC(IL)	Location code.
HOVTYP(IL)	Type code.
HOVCOD(IL)	Lane use code.
HOVBEG(IL)	Beginning location.
HOVEND(IL)	End location.
HOVWN1(IL)	Location of the warning sign for the beginning.
HOVENT(IL, 3)	Number of vehicles that were in the HOV lane when they crossed the beginning of the facility.
HOVEXT(IL, 3)	Number of vehicles that were in the HOV lane when they crossed the end of the facility.
ZVILAT(IL)	Percentage of HOV lane violators on entry link IL.
ZHTRVL(IL, ILANE)	Total travel time of HOV vehicles in lane ILANE on link IL, in seconds.
ZSTRVL(IL, ILANE)	Total travel time of SOV vehicles in lane ILANE on link IL, in seconds.
ZHVLAT(IL, ILANE)	Total travel time of violators in lane ILANE on link IL, in seconds.
ZHDIST(IL, ILANE)	Total distance traveled by HOV vehicles in lane ILANE on link IL.
ZSDIST(IL, ILANE)	Total distance traveled by SOV vehicles in lane ILANE on link IL.
ZHSTOP(IL, ILANE)	Stop time incurred by HOV vehicles on link IL, in seconds.
ZSSTOP(IL, ILANE)	Stop time incurred by SOV vehicles on link IL, in seconds.
ZHDLAY(IL, ILANE)	Delay time incurred by HOV vehicles on link IL, in seconds.
ZSDLAY(IL, ILANE)	Delay time incurred by SOV vehicles on link IL, in seconds.
ZHOCCS(IL, ILANE)	Total occupancy in the HOV portion of lane ILANE on link IL.
ZSOCCS(IL, ILANE)	Total occupancy in the SOV portion of lane ILANE on link IL.
ZHVPCT(IL)	The percentage of qualifying HOVs that will use a specific HOV facility.
ZHWARN(IL)	The position of the HOV warning sign that a vehicle has crossed.
FREUSN(IL)	Upstream node number of the link (Not exported) .
FREDSN(IL)	Downstream node number of the link (Not exported).

2.3 FRESIM Node Database

All node-specific array in this section have the generic subscript IND. These arrays are exported.

Name	Common	Description
FAPRCH(IND)	PRI073	Link ID for the [mainline, onramp] approach to node IND, where IND = (IND -1)*2 + [1,2].
NFMAP(IND)	PRI075	User-defined node number for node IND.

2.4 FRESIM Vehicle Database

These arrays contain data required for all vehicles on the subnetwork. All vehicle-specific arrays are one-dimensional with the generic subscript IV. These arrays are dynamically allocated and defined in the FRESIM_VEHICLES module. They are exported and can be externally referenced as FRESIM_VEHICLES_mp_XXX, where XXX represents the listed name of the variable.

Name	Description
BUSIV(IV)	Bus ID number.
DRVTYP(IV)	Driver characteristic code.
FLEET(IV)	Fleet component. 0-Auto 1-Truck, 2-Carpool, 3-Bus.
FOLOW1(IV)	ID number of the follower of vehicle IV.
FPROBE(IV)	Flag (1,0) indicating the vehicle (is, is not) a probe vehicle.
FREGVH(IV)	Global ID of vehicle in FRESIM network.
FATISACCESS(IV)	Integer representing ATIS accessibility.
FCOMPLIANCEFACTOR(IV)	Degree of compliance with VMS instructions.
LNKPRV(IV)	Link previously occupied.
ONRAMP(IV)	When vehicle is merging (as in an onramp merge) the vehicle will set its speed to the speed of the adjacent lane. This array holds the lane ID that when reached, the vehicle will reset its desired speed.
USNFRE(IV)	Upstream node number of the link the vehicle is on.
VANTDX(IV)	Index to the last anticipatory warning sign crossed by the vehicle in the object list.
VCAND(I)	Candidate lane code for lane ILN and vehicle IV, where I = (IV-1)*11 +ILN: 0-acceptable lane 1-must be vacated 2-does not exist 3-Anticipatory lane 4-Incident blocks candidate lane 5-Incident blocks non-candidate lane.
VDETCH(IV)	ID of detector vehicle IV is currently activating.
VDRPDX(IV)	Index to the last lane drop warning sign crossed by the vehicle in the object list.

VDRVRC(IV)	Driver code of vehicle. Values range from 0 (timid) to 10 (aggressive).
VEHCDE(IV)	Code (0,1) if vehicle (has, has not) been processed for this time step.
VEHDST(IV)	Link ID of either the first link of the offramp or the exit link of the mainline for which the vehicle is destined.
VEHIDF(IV)	Vehicle ID of the IVth vehicle. Ordered by position, downstream to upstream order.
VEHINDF(IV)	Position of vehicle in the sorted array VEHIDF.
VEHLDO(IV)	Object ID of the next object in the object list after the last one crossed by vehicle IV. This is where the search for objects to cross begins for the next timestep.
VEHLNK(IV)	ID of the link on which vehicle IV currently is found.
VEHORG(IV)	Link ID of the origin of this vehicle.
VEHRDW(IV)	Roadway ID on which vehicle IV currently is found.
VEHSEG(IV)	Segment ID on which vehicle IV currently is found.
VEHTYP(IV)	Vehicle type of vehicle IV.
VFSTAT(IV)	Vehicle status code 0 - if not behind a meter, otherwise: 1- Stopped in a queue 2- Moving in a queue 3- Stopped behind a signal 4- Moving past a signal.
VHINCB(IV)	Incident ID of an incident that IV is currently crossing.
VINCDX(IV)	Object ID of incident affecting vehicle IV.
VLANE(IV)	Lane ID on which vehicle IV currently is found.
VLEAD1(IV)	ID of the primary leader of vehicle IV.
VLEAD2(IV)	ID of the secondary leader of vehicle IV. (Valid only when IV is the first vehicle upstream of the gore of a multiple destination lane and is exiting at that point.).
VOFFDX(IV)	Index to the last offramp warning sign crossed by the vehicle in the object list.
WEMDEC(IV)	Vehicle emergency decleration flag. True if vehicle movement is influenced by the blockage.
ZACCEL(IV)	Current acceleration of vehicle IV in ft/sec/sec.
ZACCP(IV)	Acceleration of vehicle IV in the previous time step.
ZADSPD(IV)	Speed adjustment for lane change.
ZCHTIM(IV)	Time vehicle has left to complete the lane change.
ZENTIM(IV)	Entry time to the current link.
ZFSPD(IV)	Desired speed in ft/sec.
ZIMPT(IV)	Lane changing impatience factor.

ZLCHTM(IV)	If greater than zero, time until its previous leader completes its lane change. Otherwise it is set to 0 (leader not changing lanes).
ZPRTIM(IV)	Time in seconds vehicle IV should be processed for this time step. Usually this value is equal to the timestep or zero, except when crossing segment boundaries.
ZSPEED(IV)	Speed of vehicle IV in ft/sec.
ZTLRSP(IV)	Vehicle tolerable speed in ft/sec.
ZVDSCH(IV)	Time to discharge from queue, in seconds.
ZVHDST(IV)	Distance in feet from the front bumper of the vehicle to the most downstream position on the freeway segment.
ZVRUB(IV)	Rubbernecking factor.
FRELNK(IV)	Counter for a path following vehicle, indicating which link in the path the vehicle is currently on.
FREPID(IV)	Path ID for a path following vehicle.
ZHOVWN(IV)	Location of the warning sign for the beginning of an HOV facility that vehicle IV has crossed.
ZHOVPS(IV)	Location of the beginning of an HOV facility that corresponds to the warning sign that vehicle IV has crossed.
VILATR(IV)	Flag indicating that vehicle IV is or is not an HOV lane violator, 0=no, 1=yes.
WENTER(IV)	Flag indicating that vehicle IV should keep considering if it should enter an HOV facility.
VHOVDX(IV)	Index to a warning sign for the beginning of an HOV facility that vehicle IV is approaching.
LNTYPE(IV)	Flag indicating the type of lane that vehicle IV is currently located in, 0=SOV, 1=HOV.
WRENTR(IV)	Flag indicating that vehicle IV should try to re-enter an HOV facility that it was previously forced out of because of an incident.
VHOFFX(IV)	The index to the HOV warning sign that a vehicle has crossed.
WTPGFLG(IV)	Flag indicating that two vehicles can discharge during the green.

2.5 FRESIM Vehicle Type Specific Database

These arrays contain the performance characteristics and fleet component percentages for up to 16 different types of vehicles as specified by the user. These arrays are all one-dimensional, with the generic subscript ITYP. This subscript corresponds to the vehicle type index specified in Entry 1 of Card Type 71. These arrays are exported.

Name	Common	Description
VACTBL(ITYP)	PRI407	Vehicle performance index, IPTYP, for vehicle type ITYP.
VTYPLN(ITYP)	PRI239	Length in feet of vehicle type ITYP.
ZEMDEC(ITYP)	PRI277	Maximum emergency deceleration in ft/sec^2.
ZNEMDEC(ITYP)	PRI480	Maximum non-emergency deceleration in ft/sec^2.
ZFLAUT(ITYP)	PRI242	Percentage of auto fleet represented by vehicle ITYP.

ZFLBUS(ITYP)	PRI244	Percentage of bus fleet represented by vehicle ITYP.
ZFLCAR(ITYP)	PRI363	Percent of carpool fleet represented by vehicle ITYP.
ZFLTRK(ITYP)	PRI243	Percentage of truck fleet represented by vehicle ITYP.
ZMNDEC(ITYP)	PRI280	Minimum acceptable risk for mandatory lane changing. (-8 ft/sec^2 for all).
ZMXDEC(ITYP)	PRI241	Maximum acceptable risk for mandatory lane changing. (-15 ft/sec^2 for all).
ZMXJRK(ITYP)	PRI240	Maximum acceptable jerk (7 ft/sec^3 for all types).
ZVHOCC(ITYP)	PRI468	Vehicle occupancy by the type of vehicle.
VTYPOC(ITYP)	PRI877	Number of occupants for vehicle type ITYP.

2.6 FRESIM Bus Database

These arrays contain data required for each bus vehicle, (IB), and bus route (IBR), on the subnetwork. These arrays are exported.

Name	Common	Description
FBUSRT(IB)	PRI460	Bus route ID number.
FPNTER(IBR)	PRI462	Pointer to maneuver array.
ZBSTIM(IBR)	PRI463	Time to next bus emission.

2.7 FRESIM Detector Database

All the detector related arrays are one dimensional with the subscript IDT (detector identification number). These arrays are exported.

Name	Common	Description
DETCD(IDT)	PRI095	Detector type code: 0-Doppler Radar 1-Short loop 3-Downstream loop of coupled pair 4-Upstream loop of coupled pair
DETCNT(IDT)	PRI335	Number of vehicles that have crossed this detector.
DETIME(IREC)	PRI370	Detection time of record IREC in detector file.
DETLK(IDT)	PRI249	Link ID of the detector.
DETLN(IDT)	PRI096	Lane number of the detector.
DETNUM(IREC)	PRI369	Detector ID number of the IRECth detection in detector file.
DETSTA(IDT)	PRI097	Station ID of detector IDT.
DTONOF (IREC)	PRI368	(0,1) if record IREC in detector file is an (on, off) pulse, or speed for Doppler detector.
DTSTUS(IDT)	PRI098	Status code of detector. 0-never activated

		1-deactivated 2-activated
ISAMP(I)	PRI413	Sample size at detector IDET for [count, speed, headway], where I = (IDET-1)*3 + [1,2,3].
NDETCP(IDT)	PRI103	ID of the other detector for a coupled pair.
NDETLK(IDT)	PRI104	ID of the next detector on the same link.
NDETST(IDT)	PRI105	ID of the next detector on the same station.
WACT(IDT)	PRI343	Flag .TRUE. if detector is activated.
FDTOCC(IDT)	FOCC01	Cumulative on time for detector IDT
OCCLON(IDT)	FOCC02	Time detector IDT was last actuated
WDTINC(IDT)	PRI107	Flag .TRUE. if the detector is used for incident detection.
ZDTDIS(IDT)	PRI111	Distance between couple loop detectors, in feet.
ZDTIME(IDT)	PRI352	Time of last vehicle crossing over the detector, in seconds.
ZDTLEN(IDT)	PRI112	Length of loop detector, in feet.
ZDTPOS(IDT)	PRI114	Distance between detector and upstream node of the link it is on (if loop, trailing edge).
ZDTSPD(IDT)	PRI350	Cumulative sum of speeds of vehicles crossing over the detector.
ZLSTOF(IDT)	PRI102	Last deactivation time in seconds.
ZLSTON(IDT)	PRI101	Last activation time in seconds.
ZLSTVU(IDT)	PRI414	Time that the last vehicle crossed the upstream detector of a coupled pair.
ZOCC1(IDT)	PRI099	Unscaled occupancy.
ZOCC2(IDT)	PRI100	Scaled down occupancy.
ZRAW(I)	PRI412	Point processing results, I = (IDT - 1) * 3 + J. J = 1 Cumulative sum of speeds. J = 2 Cumulative sum of headways.
FLVACT(IDT)	PRI732	ID of last vehicle to activate detector IDT.
ZFDOCC(DT)	FDET02	Average occupancy for the last DPPINT number of seconds.
ZFDSPD(DT)	FDET01	Average speed for the last DPPINT number of seconds.
ZFDVOL(DT)	FDET03	Average volume for the last DPPINT number of seconds.

2.8 FRESIM Fuel Consumption and Vehicle Emissions Data

These arrays contain fuel consumption and vehicle emission rate tables for varying acceleration, speed and vehicle performance index. Each array is the vector representation of an equivalent three dimensional array, where

IPTYP	Denotes 7 performance indexes
ISPD	Denotes 111 values of vehicle speed in the range of (0, 110) fps.

IACC	Denotes 21 different values of vehicle acceleration in the range of (-10, 10) fpss.

Hence, FFLCN array totals 21 * 111 * 7 = 16317 which is divided into 7 arrays of size 2331. FCOEM, FHCEM, and FNOEM array totals 21 * 111 * 7 = 16317 elements which is divided into 91 different arrays due to the 19 continuation lines restriction of the FORTRAN 77 compiler. These arrays are exported.

Name	Common	Description
FCOEM(I)	PRI376	CO emissions table. I = ((IPTYP-1) * 111 + ISPD) * 21 + IACC
FFLCN(I)	PRI377	Fuel consumption table. I = ((IPTYP-1) * 111 + ISPD) * 21 + IACC
FHCEM(I)	PRI378	HC emissions table. I = ((IPTYP-1) * 111 + ISPD) * 21 + IACC
FNOEM(I)	PRI379	NO emissions table. I = ((IPTYP-1) * 111 + ISPD) * 21 + IACC
ZACEM(I)	PRI299	Correction factor for acceleration for vehicle emission computation at grade. I = (IPTYP-1) * 111 + ISPD
ZAMAX(I)	PRI300	Maximum acceleration for vehicle performance type IPTYP and speed ISPD in ft/sec/sec. I = (IPTYP-1) * 111 + ISPD
ZAMXGD(I)	PRI301	Correction factor for maximum acceleration at a grade. I = (IPTYP-1) * 111 + ISPD

2.9 FRESIM Algorithm Database

These arrays contain data required for each algorithm parameter, (IALG), used in the subnetwork. These arrays are exported.

Name	Common	Description
MOEALG(IALG)	PRI236	Code, if >0, MOE estimation algorithm IALG is to be used.
ZINCPR(IALG, PARM)	PRI152	Parameter for algorithm.
ZMOEPR(IALG, II)	PRI160	Parameter values by algorithm.

2.10 FRESIM Data Station Database

These arrays contain data required for each freeway data station, (IDS), on the subnetwork. These arrays are exported.

Name	Common	Description
DTSTLK(IDS)	PRI233	Link ID number of the detector station.
HDWCNT(IHDX)	PRI084	Headway count by freeway data station, lane, and histogram category (ICT), where IHDX = ((IDS-1)*11+ILN))*20+ICT.
LNKSTN(IDS)	PRI085	Link ID at data station location.
ZDSPOS(IDS)	PRI090	Distance between data station and upstream node of link.

ZHDSUM(IDS)	PRI091	Cumulative headway sum, in seconds.
ZSPSUM(IDS)	PRI092	Cumulative speed sum, in seconds.
ZVTIME(IDS)	PRI093	Time last vehicle crossed data station.
SPDCNT(ISDX)	PRI087	Speed count by freeway data station, lane and category (ICT), where ISDX = ((IDS-1)*11+ILN))*20+ICT.
SPDSTN(IDS)	PRI341	Total vehicle count by data station.
SPDTTL(IDX)	PRI342	Total count of speed entries by lane by station, where IDX = (IDS-1)*11+ILN.

2.11 FRESIM Detector Data Station Database

These arrays contain data required for each detector data station, (IDTS), on the subnetwork. These arrays are exported.

Name	Common	Description
FDETST(IDTS)	PRI116	ID of first detector in station.
STAID(IDTS)	PRI232	User defined detector station ID number.
ZABSD(IDTS)	PRI163	Mean absolute deviation time series.
ZSQERR(IDTS)	PRI168	Sum of the squared error terms.
ZSERR(IDTS)	PRI165	Sum of error terms.
ZSMTH1(IDTS)	PRI166	Single exponentially smoothed time series.
ZSMTH2(IDTS)	PRI167	Double exponentially smoothed time series.
ZMOEDT(IDTS, IALG)	PRI433	MOE estimate of in-volume, out-volume speed and density.
ZMOETR(IDTS)	PRI432	Estimated travel time at the detector station.
ZOCPST(IDTS*5)	PRI117	Occupancy data for station for ITS time steps previously. The dimension is (IDTS -1)*5 +ITS.
ZINCOP(IDTS*3)	PRI118	Occupancy data for station before incident (II always =1). The dimension is (IDTS -1)*3 +II.
ZERR(IDTS)	PRI164	Series of estimation errors.

2.12 FRESIM Incident Database

This section describes all FRESIM variables for freeway incidents with the subscript IC (incident identification number). These arrays are exported.

Name	Common	Description
INCALG(IAG)	PRI145	Off-line incident detection algorithms to be applied.
INCCD(IC)	PRI136	Incident code for lane ILN, where IC =(IC-1)*11+ILN: 0- unaffected by incident. 1- rubbernecking. 2-lane blocked.
INCDUR(IC)	PRI137	Incident duration in seconds.

INCODE(IAL)	PRI153	Incident code between stations, where IAL = (IAL-1) * 50+IDTST.
INCSTN(IDTS)	PRI149	ID of station used for incident detection.
INCTME(IC)	PRI138	Incident onset time, in seconds.
INCWDX(INDEX)	PRI322	Code if lane should be avoided due to an incident.
		INDEX = (INCW - 1) * 12 + 12
NINCID(IC)	PRI139	ID of next incident on this link.
WINCFL(IC)	PRI140	Flag .TRUE. if incident in effect.
ZINCLT(IC)	PRI141	Length of incident, in feet.
ZINCPS(IC)	PRI142	Distance between upstream end of incident and the upstream node of the link it is on.
ZINCRB(IC)	PRI143	Rubbernecking factor.
ZINCWR(IC)	PRI144	Distance between upstream end of incident and warning sign, in feet.
ZINCWX(INCW)	PRI409	Distance between incident warning sign, (INCW), and incident.

2.13 FRESIM Metering Database

This section describes all FRESIM variables used in freeway ramp metering. They have the subscript IMTR (meter identification number). These arrays are exported.

Name	Common	Description
DMCPTY(IMTR)	PRI125	Freeway capacity in vehicles per hour.
GACT(IMTR)	PRI422	Pointer to the earliest gore arrival time in array ZGORE.
GWK(IMTR)	PRI423	Pointer to last stored arrival time in ZGORE.
LMVOL(IMTR)	PRI337	Last minute volume for demand/capacity meter.
MCLKTM(IMTR)	PRI126	Time in seconds, from the beginning of simulation, that metering commences.
NCLKTM(IMTR)	PRI726	Time in seconds, from the beginning of simulation, that metering commences in a subsequent time period.
MFACE(IMTR)	PRI295	Code indicating current meter signal.
MTRCNT(IMTR)	PRI336	Accumulated vehicle count for demand capacity meters.
MTRDID(IMTR)	PRI127	ID of detector serving meter, where IMTR = (IM - 1)*10+IMDT. IMDT = 1,10.
MTRLNK(IMTR)	PRI128	Link ID of detectors serving meter, where IMTR = (IMTR - 1)*10+IMDT. IMDT = 1,10.
MTRMAP(IND)	PRI074	Meter ID found at node IND.
MTRMV(IMTR)	PRI358	Number of vehicles moving through the meter this time step, where IMTR = (IMTR-1)*3+ILN.
MTRTOT(IMTR)	PRI404	Maximum number of signal cycles in this time step.
NCNTL(IMTR)	PRI130	Metering type code 1-Clock time

		2- Demand capacity 3- Speed threshold 5-Multiple-threshold occupancy control 6-ALINEA
NODMTR(IMTRR)	PRI339	Node ID where the meter is located.
NRAMPL(IMTR)	PRI348	Number of lanes at the meter.
NUMDTM(IMTR)	PRI402	Number of detectors associated with this meter.
NWK(IMTR)	PRI419	Pointer to the last stored actuation.
ZAVSPD(IMTR)	PRI344	Computed average speed for speed threshold metering. (Dimensioned larger than needed.)
ZDGORE(IMTR)	PRI428	Distance from detector to gore.
ZGORE(IMTR)	PRI421	Array of estimated gore arrival times, where IMTR = (IMTR-1)*100+GWK(IMTR).
ZMGORE(IMTR)	PRI427	Distance in feet from the meter to the gore.
ZMTHDY(IMTR)	PRI131	Metering headway.
RMTHDY(IMTR)	PRI731	Metering headway for a subsequent time period.
ZMTIME(IMTR, 5)	PRI359	Time having to do with meter actuations 5 per meter.
ZMTRT1(IMTR)	PRI133	Meter timer 1.
ZMTRT2(IMTR)	PRI134	Meter timer 2.
ZMTSPD(IMTR, DET)	PRI349	Sum of vehicle speeds across meter's detector.
ZSPHDY(IMTR)	PRI345	Metering rate for speed threshold metering, where IMTR = (IMTR - 1)*3 +ITHR. (ITHR = 1,3).
ZSPTHR(IMTR)	PRI132	(IMTR - 1)*3 +ITHR Speed thresholds for metering, where IMTR = (IMTR - 1)*3 +ITHR. (ITHR = 1,3).
ZOCTHR(INDEX+I)	FOCC04	Occupancy thresholds for multiple occupancy threshold metering. Where INDEX = (IMTR - 1) * 10, and I=1,10
ZOCMRT(INDEX+I)	FOCC05	Metering rates for multiple occupancy threshold metering . Where INDEX = (IMTR - 1) * 10, and I=1,10
MRUPDAT(IMTR)	FOCC06	Time interval in seconds at which metering rate will be updated.
MGRNTIM(IMTR)	FOCC07	Meter green time timer.
NVPGRN(IMTR)	FOCC08	Number of vehicles per green.
ZMRATE(IMTR)	FOCC11	Metering rates for ALINEA ramp metering.
ZKR(IMTR)	FOCC12	Input parameter used in ALINEA algorithm.
ZMOHAT(IMTR)	FOCC13	Input parameter used in ALINEA algorithm
ZMINMR(IMTR)	FOCC14	Minimum metering rates for ALINEA ramp metering.

2.14 FRESIM Miscellaneous Database

This section describes all FRESIM variables which do not belong to any other grouping. These arrays are exported.

Name	Common	Description
ZCDATA(JJ)	PRI294	Array used in car following.
ZDNAR(IGAP)	PRI415	Array of downstream detector events.
ZDRPWX(INDX)	PRI408	Distance, in feet, between the lane drop warning sign and its lane drop.
ZDTBUF(IBUF)	PRI374	Detector array storing trajectory data.
ZFOLK(IDRV)	PRI319	Following distance of driver in tenths of a second.
ZFRICT(IPVT)	PRI238	Friction coefficient by pavement type.
ZGAPTM(IGAP)	PRI426	Time for vehicles to reach the ramp gore.
ZGPUP(IGAP)	PRI417	Array of uncorrelated upstream arrivals.
ZUPAR(IGAP)	PRI429	Array of upstream detector events.
ZURGTH(IDRV)	PRI199	Urgency threshold (from .95 to .5).
ZVFH(IDRV)	PRI288	Percent of desired free flow speed by driver code.
ZLMNSP(I)	PRI287	Correction factor for speed in lane L on a link with NLANES through lanes, where I = (NLANES-2)*5+L. (Currently all values are set to 1).

2.15 FRESIM Object Database

This section contains arrays related to FRESIM objects. They have the subscript IOBJ (object list index). These arrays are dynamically allocated and defined in the OBJECTS module. They are exported and can be externally referenced as OBJECTS_mp_XXX, where XXX represents the listed name of the variable. However, OBVAHD is still exported via a common block.

Name	Common	Description
OBJDAT(IOBJ)		An index that references object specific data, depending upon the object type. See Object type chart.
OBJLNE(IOBJ)		Lane ID number affected by object IOBJ (or 99 for all lanes).
OBJNDX(IOBJ)		Position the object has in the object list.
OBJRDW(IOBJ)		Roadway ID of the object.
OBJSEG(IOBJ)		Segment ID of the object.
OBJTYP(IOBJ)		Code identifying the type of object. See object chart.
OBVAHD(IOBJ)	PRI297	Ordered list of object indexes IOBJ which the vehicle may cross this time step.
ZOBJPS(IOBJ)		Distance between object and the end of the segment (feet).

2.16 FRESIM Segment Database

This section contains arrays and scalars related to FRESIM segments. The arrays have various subscripts. These arrays and scalars are dynamically allocated and defined in the SEGMENTS module. They are exported and can be externally referenced as SEGMENTS_mp_XXX, where XXX represents the listed name of the variable.

Name	Description
BSEGOB(ISEG)	Position of the first object of this segment in the object list.
FRANK(LKFRIN(IL))	Link ID number.
FRDWY(LKFRIN(IL))	Roadway number of each link in FRANK array.
FRENSG(LKFRIN(IL))	Link ID of entry IENT of segment ISEG, where ISEG = (ISEG-1)*35+IENT.
FREXIT(LKFRIN(IL))	Link ID of destinations IEXT accessible from entry IENT of segment ISEG, where ((ISEG-1)*35+IENT -1)*35+IEXT.
LAENSG(IENT)	ID of entry link by freeway segment and roadway.
LEXSEG(INBF)	List of entries by freeway segment, where (ISEG-1)*35+IEXIT.
	INBF = (ISEG - 1)*MXEXIT + IEXIT
ZAGREN(INENT)	Gravity model balancing factors.
	INENT = (ISEG - 1) * MXENTR + IECNT
ZBATRC(INBF)	Attraction factor.
	INBF = (ISEG - 1)*MXEXIT + IEXIT
ZFREND(INEXIT)	Distance, in feet, between entry IENT and exit IEXT on segment ISEG.
	INEXIT = (IENT - 1)*MXEXIT + IEXT
ZFRENP(INDEXT)	O-D fraction for entry interface IENT going to exit IEXT.
	INDEXT = (IENT - 1)*MXEXIT + IEXT
RFRENP(INDEXV)	O-D fraction for vehicle type ITYP for entry interface IENT going to exit IEXT.
	INDEXV = ((IENT - 1)*MXEXIT + IEXT)*ITYP
ZFRPOS(LKFRIN(IL))	Distance to the end of the segment from the upstream node of link IL.
ZOFRAC(INBF)	Exiting fraction representing desirability of the destination.
	INBF = (ISEG - 1)*MXEXIT + IEXIT
ZRDWB(INDEX)	Beginning position of roadway on each segment.
	INDEX = (ISEG - 1) * MAXRDW + IRDW
ZRDWE(INDEX)	Ending position of roadway on each segment.
	INDEX = (ISEG - 1) * MAXRDW + IRDW
SEGLNK(ISEG)	Number of links in the freeway segment.
ALGN1(K)	K = (IALN-1)*11+ILN.
	Contains the lane number of the mainline downstream lane which is aligned with the upstream lane, ILN, in the IALNth lane alignment, where.
ALGN2(K)	K = (IALN-1)*12+ILN.
	IALN is the index of an alignment object associated with either an off-ramp warning sign or an off-ramp alignment. When ILN is less than 12 ALGN2 contains the lane ID of the lane on the downstream ramp that is aligned with lane ILN. When ILN equals

	12 ALGN2 contains either the upstream node of the first link of the off-ramp (when IALN refers to an off-ramp warning sign) or the roadway number of the off-ramp (when IALN refers to an off-ramp alignment).
ALGNDX(IALN)	Contains the object index of lane alignment number IALN.
ZODTABLE(I,J,K)	Calibrated OD table indexed by segment, entry and exit (ZOD74TABLE not included).
ZORG(I,K)	Origin volume indexed by segment and exit.
ZORGFACT(I,J)	Ration of calculated and desired origin volumes indexed by segment and entry.
ZDEST(I,K)	Destination volumes indexed by segment and exits.
ZOFACT1(I,K)	Saved ZOFACT volume indexed by segment and exit.
ZDESTFACT(I,J)	Ratio of calculated and desired destination volumes indexed by segment and entry.
ODNODE(I,J,K)	Index to accessible exit indexed by segment, entry and exit.
MAXEXITNODES(I)	Maximum number of exits on a segment.
ODRT74(I,J,K)	OD volume status to distinguish if this OD is an entry form RT74 indexed by segment, entry and exit.
ZOD74TABLE(I,J,K)	User specified OD table converted from RT74 input indexed by segment, entry and exit.
ORGURNODE(I,J)	User defined entry node ID indexed by segment and entry.
ODURNODE(I,J,K)	User defined node ID indexed by segment, entry and exit.
ZDESTTEMP(K)	Temporary array to store ZDESTFACT indexed by exit.
ZORGTEMP(K)	Temporary array to store ZORGTFACT indexed by exit.
NODEMAXEXIT(I)	Entry node which has maximum number of exits for a given freeway segment.
ORGLINK(I,J)	The entry link ID indexed by segment and entry.
ENTRYEXITMATCH(I,J)	The exit ID on another segment if this entry node is originated from indexed by segment and entry.
ENTRYSEGMATCH(I,J)	The segment index if this entry is originated from indexed by segment and entry.
MAXODLOOP	User defined maximum number of iterations for OD balance (Not exported).
MAXVOLLOOP	User defined maximum number of iterations for traffic volume exchange between freeway segments (Not exported).
REPS	Accuracy to perform OD iteration.
ZEPS	Accuracy to perform destination volume iteration (Not exported).

2.17 FRESIM HOV Database

This section contains data related to FRESIM HOV lanes. This scalar is exported.

Name	Common	Description
HOVTTL	PRI850	Total number of links that contain HOV facilities in the network.

3 NETSIM Database

The following array dimensions may be revised in NETSIM in order to create larger (or smaller) versions of the code.

Note: IMXGLK should be greater than or equal to IMXLNK.

Scalar	Located in:	Value
IMXVEH	Module NETSIM_VEHICLES	DYNAMIC
IMXLNK	Module NETSIM_VEHICLES	DYNAMIC
IMXNOD	NETSIM.INC	8999
IMXDET	NETSIM.INC	7000
IMXACT	NETSIM.INC	1000
IMXGLK	Module GLOBAL_LINKS	DYNAMIC
INMAX	GLOBAL.INC	6999
IMXBUS	NETSIM.INC	2000
IMXEVT	NETSIM.INC	200
IMXIND	NETSIM.INC	IMXNOD*12

The NETSIM database is presented by the following groupings:

- Scalars
- Link Data
- Node Data
- Vehicle Data
- Vehicle Type Data
- Vehicle Trajectory Data
- Bus Data
- Bus Station Data
- Surveillance Detector Data
- Event Data
- Section Data
- Subnetwork Calibration Data
- Fuel Consumption and Vehicle Emission Data
- Miscellaneous Data
- Actuated Controller Data

- Traffic Assignment
- Interchange Data
- Micro-node Data

3.1 NETSIM Scalars

This section contains all scalars in the NETSIM database. These scalars are exported.

Name	Common	Description
AGGFCT	SIN829	Driver type factor used to compute driver aggressiveness.
BEGDEC	SIN830	Deceleration at beginning of lane change maneuver for computation of acceptable risk.
BSCAN	SIN103	Bus identification number scanner.
CLOCK	SIN104	Elapsed time since beginning of simulation, secs.
COOPCT	SIN831	Percentage of drivers who cooperate with a lane changer.
DECFOL	SIN833	Deceleration rate of follower vehicle.
DECLED	SIN834	Deceleration rate of lead vehicle.
EATBL	SIN100	Number of acceleration elements in environmental tables.
EXSEED	SIN180	Actual number of entries in XLSEED array.
ETTBL	SIN101	Number of vehicle types in environmental tables.
EVTBL	SIN102	Number of speed elements in environmental tables.
HEDMAX	SIN835	Headway above which no drivers will attempt to change lanes.
HEDMIN	SIN836	Headway below which all drivers will attempt to change lanes.
HIGHTM	SIN372	Clock time when highest number of vehicles were on network.
HIGHVH	SIN371	Highest number of vehicles on network.
LANDUR	SIN837	Duration of NETSIM lane change maneuver.
LANWID	SIN999	Width of lanes, in feet. Currently set to 12.
LHORZN	SIN838	Longitudinal distance over which drivers decide to perform one lane change.
MAXBUF	SIN087	Dimension of DATBUF array.
MAXBUS	SIN010	Highest bus identification number that can be used on this subnetwork.
MAXCTD	SIN174	Maximum number of upstream entering movement codes and their associated downstream turn percents, or vehicle counts, that can be stored in CTDATA array.
MAXEVT	SIN079	Maximum allowable number of long term events.
MAXHED	SIN436	Maximum number of entries allowed in XHEDWY.
MAXICH	SIN849	Highest allowable interchange number.

MAXIND	SIN077	Maximum number of interval durations that can be stored in DURINT array.
MAXLCH	SIN839	Maximum allowable number of lane - changing vehicles.
MAXLNK	SIN016	Maximum allowable number of links in subnetwork.
MAXMIC	SIN865	Maximum allowable number of micronodes.
MAXND	SIN071	Maximum allowable number of nodes in subnetwork.
MAXTRJ	SIN881	Maximum number of rebs traversing a single trajectory.
MAXVEH	SIN001	Highest vehicle identification number that can be used on this subnetwork.
MAXVIN	SIN882	Maximum number of vehicles expected in micro-intersection.
MXFUEL	SIN170	Scalar used to compute dimension of fuel arrays, XDIST and XFUELT. This scalar is set equal to MAXLNK.
MXMOVE	SIN169	Dimension of each of the 10 link arrays associated with movement-specific output. Each turn movement specific MOE accumulator is dimensioned the same as other link arrays.
MXPARK	SIN168	Dimension of each of the 8 link arrays associated with parking activity. Each parking array is dimensioned to the same value as the other link arrays (i.e. MXPARK=MAXLNK).
MXSCLK	SIN710	Maximum number of links which may be contained within sections
MXSEC	SIN711	Maximum number of sections
MXSEED	SIN177	Dimension of XLSEED array.
NMAPRC	SIN864	Maximum number of approaches and receivers associated with a micro-node.
QFREQ	SIN192	Frequency at which average and maximum queues are collected (in secs).
RDCFOL	SIN840	Deceleration rate of follower vehicle.
RDCLED	SIN841	Deceleration rate of lead vehicle.
RHEDMX	SIN842	Headway above which no drivers will attempt to change lanes.
RHEDMN	SIN843	Headway below which all drivers will attempt to change lanes.
SAFACT	SIN844	Safety factor.
SUCTME	SIN845	Time required for successive lane changes.
STPTHR	SIN106	Threshold value for speed, as a basis for calculating "stop delay".
TDIFER	SIN107	Percent change in subnetwork occupancy (previous Time Interval) during initialization.

TOTCTD	SIN175	Number of upstream entering movement codes and their associated downstream turn percents, or vehicle counts, that were specified by user on Type 22 Card.
TOTDET	SIN109	Number of surveillance detectors specified for this subnetwork. The data for each detector is contained in arrays DTCTR1, DTCTR2 and DTCTR3.
TOTEVT	SIN108	Number of long term events that were specified for this subnetwork. The data for each event is contained in arrays, LEVNT1, LEVNT2 and LEVNT3.
TPTIME	SIN421	Elapsed time since beginning of Time Period (minutes).
TTLBUS	SIN111	Highest bus identification number used so far in this subnetwork.
TTLIEN	SIN112	Total number of interface entry links.
TTLILK	SIN113	Total number of internal links, including entry interface and exit interface links.
TTLINT	SIN114	Total number of entries in DURINT array.
TTLMIC	SIN892	Total number of micro-nodes in subnetwork.
TTLND	SIN115	Total number of nodes in subnetwork.
TTLNK	SIN116	Total number of links in subnetwork.
TTLVEH	SIN117	Highest vehicle identification number used so far in this subnetwork.
TVEHS	SIN118	Total number of vehicles on subnetwork of previous Time Interval during initialization.
UDTCNT	SIN119	Elapsed time (seconds) since last intermediate output.
URGTHR	SIN847	Urgency threshold.
VSCAN	SIN121	Vehicle ID number scanner.
XVTRPN	SIN123	Cumulative number of vehicle trips completed in subnetwork since beginning of simulation.
XVTRPP	SIN139	Number of vehicle trips completed in subnetwork since beginning of simulation to beginning of current time period.
YSEC	SIN164	Flag (T,F) if section-specific Measures of Effec-tiveness (are, not) to be provided.

3.2 NETSIM Link Database

All link-specific arrays presented in this section are one-dimensional with the generic subscript, IL. A few arrays in this group are link-and-lane specific, or link-and-vehicle type specific; these are identified explicitly. These arrays are dynamically allocated and defined in the NETSIM_LINKS module. They are exported and can be externally referenced as NETSIM_LINKS_mp_XXX, where XXX represents the listed name of the variable.

To conserve storage, certain arrays are defined differently between Internal and Entry Links. These arrays are marked with an asterisk.

Name	Description
ALIGN_LANE_DWLNK(IL)	1-3 Through lane number of subject link, IL, which is aligned with the lane on THRU(IL), that is specified in ALIGN_LANE_UPLNK(IL).
ALIGN_LANE_UPLNK(IL)	Lane number on the "through" receiving link (THRU(IL)), which is aligned with the lane specified in ALIGN_LANE_DWLNK(IL). Lane numbers are counted from curb to median (curb lane is lane 1).
ALIGN_LEFT_CODE(IL)	Code (0,1) if a left turn (is not, is) prohibited.
ALIGN_THRU_CODE(IL)	Code (0,1) if thru movement (is not, is) prohibited.
ALIGN_RGHT_CODE(IL)	Code (0,1) if a right turn (is not, is) prohibited.
ALIGN_DIAG_CODE(IL)	Code (0,1) if a diagonal turn (is not, is) prohibited.
ALIGNO(IL,ITRN,ILN)	Optional lane alignments. IL=link, ITRN=turn movement, ILN= lane.
ARIGHT(IL)	Number of link receiving right-turning vehicles from link, IL. The downstream node number of a receiving link is stored, instead, if it is an Exit Link.
BLOKL(ILN)	Lane Specific Array - Blocker distance form upstream node, feet.
BLKLOC_LANE(IL)	Lane in which blocker resides.
BLKLOC_DIST(IL)	Distance of blocker from upstream node, feet.
BLKR(IL)	Time remaining, sec., for blockage on link, IL.
BLOKT(ILN)	Lane Specific Array - Remaining time for blocker.
BSTIME(IL)	Total bus travel time on link, tenths-of-a-second.
BUSES(IL)	Total number of buses discharged from this link since beginning of simulation.
BUSESP(IL)	Number of buses discharged since beginning of simulation to beginning of current time period.
BUSTP(IL)	Total number of buses that stop at least once on this link.
CCNAME(IL)	Character array used to store the street name given to each link on card type 10. Each entry in the array can store a 12 character name.

CFAILP(IL)	Number of signal phase failures since beginning of simulation to beginning of current time period.
CPTPL(IL)	Number of left turning vehicles discharged from link, IL, since beginning of simulation to beginning of current time period.
CPTPR(IL)	Number of right turning vehicles discharged from link, IL, since beginning of simulation to beginning of current time period.
CNTENT(IL)*	Number of vehicles currently on link. For Traffic Assignment: Traffic volume on an entry link, vehicles/hour.
CROSFR(IL)	Identifying number of the far-cross link approach to downstream node.
CROSNR(IL)	Identifying number of the near-cross link approach to downstream node.
CTVECT(IL)	Pointer to CTDATA array for beginning of conditional turn movement data for link, IL.
CUMLC(IL)	Cumulative number of lane changes
CUMVEH(IL)	Number of vehicles discharged from link since beginning of simulation. (Includes vehicles discharged via a sink node).
CUMPTP(IL)	Number of vehicles discharged from link, IL, since beginning of simulation to beginning of current time period.
CUMVL(IL)	Number of left-turning vehicles discharged from link since beginning of simulation.
CUMVR(IL)	Number of right-turning vehicles discharged from link since beginning of simulation.
CYCFAL(IL)	Number of signal phase failures.
DIAGNL(IL)	Number of link receiving diagonal-turning vehicles from link, IL. The downstream node number of the receiving link is stored if it is an Exit Link. (This entry is set negative for a left-diagonal movement).
MEAN_Q_DISC_HWY(IL)	Mean queue discharge headway, sec * 10.
STRTUP_LOST_TIM(IL)	Start up (lost) time, tenths-of-a-second..
PED_CODE (IL)	Pedestrian code.

Code	Meaning
0	No pedestrian traffic.
1	Light pedestrian traffic.
2	Moderate pedestrian traffic.
3	Heavy pedestrian traffic.

MEAN_FF_SPEED(IL)	Mean desired free-flow speed, ft/sec.
IRTOR_CODE (IL)	Code (0,1) if RTOR (is, is not) permitted.

LINK_TYPE_CODE(IL)		Code identifying queue discharge charac-teristics (i.e. Link "type" code).
CLOSE_LANES(I,IL)		Code identifying closed lanes (1=closed) for all I=1,7 lanes that may exist on the link.
	14-20	Code identifying closed lanes (1=closed) for all 7 lanes that may exist on the link.
DSCSPD(ILL)		Code identifying discharge speed of vehicle currently first in queue. Speed = 5 * Code + 10 subject to free-flow speed limit, and maximum of 45 fps. When code = 0, Speed = 0.
DWNOD(IL)		Internal downstream node number.
ECOUNT(IL)		Number of vehicles which could not be admitted onto link IL from the source node and are now waiting entry.
EHDWY(IL)		Extraction headway (+,-), tenths-of-a-second, at a sink node. Value must be less than or equal to 10000 tenths-of-a-second.
ITIMEP(IL)		Aggregated travel time (sec) of vehicles on link since beginning of simulation to beginning of current time period.
IVFTP(IL)		Cumulative vehicle-feet of travel of vehicles on link, IL, since beginning of simulation to beginning of current time period.
LANEF(ILL)		Number of first vehicle in lane, K, of link, IL, where ILL=(IL-1)*7+K.
GRADE_CODE(IL)		Code identifying link grade. **Code** / **Grade(%)** 0 -2 to +2 1 +2 to +5 2 Above +5 3 -2 to -5 4 Below -5
NFULL_LANES(IL)		Number on full lanes on link.
NLANES_RTPOCK(IL)		Number of lanes in right turn pocket.
NLANES_LTPOCK(IL)		Number of lanes in left turn pocket.
NCURBLANES_RTURNERS(IL)		Number of curb lanes channelized for right turners.
NINSIDELANES_LTURNERS(IL)		Number of inside lanes channelized for left turners.
LANEL(ILL)		Number of last vehicle in lane K of link IL, where ILL=(IL-1)*7+K.
LANEV_STATUS(ILL)*		Information for an entry link stored in LANE_STATUS, where ILL=(IL-1)*7+K. Code indicating lane status. **Code** / **Meaning** 1 Lane empty 2 Lane occupied

		3 Lane does not exist 4 Lane closed or otherwise not available
LANEV_DATA(ILL)*		Information for an entry link stored in LANE_DATA, where ILL=(IL-1)*7+K. If the lane is empty then these bits contain the clock time, modulo 8000, tenths-of-a-second that the next vehicle can discharge, no sooner than this clock time. If the lane is occupied then these bits contain the vehicle identification number.
LEFT(IL)		Number of link receiving left-turning vehicles from link, IL. Downstream node number of receiving link if it is an Exit Link.
LGLPK(IL)*		Length of left turn pocket. Value is less than or equal to 1000 feet. <u>For Traffic Assignment</u>: Count of vehicles to be emitted before next car pool emission, tenths-of-a- vehicle.
LGRPK(IL)*	1-10	Length of right turn pocket. Value is less than or equal to 1000 feet. (if negative, the word is packed as follows):
	1-9	Length of a protected bus station which serves as a right turn pocket (feet). (If negative, the word is packed as follows):
	1-9	Length of a protected bus station which serves as a right turn pocket (feet).
	10-15	Station number of the protected station. <u>For Traffic Assignment</u>: Count of vehicles to be emitted before next truck emission, tenths-of-a-vehicle.
LINK(JL)		Array which maps between the user-specified sequence of links and the sequence generated internally by the program for simulation purposes. This internal sequence is completely transparent to the user. All link arrays described in the manual are sequenced according to the internally generated sequence of links. The properties of link number JL (according to user sequence) are located in all link specific arrays in location, IL (internal sequence number) where IL = LINK(JL).
LNDCH(ILL)		Cumulative number of vehicles discharged from lane K of link, IL, where ILL = (IL-1) * 7 + K
LNDCHO(ILL)		Cumulative number of vehicles discharged from lane K of link IL at the end of last time period. ILL = (IL-1)*7 + K.
NEXTE(IL)		Time elapsed since emission at sink/source node, tenths-of-a-second.

OPPOSE(IL)	Number of opposing link, i.e., link which services oncoming traffic which impedes left turning vehicles from link, IL.
PARKL_LEN(IL)	Length of parking zone on left-hand curb, divided by 8 (feet).
PARKL_DIST(IL)	Distance from stop-line to front of parking zone on left-hand curb, divided by 8 (feet).
PARKR_LEN(IL)	Length of parking zone on right-hand curb, divided by 8 (feet).
PARKR_DIST(IL)	Distance from stop-line to front of parking zone on right-hand curb, divided by 8 (feet)
PCTLR_LEFT(IL)	Percentage of vehicles on link making left turns.
PCTLR_RGHT(IL)	Percentage of vehicles on link making right turns.
PEDFED_CODE(IL)	Movement code (0,1,2,3,4) if the feeder link whose pedestrian flow blocks traffic on link IL is a (L,T,R,LD,RD) feeder link.
PEDFED_LINK(IL)	Feeder link whose pedestrian flow blocks traffic on link IL.
PEDTMR(IL)	Time since onset of pedestrian flow (secs).
PLSEED(IL)	Pointer to XLSEED array for beginning of random number seeds for link, IL.
PRKACT_REMTIMR(IL)	Time remaining to complete parking maneuver at right-hand curb, sec.
PRKACT_REMTIML(IL)	Time remaining to complete parking maneuver at left-hand curb, sec.
PRKDEM_MEANDUR(IL)	Mean duration of parking maneuvers, divided by 4 (tenth-of-a-second, i.e. resolution of $\pm$0.125 seconds).
PRKDEM_MEANINT(IL)	Mean interval between successive parking maneuvers on the link, divided by 2 (seconds, i.e. resolution of $\pm$ 1 seconds).
PRKLOC_DISRHS(IL)	Distance of parker from upstream node on right-hand side now executing maneuver, divided by 8 (feet).
PRKLOC_DISLHS(IL)	Distance of parker from upstream node on left-hand side now executing maneuver, divided by 16 (feet).
PRKTMR(IL)	Time to next parking maneuver, sec. (Can be negative).
PCT_THRU(IL)	Percentage of vehicles proceeding through.
PCT_DIAG(IL)	Percentage of vehicles on link making diagonal turns.
QAVG(K)	Average queue length on link IL, in lane ILN, where K = (IL-1)*7 + ILN.
QDELAY(IL)	Cumulative delay in queue, sec. since beginning of simulation.
QDELL(IL)	Cumulative delay in queue, seconds, experienced by left turning vehicles, since beginning of simulation.

QDELR(IL)	Cumulative delay in queue, seconds, experienced by right turning vehicles, since beginning of simulation.
QDLAYP(IL)	Delay of vehicles (secs) in queue on link, IL, since beginning of simulation to beginning of current time period.
QMAX(IL)	Maximum queue length on link IL, in lane ILN, where K = (IL-1)*7 + ILN.
RVILAT(IL)	Percentage of SOVs entering the network on entry link IL that are freeway HOV lane violators.
RVMSSPEED(IL)	VMS speed. Can be set by using export function PutVMSSpeed.
SIGNAL(IMOV,IL,TSCODE)	Signal code of the current signal controlling traffic on link IL for the traffic movement IMOV and time step code TSCODE. Where IMOV is defined as follows: 0 T_LEFT 1 T_THRU 2 T_RGHT 3 T_LDIA 4 T_RDIA Where TSCODE is defined as follows: 1 S_PREV (previous time step) 2 S_CURR (current time step)
AMBTIM(ITRN, IL)	Remaining yellow time by turn movement for link IL. (Used by the external control logic)
SHEVNT_MEANDUR(IL)	Mean duration of short term events. Value stored is less than or equal to 60 seconds.
SHEVNT_FREQ(IL)	Inter-arrival gap of short term events. Value stored is less than or equal to 500 seconds.
SIGHT(IL)	Forward sight distance at stop-line of link IL.
SINK(IL)*	Number of vehicles extracted at sink node. <u>For Traffic Assignment or Entry Link</u>: Percentage of all emitted vehicles on this link which are car pools.
SINKP(IL)	Number of vehicles extracted at sink node on link, IL, since beginning of simulation to beginning of current time period.
SOURCE(IL)*	Number of vehicles emitted at source node. <u>For Traffic Assignment or Entry Link</u>: Percentage of all emitted vehicles on this link which are trucks.
SOURCP(IL)	Number of vehicles emitted from source node on link, IL, since beginning of simulation to beginning of current time period.

SPLBK_TIME(IL)		Time of onset of spillback, secs. modulo (4000 secs.)
SPLBK_TIMER(IL)		Timer for left-turn laggers (decremented each second).
SSNODE(IL)		User node number of Source/Sink node servicing link, IL.
STOPP(IL)		Number of vehicles discharged that stopped at least once on link, IL, since beginning of simulation to beginning of current time period.
STOPL(IL)		Cumulative number of discharged left turning vehicles forced to stop at least once.
STOPR(IL)		Cumulative number of discharged right turning vehicles forced to stop at least once.
STPDL(IL)		Cumulative stopped delay time, sec, for left turning vehicles.
STPDLY(IL)		Cumulative stopped delay time, sec.
STPDP(IL)		Stopped delay time, sec, on link, IL, since beginning of simulation to beginning of current time period.
STPDR (IL)		Cumulative stopped delay time, sec., for right turning vehicles.
SUMCNT(IL)		Sum of link content, since beginning of simulation, divided by 10.
SUMCP(IL)		Sum of link content since beginning of simulation to beginning of current time period, divided by 10.
TEVENT(IL)		Time remaining until the next short term event vehicle is generated on link.
THRU(IL)		Through receiving link number or downstream node number of through receiver if it is an exit link.
TRVLL(IL)		Cumulative travel time, sec., for left turning vehicles discharged since beginning of simulation.
TRVLR(IL)		Cumulative travel time, sec., for right turning vehicles discharged since beginning of simulation.
TRVLTM(IL)		Total travel time, since beginning of simula-tion, of all vehicles discharged, sec.
TYPLN_BUSES(IL)		Lane channelized for buses only.
TYPLN_CARPOOL(IL)		Lane channelized for car-pool vehicles only.
UPNOD(IL)		Internal upstream node number.
VPCT(IL,4,16)		Vehicle-type specific turn multipliers.
VPROC(IL)		Sum of vehicle-feet for vehicles occupying link IL at the end of initialization.
VREM(IL)	1-8	Number of vehicles occupying the link at the end of the previous Time Period. Value stored is less than or equal to 256.
WONEWY(IL)		(T/F) if link represents a one-way street

XCANDL(IL, ITRN)

Candidate lane codes, based on two consecutive turn movements. These codes indicate which lanes are available for vehicles that are entering link IL if their turn movement when discharging from IL will be ITRN. ITRN ranges from 1 to 4, where 1=left, 2=through, 3=right, and 4=diagonal. The vehicle's next downstream turn movement determines which block of 7 bits represents the candidate codes for the lanes on link IL.

1-7	ITRN and LEFT. A "l" in bit position n indicates that lane n is accessible to vehicles that are making an ITRN turn when discharging from IL and will be making a left turn from the ITRN receiving link.
8-14	ITRN and THROUGH. A "l" in bit position n indicates that lane n is accessible to vehicles that are making an ITRN turn when discharging from IL and will be making a through movement from the ITRN receiving link.
15-21	ITRN and RIGHT. A "l" in bit position n indicates that lane n is accessible to vehicles that are making an ITRN turn when discharging from IL and will be making a right turn from the ITRN receiving link.
22-28	ITRN and DIAGONAL. A "l" in bit position n indicates that lane n is accessible to vehicles that are making an ITRN turn when discharging from IL and will be making a diagonal turn from the ITRN receiving link.

XDIST(K)

Array containing distance traveled (vehicle-feet) on each link since the start of simulation, stratified by vehicle type. This array is the vector (i.e. one-dimensional) representation of the equivalent array, XDIST(IL,I) where IL = link number and I = vehicle type = 1,2,3 (auto, truck, bus). Therefore: K = TTLILK * (I-1) + IL where TTLILK is the total number of internal links.

XEBUF(K)

Array of data describing resources consumed on each link by vehicle type. The array is internally stratified as XEBUF(IL, J, I) where,

IL = link number: 1,2,...,TTLILK
J = resource code:1,2,3,4(Fuel,HC,CO,NOX)
I = Vehicle type: 1,2,3 (auto, truck, bus)
K = [TTLILK * 4 * (I-1) + TTLILK] * (J-1) + IL

TTLILK is the total number of internal links.

XINT1(IL)

Contains the movements from the link which are permitted during signal intervals 1 through 6.

Bit Position:

15, 14	13	12	11	10, 9	8	7	6	5, 4	3	2	1
W	X	Y	Z	W	X	Y	Z	W	X	Y	Z
Interval 3				Interval 2				Interval 1			

Bit Position:

30, 29	28	27	26	25, 24	23	22	21	20, 19	18	17	16
W	X	Y	Z	W	X	Y	Z	W	X	Y	Z
Interval 6				Interval 5				Interval 4			

W = Code for left-turn movement

	X = Code for diagonal movement Y = Code for through movement Z = Code for right-turn movement
	Code **Movement** 0 Is permitted and protected (GO) 1 Is not permitted (NO GO) 2 Is permitted but unprotected (COND. GO); applies to left turn movement only
	Note: If XINT1(IL) = 30, approach is controlled by STOP sign; If XINT1(IL) = 31, approach is controlled by YIELD sign; If XINT1(IL) = 16, the control facing link IL is a perpetual green where left turners from link IL face opposing traffic; If XINT1(IL) = 0, the control facing link IL is a perpetual green where left turners face no opposing traffic. If an interval-specific 5-bit string equals 24, the interval is an amber clearance for the movements serviced during prior interval.
XINT2 (IL)	Contains the movements from link, IL, which are permitted during intervals 7 through 12. The format is identical to that for XINT1.
	Note: XINT2 need only be referenced if link IL is not controlled by a sign and if there are more than 6 intervals defined for the node at the downstream end of link IL.
XLNGTH1(IL)	Length of link, in feet. Must be less than or equal to 2000 feet.
XLNGTH2(IL)	This is the total number of lanes on the cross-street approaches to the upstream node of link IL.
XPERS(IL)	Cumulative count of people * 100 discharged from link since beginning of simulation. This value is incremented by the occupancy rate * 100 of each vehicle which discharges the link from the stopline or via a sink node.
XPINT1(IL)	Same as XINT1 except control code settings reflect inputs for a subsequent Time Period.
XPINT2(IL)	Same as XINT2 except control code settings reflect inputs for a subsequent Time Period.
XSTOP(IL)	Cumulative number of discharged vehicles forced to stop at least once.
PERCENT_VEH_LEFT(IL)	Percentage of vehicles turning left.
PERCENT_VEH_ THRU (IL)	Percentage of vehicles traveling through.
PERCENT_VEH_ RGHT (IL)	Percentage of vehicles turning right.
PERCENT_VEH_ DIAG (IL)	Percentage of vehicles turning diagonally.
XWIDTH(I,IL)	Two dimensional array where IL is the link and
I=1	Width of lane 1
I=2	Width of lane 2
I=3	Width of lane 3
I=4	Width of lane 4

	I=5	Width of lane 5
	I=6	Width of lane 6
	I=7	Width of lane 7
XWIDT2_WIDTH(IL)		Width of parking lane(s) (0 if none)
XWIDT2_DIST(IL)		Distance from stop-line to curb (ft)
XWIDT2_INTERNUM(IL)		Interchange number if link is part of an interchange
XWIDT2_ANGLE(IL)		Angle of link relative to due north
ZFACTL(IL)		Ratio of NETSIM/Animation Distance for left turns.
ZFACTR(IL)		Ratio of NETSIM/Animation Distance for right turns.
XLSED2(IP)		Random number seed for ENTRY links and internal links with source points. This seed is used to generate a random variation in emission headways. The index, IP is obtained for movement, IMOVE on ENTRY link, IL as: IP = PLSEED(IL) -1 + IMOVE The index, IP for an internal link, IL with a source point is: IP = PLSEED(IL).
CTDATA(IVP)		Upstream entering movement code (set to -1, -2, -3, -4 for left, through, right, diagonal) that specifies the first upstream movement for link, IL, that has conditional downstream turn percents, or vehicle counts, assigned. This code is set negative for the first upstream movement only. IVP is a pointer taken from the CTVECT(IL) array. Elements IVP+1, IVP+2, IVP+3, and IVP+4, contain the conditional downstream turn percentage (*100) or vehicle counts for (left, through, right, diagonal) movements respectively on link associated with upstream feeding movement in CTDATA(IVP).
VOLUME (IL)		Volumes of right/thru/diagonal/left movements at the downstream node on link IL
VOLSRC (IL)		Source volumes (VPH), on link IL
VOLSNK (IL)		Sink volumes (VPH), on link IL
DTFLNK(IL)		Detector identification number of the first detector on the referenced link.
DTMAXP(IL)		Distance between the trailing edge and the upstream node of the most upstream detector on the referenced link.
SAVE_ICHAN(IL)		Contains movements serviced by each lane for a link
RICDELAY(IL)		Total delay in queue (sec)
RICDLAYP(IL)		Total delay in queue (sec) start of TP
RICDELL(IL)		Left turn vehicle delay in queue (sec)
RICDELR(IL)		Right turn vehicle delay in queue (sec)

NETUSN(IL)	Upstream node number of the link (Not exported)
NETDSN(IL)	Downstream node number of the link (Not exported)

3.3 NETSIM Node Database

The node-specific arrays are one-dimensional with the generic subscript, IN. A few arrays in this group are related to the node number but have different subscripts; these are identified explicitly. These arrays are exported.

Name	Common		Description
DURINT(J)	SIN078	1-7	Duration of interval, I, at node IN where J = IP + I and IP = (IN – 1) * 12 + 1
		8-15	Reference offset (seconds) at node IN. This entry applies only when bits 1-7 contain the duration of interval one (i.e. I=1).
FNINT(IN)	SIN072		Number of signal intervals at the node, IN. (Set to 1 if uncontrolled or sign control).
GMIN(IN)	SIN188		Minimum allowable main street green duration (seconds) during transition.
NACT(IN)	SIN073		Code describing status of control at node IN: ACN if actuated, where ACN is the actuated controller number, set internally 0 if pre-timed signal control -1 if stop sign control or uncontrolled -2 if yield sign control
NMAP(IN)	SIN075		User-specified node number that corresponds to the internally-assigned NETSIM subnetwork node, number, IN.
PDURNT(J)	SIN181	1-7	New duration of interval I specified at node IN during subsequent Time Period. J = IP + I and IP = (IN – 1) * 12 + 1
		8-15	Reference offset (seconds) at node IN specified for a subsequent Time Period. This entry applies only when bits 1-7 contain the duration of interval one (i.e. I = 1).
PFNINT(IN)	SIN182		Number of signal intervals at node IN during a subsequent Time Period.
PNACT(IN)	SIN183		Code describing status of control for subsequent Time Periods. 0 = Fixed time controller, new signal settings specified for subsequent Time Period, and transition not yet begun. -9 = Control not fixed time or control is fixed time <u>and</u> transition period begun.
SIGI(J)	SIN076		Identification number of link IL, for approach IA, to node IN, where J = 5(IN-1) + IA; IA = 1,2,..., 5, determined internally.

SIGT(IN)	SIN074		Current signal status (interval and elapsed time) at node IN.
		1-4	Current interval number. (0 for perpetual green or stop sign or yield sign).
		5-12	Elapsed time that current signal interval has been active, sec.
		13	1, if node IN is to be modeled as a micronode
			0, otherwise
		14	External control flag
XCOORD(IN)	SIN136	1-15	X coordinate defining node position in feet/10.
		16-30	Y coordinate defining node position in feet/10.
XSGTRN(IN)	SIN187	1-7	Blank
			8-14 Main street green duration (seconds) during 1st cycle of signal transition.
		15-21	Main street green duration (seconds) during 2nd cycle of signal transition.
		22-28	Main street green duration (seconds) during 3rd cycle of signal transition.
YEXTCO(IN)	EXT001		Flag indicating that node IN is being externally controlled.

3.4 NETSIM Vehicle Database

These arrays contain data required for all vehicles on the subnetwork. All vehicle-specific arrays are one-dimensional with the generic subscript, IV. These arrays are dynamically allocated and defined in the NETSIM_VEHICLES module. They are exported and can be externally referenced as NETSIM_VEHICLES_mp_XXX, where XXX represents the listed name of the array.

Name		Description
ACCEL(IV)		Vehicle acceleration, fpss.
ACCODE(IV)		Code to identify whether vehicle is accelerating (0) or decelerating (1).
ARIVAL1(IV)		Arrival time at a control sign.
BLQUE(IV)		Set to (0,1) if vehicle (is not, is) in queue formed by blocker or parker.
DISTUP(IV)		Distance of vehicle, IV, from upstream node, feet (= traveled distance on current link).
ENNODE(IV)	1-25	Node from which vehicle, IV, entered network.
	26-31	Time period when vehicle entered network.
ENTIME(IV)		Time in tenths-of-a-second that vehicle, IV, entered network.
ENTRTM(IV)		Time in tenths-of-a-second that vehicle, IV, entered current link, relative to the start of simulation.

FOLOWR(IV)	Number of vehicle following this vehicle, IV, in the same lane and link.
KDRVRC(IV)	Used to store the original value of NDRVRC(IV).
LASTLNK(IV)	Link number occupied by vehicle in last second.
LCHGTM(IV)	Last time vehicle changed lanes
LCLEAD(IV)	Lead vehicle for lane change to force gap
LEADER(IV)	Number of vehicle leading this vehicle, IV, in current lane and link.
LKSTOP(IV)	Code set to 1 if vehicle stops on link currently occupied.
LSWCH(IV)	Set to (0,1) if vehicle (has not, has) switched lanes to bypass blocker.
NATISACCESS(IV)	Integer representing ATIS accessibility.
NCOMPLIANCEFACTOR(IV)	Degree of compliance with VMS instructions.
NLANE(IV)	Lane currently occupied.
NDRVRC(IV)	Driver code to indicate the driving habits of operator, IV; ranges from 0 (extremely cautious) to 9 (highly aggressive).
NBUSIV(IV)	Bus number (less than or equal to 256). If = 0, then vehicle is not a bus.
NETGVH(IV)	Global vehicle ID (for NETSIM vehicle, IV).
NETLNK(IV)	Counter for a path following vehicle, indicating which link in the path the vehicle is currently on.
NETPID(IV)	Path ID for a path following vehicle.
NFLEET(IV)	Code identifying vehicle fleet component:

Code	Vehicle Type
0	Auto
1	Truck
2	Car pool
3	Bus

NPROBE(IV)	Flag (1,0) indicating the vehicle (is, is not) a probe vehicle.
NVHCDE(IV)	Vehicle process code (0,1) if vehicle, (has not, has) been processed in the current time step.
NVHLNK(IV)	Link number currently occupied.
NVHTYP(IV)	Vehicle type. The range is from 1 to 16
PEDDLY(IV)	Time remaining for pedestrian delay.
PREVLN(IV)	Lane number occupied by the vehicle on previous link.
PRVLNKICD(IV)	Link number occupied by the vehicle on previous link.
PREVTC(IV)	Turn code on previous link.

Code	Movement
0	Left
1	Through

		2 Right 3 Left diagonal 4 Right diagonal 5 Source emission
PRFDLN(IV)		Preferred lane.
PRVDIST(IV)		Distance of vehicle from upstream node at the previous link.
PRVGLN(IV)	1-7	Vehicle's previous goal lanes before changed due to an intersection blockage
	8-10	Vehicle's previous turn code before changed due to an intersection blockage
PRVLNK(IV)		Vehicle's previous link
SPDICD(IV)		Control delay speed
SPDLN(IV)		Vehicle speed, ft/sec.
TCODE(IV)		Current turning code:

Turn Code	Movement
0	Left
1	Through
2	Right
3	Left diagonal
4	Right diagonal

TIMDIS(IV)		Time remaining to discharge from head of queue, seconds x 10.
TIMYLD(IV)		The number of seconds that a vehicle has been waiting to discharge because of spillback on the receiving link.
USNNET(IV)		Upstream node number of the link the vehicle is on.
VCHNG(IV)	1-3	Time remaining to check for lane change
	4-6	Interval between lane changes
	7	1; if vehicle not in goal lane
		0; if no goal lanes or in goal lane
	8-13	Not used
	14	1, if vehicle must slow to allow a forced lane changer in front
		0, otherwise
	15	1, if vehicle is a mandatory lane changer
		0, otherwise
VEHICD(IV)		Status code for control delay collection.
VEHICD0(IV)		Flag (T,F) if vehicle is within influence of control.
VFFSPD(IV)		Desired free-flow speed, fps. This value may be set to 125, temporarily, to flag a left-turn "jumper".
VHLANE(IV)	1-3	Target lane (set 0 if current lane OK)
	4-10	Goal lanes

	11	1, if driver is cooperative to a lane changer
		0, otherwise
	12	1, if vehicle is a bus which seeks a lane change because another bus is in dwell downstream and it does not service that station
		0, otherwise
	13-15	Time remaining to change lanes now
VIOLTR(IV)		Flag indicating that an SOV is a freeway HOV lane violator.
VLNRLK (IV)		Lane on receiving link
VLNCHG(IV)		Vehicle downstream of IV that is leaving IV's lane but still influences IV's movement.
VPATH_ICHG(IV)		ID of the interchange the vehicle is in.
VPATH_PATH(IV)		ID of the path the vehicle is following in an interchange.
VPATH_STEP(IV)		Step number in the path the vehicle is following in an interchange.
VRPATH(IV)		Index to XREBP array identifying current reb occupied by the vehicle iv (0 if vehicle is not within intersection)
VSTATE(IV)		Vehicle status code:

Code	**Meaning**
0	In motion or blocked by bus waiting entry to station or by vehicle awaiting entry to turn pocket
1	Stopped in queue
2	Moving in queue
3	Blocker or bus awaiting entry to station
4	Bus in dwell
5	In queue formed by inadequate storage capacity of a turn pocket
6	Stopped in queue and tagged for cycle failure
7	Moving in queue and tagged for cycle failure.

WILL_YIELD(IV)		Logical flag, indicating that the vehicle is waiting because of spillback on the receiving link and will yield to other vehicles entering the intersection.
XGOALN(JJ)		Three words are reserved to store the goal lanes for each link in the path for each vehicle as it approaches it's next turn, where JJ = (IV – 1) * 3 + I.
		I = 1
	1-7	Goal lanes on 1st link in path for vehicle IV
	8-14	Goal lanes on 2nd link in path
	15-21	Goal lanes on 3rd link in path
	22-28	Goal lanes on 4th link in path
	29-31	Number of links in path in word 1
		I = 2

		1-7	Goal lanes on 5th link in path for vehicle IV
		8-14	Goal lanes on 6th link in path
		15-21	Goal lanes on 7th link in path
		22-28	Goal lanes on 8th link in path
		29-31	Number of links in path in word 2
			I = 3
		1-7	Goal lanes on 9th link in path for vehicle IV
		8-14	Goal lanes on 10th link in path
		15-21	Goal lanes on 11th link in path
		22-28	Goal lanes on 12th link in path
		29-31	Number of links in path in word 3
XVSEED(IV)			Random number seed.

3.5 NETSIM Vehicle Type Specific Database

These arrays contain the performance characteristics and fleet component percentages for up to 16 different types of vehicles as specified by the user. These arrays are all one-dimensional, with the generic subscript, ITYP. This subscript corresponds to the vehicle type index specified in Entry 1 of Record Type 58. These arrays are exported.

Name	Common		Description
FLTAUT(ITYP)	SIN128		Percent of auto fleet which consists of vehicles of type, ITYP.
FLTBUS(ITYP)	SIN129		Percent of bus fleet which consists of vehicles of type, ITYP.
FLTPUL(ITYP)	SIN130		Percent of car pool fleet which consists of vehicles of type, ITYP.
FLTTRK(ITYP)	SIN131		Percent of truck fleet which consists of vehicles of type, ITYP.
VTYPAH(ITYP)	SIN132	1-7	Maximum acceleration at zero speed, fpss * 10 (Maximum value = 86 or 8.6 fpss).
		8-15	Factor applied to mean discharge headway, percent, I/2 (Maximum value is 250, i.e. 500 percent).
VTYPLD(ITYP)	SIN134		Average person occupancy * 100.
VTYPSP(ITYP)	SIN823		Maximum speed at zero acceleration, ft/sec.
VTYPLE(ITYP)	SIN135		Effective vehicle length, feet.

3.6 NETSIM Vehicle Trajectory Database

When vehicle fuel consumption and emission measures are requested, the program stores trajectory information in the DATBUF array for each vehicle on the subnetwork. This array is exported. The DATBUF array is constructed as follows:

Name	Common		Description
DATBUF(1)	SIN088		Number of entries (vehicle trajectory points) stored in DATBUF array. Value stored is less than or equal to 500.
DATBUF(2)			Elapsed time since beginning of simulation, seconds.

The following two words are stored for each vehicle on the subnetwork. IMAXV is defined as 2*DATBUF(1)+2 and is less than or equal to 502. When DATBUF is full or all vehicles have been processed the contents of DATBUF are written to LU32.

Name		Description
DATBUF(3,5,7,...IMAXV-1)		Vehicle trajectory word 1.
	1-16	Link Number.
	17-22	Distance of vehicle from upstream node in units with increments of 32 feet. That is distance = (feet) / 32.
DATBUF(4,6,8,IMAXV)		Vehicle trajectory word 2.
	1-7	Vehicle speed, feet/sec. Value is less than or equal to 128.
	8-10	Vehicle type code: **Code** — **Vehicle Type** 0 — Auto 1 — Truck 2 — Bus
	11-14	Vehicle acceleration/deceleration in fpss. (Less than or equal to 12).
	15	Set to (0,1) if the contents of bits 11-14 indicate (acceleration, deceleration).
		The DATBUF array is also used as a buffer to store lane-specific vehicle numbers in subroutine RESCHN. This information is then used to define the vehicle chains at the end of every second of simulation.

3.7 NETSIM Bus Database

These arrays contain data required for each bus vehicle, (IB), on the subnetwork. These arrays are exported.

Name	Common		Description
BDWELL(IB)	SIN816		Time remaining in dwell for bus IB.
BSWAIT(IB)	SIN025		Code (0,1) if bus, IB, (is not, is) waiting at a station to be discharged. Bus, IB, could not be discharged previously due to vehicles in desired lane.
BUSGL(IB)	SIN807	1-7	Goal lanes from VLANE array
		8	1, if station serviced on link
			0, otherwise
BUSRT(IB)	SIN011		Bus route number for bus IB.
PNTER(IB)	SIN012		Current value of index to MANUVR array identifying the next target, either bus stop or node, of bus, IB.

			Note: PNTER(IB) is set negative when bus IB has already serviced at least one station on the link but has no more stations to service on the link it currently occupies. If there are no stations on this link, PNTER(IB) is positive.

3.8 NETSIM Bus Station Database

These arrays contain data required for each bus station on the subnetwork. These arrays are exported.

Name	Common	Description
STALOC(IBS)	SIN808	NETSIM link number that station is located on.
STEMTY(IB)	SIN013	Cumulative time that station is empty, sec.
STUSE(IB)	SIN014	Cumulative time that station capacity is exceeded, sec.
STVOL(IB)	SIN015	Total number of buses serviced.

3.9 NETSIM Event Database

These arrays contain data required to define the status of all long-term events, (IVNT), specified by the user. These arrays are exported.

Name	Common		Description
LEVNT1(VT)	SIN080		Time when long-term event will be activated, measured from beginning of simulation, secs.
LEVNT2(VT)	SIN081		Time when long-term event will be deactivated, measured from beginning of simulation, secs.
LEVNT3(VT)	SIN082		Location of long-term event:
		1-3	Lane number
		4-15	Link number
LEVNT4(VT)	SIN713	1	Code (0,1) if event (is not, is) within intersection, entered on 55 card
		2-4	Lane number on cross street on curb side that locates the intersection blockage, entered on 55 card (0 if event occurs outside intersection)
		5-7	Previous lane number of vehicle path that blockage will emulate for animation (0 if event occurs outside intersection)
		8-10	Previous turn code of vehicle path that blockage will emulate for animation (0 if event occurs outside intersection)
LEVNT5(VT)	SIN714		Distance from upstream stop-line for vehicle that blockage will emulate for animation (0 if event occurs outside intersection)
LEVNT6(VT)	SIN715	1-3	Lane number on approach to micro-node that locates the reb to be blocked, from 55 card (0 if event occurs outside intersection)

		4-15	Link number of approach to micro-node that locates the reb to be blocked, from 55 card (0 if event occurs outside intersection)
LEVNT7(VT)	SIN716		Pointer to XREBP array identifying reb blocked within path of vehicle that blockage will emulate in animation (0 if event occurs outside intersection)

3.10 NETSIM Section Database

The user may wish to have the program analyze the aggregate performance of specific sets of links known as sections. The SECSHN array contains the information needed by the program to properly assemble each section. These arrays are exported.

Name	Common	Description
SECNUM(I)	SIN709	List of user defined section numbers.
SECSHN(J)	SIN163	List of links contained within sections. The last link in a section is stored as a negative. A link ID of zero means there are no more sections.

3.11 NETSIM Subnetwork Calibration Database

A substantial portion of the calibration database operated upon by the model is defined within the program in the form of "default" values. This data may be selectively revised for each run through the use of card types 140 through 150. These arrays and scalars are exported.

Name	Common	Description
AMBER(I)	SIN094	Amber Phase Response The response of drivers to the onset of the amber indication is expressed in terms of an acceptable deceleration. The attendant logic applies only to the lead moving vehicle in a lane which has no queue at the instant the signal turns amber. The deceleration that is required for the vehicle to stop is readily calculated knowing the current position and speed of the vehicle. Using the driver characteristic code, I, a decile statistical distribution, AMBER(I) is entered, to determine whether the acceptable deceleration extracted from this distribution exceeds the required value. If so, the vehicle will stop; otherwise it will continue through the intersection. The contents of this array, in fpss, are: I = 1 2 3 4 5 6 7 8 9 10 Accept. Decel. = 21 18 15 12 9 7 6 5 4 4 The values contained in the AMBER array may be altered using card type 144.
FSGAP(I)	SIN093	Additive factor for a stopped vehicle seeking a gap in far-side cross-street traffic. A vehicle at a stop line facing a STOP sign cannot discharge until an acceptable gap is available in the cross-street traffic. The decile distribution of acceptable gaps for near-side (or one-way) cross-street traffic is stored in the NSGAP array. The

acceptable gap in the traffic stream on the far-side cross-street approach must be larger than that on the near-side cross-street approach. This additional time is required by the vehicle to cross the width of the near-side cross-street approach.

The array FSGAP contains this additional time, in tenths-of-a-second, for a vehicle to reach the far-side cross-street approach, when the vehicle must effectively travel I lanes to cross the near-side cross-street approach.

I	=	1	2	3	4	5	6	7	8	9	10
Additional Time	=	12	21	26	31	35	39	42	46	49	51

The values contained in FSGAP may be altered using card type 143.

LSALAG SIN099 Lane-Switching Acceptable Lag

A vehicle cannot switch lanes unless an acceptable lag is available in the target lane. This value, deterministically applied, is 31 tenths of a second. To alter this value use card type 140.

LTJGAP(I) SIN098 Left-Turn Jumper Probabilities

A left-turn jumper is a vehicle that is first in queue when the signal changes to GREEN, and executes the left-turn maneuver (immediately) before the on-coming queues can discharge. An attempt was made to relate the probability of a lead left-turner jumping, to the number of lanes of on-coming traffic he had to cross. The data, however, revealed no statistically significant relation in this respect. An aggregated sample revealed that this probability is 0.38. The model retains the capability of relating this probability to the number of on-coming lanes. Each data item in the array, LTJGAP(I) is set to the probability of a lead left-turn vehicle jumping at the beginning of the GREEN phase across I on-coming lanes, expressed as a percentage. Presently, LTJGAP(I) = 38; I = 1,2,...,7. Elements of the LTJGAP array may be changed by using card type 140.

LTLAGP(I) SIN089 Left-Turn Lagger Turn Probability

This array contains the probability values that describe the left-turn lagger responses. A left-turn lagger is a queued vehicle who executes a left-turn across opposing traffic during the NO GO interval immediately following a left-turn GO (and AMBER) interval. If the left-turner is at the stop-line within 2 seconds after start of this NO GO interval, the probability (in percent) that he will execute the turn movement is stored in LTLAGP(1); if within 4 seconds, in LTLAGP(2); if within 5 seconds, in LTLAGP(3). The default values are:

LTLAGP(1) = 97

LTLAGP(2) = 77

LTLAGP(3) = 37

Elements of the LTLAGP array may be changed by inputting card type 141.

NSGAP(I) SIN092

Acceptable Gaps in near-side cross-street traffic for vehicles at a STOP Sign.

A vehicle at a stop-line facing a STOP sign cannot discharge until an acceptable gap is available in the cross street traffic. The decile distribution of acceptable gaps in the traffic stream on the near-side cross-street is stored in the NSGAP array, in units of tenths-of-a-second.

I =	1	2	3	4	5	6	7	8	9	10
Accept. Gaps =	56	50	46	42	39	37	34	30	26	20

To determine the acceptable gap in the far-side cross-street traffic see FSGAP array. The distribution contained in NSGAP may be altered by using card type 142.

PDLY(I) SIN097

Delay Due to Pedestrian Conflict

The program defines two "kinds" of conflicts: strong interaction and weak interaction. The duration of vehicular delay, in seconds, for each kind of conflict is defined by a statistical decile distribution stored in the PDLY array:

I =	1	2	3	4	5	6	7	8	9	10
Weak Inter =	0	0	0	0	0	0	0	1	2	6

I =	11	12	13	14	15	16	17	18	19	20
Strong Inter. =	0	0	0	1	2	3	4	5	8	15

It is noted that the strong interaction delay for heavy pedestrian flow is twice the table values and that I reflects a random number between 1 and 10, inclusive.

These values may be changed by using card type 146.

PDUR(I) SIN161

Duration of Parker Impedance

When a parker is due to block a lane, it is assigned a blockage duration by multiplying the specified mean duration for that link by a percentage extracted from a decile distribu-tion. A random number between 1 and 10, inclusive, is used as an index to enter this distribution which is stored in the PDUR array:

I =	1	2	3	4	5	6	7	8	9	10
	10	20	30	40	50	70	100	130	180	370

PHDWY(I) SIN162

Inter-arrival Headway between Parkers

When a parker is due to block a lane, the remaining time until the arrival of the next parker is assigned by multiplying the specified mean parker interarrival headway for that link by a percentage extracted from a decile distribution. A random number between 1 and 10 inclusive, is used as an index to enter this distribution which is stored in the PHDWY array:

I =	1	2	3	4	5	6	7	8	9	10

12 16 29 41 55 69 76 114 193 395

PPER(I)	SIN090	Duration of Strong Pedestrian Interactions The demarcation between light and strong interaction is expressed in terms of the elapsed time since beginning of the green phase that strong interaction prevails. For the remaining duration of the green phase, light interaction is in effect. The PPER array contains the duration, in seconds, of strong interaction for each of the three pedestrian intensities which are specified. Light Pedestrian Flow: PPER(1) = 0 Moderate Pedestrian Flow: PPER(2) = 10 Heavy Pedestrian Flow: PPER(3) = 25. These values may be changed by using card type 146.
SPLPCT(I)	SIN091	Probability of a Vehicle Joining (or Causing) Spillback A vehicle which faces a spillback condition on its receiving link at the time it is about to discharge into it, must "decide" whether to discharge or wait until the spillback ahead dissipates. This decision process affects only through vehicles. Left-turners always join a spillback (other conditions permitting), while right-turning vehicles are physically blocked by the spillback. The probability, in percent, of a vehicle joining a spillback comprised of I vehicles is defined in the SPLPCT array: I = 1 2 3 4 SPLPCT = 100 81 69 40 To alter these values use card type 141.
STEPCT(K)	SIN096	Duration of Short-Term Events The duration of a short-term event is assigned by multiplying the specified mean duration for that link, by a percentage extracted from a decile distribution. A random number between 1 and 10, inclusive, is used as an index to enter this distribution which is stored in the STEPCT array: K = 1 2 3 4 5 6 7 8 9 10 Percent = 10 20 30 40 50 70 100 130 180 370 To alter these values use card type 148.
TRNGAP(I)	SIN095	Acceptable Gaps for Turning Vehicles A decile distribution of acceptable gaps in the on-coming traffic facing left-turning vehicles is stored in the first 10 elements of the TRNGAP array. These values, in tenths-of-a-second, are: I = 1 2 3 4 5 6 7 8 9 10 Accept. Gaps = 78 66 60 54 48 45 42 39 36 27 (left-turners) A similar decile distribution provides acceptable lags in the traffic stream on the outside lane of the near-side cross-street is stored in the second 10 elements of the

		TRNGAP array; for right-turners completing a RTOR maneuver.

I	= 11 12 13 14 15 16 17 18 19 20
Accept. Gaps	= 104 88 80 72 64 60 56 52 48 36 (right-turners)

I represents the driver type of the vehicle. These values may be changed by using card type 145.

3.12 NETSIM Fuel and Emission Arrays

These arrays contain fuel consumption and vehicle emission rate tables for varying acceleration, speed and vehicle types. These arrays are exported. Each array is the vector representation of an equivalent three dimensional array of the form: Array(IACC, ISPD, ITYP) where:

IACC	Denotes 19 different values of vehicle acceleration in the range of (-9, 9) fpss
ISPD	Denotes 71 values of vehicle speed in the range of (0,70) fps
ITYP	Denotes 3 types of vehicles, (autos, trucks, buses)
	(only fuel array contains information for 3 types)

Hence, FLCN array totals 19 * 71 * 3 = 4047 elements, arrays COEM, HCEM, and NOEM total 19 * 71 = 1349.

The location I in these arrays, for a particular combination of acceleration, speed and vehicle type, is computed as follows:

I = 19 * (ITYP * 71 + ISPD) + IACC

COEM(I)	SIN126	Rate of carbon monoxide emissions in (grams/sec) * 1000
FLCN(I)	SIN124	Rate of fuel consumption in (gallons/ sec) * 100000
HCEM(I)	SIN125	Rate of hydrocarbon emissions in (grams/sec) * 1000
NOEM(I)	SIN127	Rate of oxides of nitrogen emissions in (grams/sec) * 1000

3.13 NETSIM Miscellaneous Data

This section describes all NETSIM variables that do not belong to any other grouping. These variables are exported.

Name	Common	Description
DECDIF(1,2)	SIN832	Array containing the difference in deceleration between the position when vehicle began to respond to object and the position of the object causing the lane change
DRCDAM(2)	SIN915	Driver familiarity codes (1 or 2 turns known)
JLKID(IL)	SIN900	Used for debugging purposes in subroutine ADJQ. Temporarily stores user node numbers.
JLKNM(IN)	SIN901	Used for debugging purposes in subroutine ADJQ. Temporarily stores link numbers.

TMREAC(I)	SIN846	Reaction time by driver type, I
SLCDIS(10)	SIN914	Distribution of longitudinal distance over which drives attempt to perform one lane change
VEHOCC(I)	SIN120	Average number of people occupying fleet component I, where Fleet

I	Component	Units	Default
1	Auto	Persons*100	130
2	Carpool	Persons*100	250
3	Truck	Persons*100	120
4	Bus	Persons*100	2500

These values may be altered through the use of either card type 52 or card type 58.

XDATA(I)	SIN122	Array containing all information input to and output from subroutine CRFLW, module 3232.2111.1. This routine executes the car-following logic:

I	Description
1	Front-to-rear separation distance to leader, feet
2	Lead vehicle speed at end of time step, fps.
3	Lead vehicle length, feet
4	Subject vehicle speed at start of time-step, fps.
5	Subject vehicle type
6	Subject vehicle free-flow speed
7	Calculated value of subject vehicle acceleration, fpss.
8	Calculated value of subject vehicle speed, fps.
9	Calculated value of distance traveled by subject vehicle during time-step, feet
10	Not used

XHEDWY(IHED)	SIN425	Array of headways for entry/source/sink emission/extracts.
XLSEED(I)	SIN178	Array of random number seeds for entry and entry-interface links and internal links with a source node. Pointer I is obtained from the PLSEED array.

For each entry link, elements I, I+1, I+2 and I+3 contain the random number seeds for left, through, right and diagonal movements, respectively. For each entry-interface link, elements I, I+1, I+2 and I+3 contain the random number seeds for each of the four vehicle types (i.e., autos, trucks, carpools, and buses) which will enter the NETSIM subnetwork via this interface node. For internal links with source nodes, only one random number seed is needed and is stored in the Ith location of this array.

These random number seeds are used as the base random number seed for generating the traffic stream in the NETSIM subnetwork.

XTMTRN(I)	SIN416	Array of turn movements to be used during sub-time periods of the current time period for specified links.

3.14 NETSIM Actuated Controller Data

This section contains all scalars and arrays used by NETSIM actuated controller logic. The NETSIM actuated database is presented by the following groupings:

- Scalars
- Detector Data
- Actuated Controller Data
- Pedestrian Phase Data
- Interface Data
- Link Data
- Phase Data
- Miscellaneous Arrays

3.14.1 Scalars

These scalars are exported.

Name	Common	Description
ACN	SIN302	Total number of actuated controllers in the subnetwork.
EFREQ	SIN167	Evaluation frequency for surveillance detectors.
MXACLK	SIN300	Maximum number of links serviced or referenced by an actuated controller.
MXDET	SIN315	Maximum number of detectors.
MXDETL	SIN316	Maximum number of detectors per link.
MXNACT	SIN301	Maximum number of actuated controllers.
MXPCD	SIN326	Maximum number of constant pedestrian demand period.
MXPDFZ	SIN324	Maximum number of pedestrian phases.
MXMTRN	SIN429	Maximum number of entries in XTMTRN array.
MXMVOL	SIN435	Maximum number of entries in XTMVOL array.
NPEDCD	SIN325	Total number of pedestrian constant demand periods.
NPEDFZ	SIN323	Total number of pedestrian phases.
NUMDET	SIN306	Total number of detectors.
STBPOS	SIN959	Location of stopbar from downstream node
TOTZHD	SIN428	Total number of entries in XHEDWY array.
UDTSUR	SIN142	Elapsed time since last surveillance detector intermediate output.

The current version allows for specification of 100 dual-ring controllers. Each controller may service or reference up to 10 links. A total of 700 detectors may be specified on the NETSIM subnetwork, serving the

			actuated controllers. The limit on pedestrian phases and pedestrian constant demand period is set to 100.
WSDET	SIN027		Flag (T,F) if there (are, are not) surveillance detectors in the network.
WSDUMP	SIN110		Flag (T,F) if surveillance data (should, should not) be dumped.

3.14.2 Detector Data

The detector related arrays are one dimensional with the subscript DT (detector identification number). These arrays are exported.

Name	Common		Description
CUMSPD(DT)	SIN145		Cumulative vehicle speeds.
DAPRCH(DT)	SIN335		Approach number of link in which the detector is positioned.
DETOCC(DT)	SIN369		Cumulative activation time of detector DT.
DETON(DT)	SIN070		Bit specific array:
		1	(0,1) if detector was (on,off) for first 0.1 second
		2	(0,1) if detector was (on,off) for second 0.1 second
		3	(0,1) if detector was (on,off) for third 0.1 second
		4	(0,1) if detector was (on,off) for forth 0.1 second
		5	(0,1) if detector was (on,off) for fifth 0.1 second
		6	(0,1) if detector was (on,off) for sixth 0.1 second
		7	(0,1) if detector was (on,off) for seventh 0.1 second
		8	(0,1) if detector was (on,off) for eighth 0.1 second
		9	(0,1) if detector was (on,off) for ninth 0.1 second
		10	(0,1) if detector was (on,off) for tenth 0.1 second
DTLANE(K)	SIN311		Lane number in which the detector DT is positioned. Two entries are allocated for each detector. The first entry (K = (DT – 1) * 2 + 1) contains the lane identification of the first lane, the second entry (K = (DT – 1) * 2 + 2) contains the lane identification of a second lane served by the same detector. A value of 9 for any of the two entries specifies that the detector is across all lanes of the link. If a turning pocket lane is specified, the detector is assumed to be across all lanes of the turning pocket.
DTLEN(DT)	SIN313		Detector length, in tenths-of-a-foot.
DTMOD(DT)	SIN314	1-3	Detector type (0=Presence, 1=Passage)
		4-10	Speed of vehicle when passing the passage detector.
		11-32	Vehicle count since beginning of simulation.
DETID(DT)	SIN719		Detector number (surveillance detectors only)
DTLNK(DT)	SIN700		Link Number surveillance detector is on.
DTNLNK(DT)	SIN308		Detector identification number of the next detector on the same link.

DTNPRT(DT)	SIN310	Detector identification number of the next detector serving the same controller, same phase and connected to the same input port.
DTPOS(DT)	SIN312	Distance between the detector's downstream edge and the downstream stop-bar, in tenths-of-a-foot.
EMVDET(DT)	SIN245	ID of emergency vehicle.
IDTIME(DT)	SIN341	Elapsed timed since beginning of the current time step, in tenths-of-a-second, when the first actuation was recorded.
LASTD(DT)	SIN342	Elapsed timed since beginning of the current time step, in tenths-of-a-second, when the last actuation was recorded.
LVACT(DT)	SIN367	Last vehicle ID that activated the detector.
OLDCNT(DT)	SIN165	Cumulative vehicle count at beginning of intermediate output period.
OLDON(DT)	SIN151	Cumulative on-time at beginning of intermediate output period.
OLDSPD(DT)	SIN148	Cumulative speeds at beginning of intermediate output period.
WDET(DT)	SIN340	Flag (.T., .F.) if an actuation (is, is not) recorded in the current time step.
WDETPR(DT)	SIN344	Flag (.T., .F.) if an actuation (was, was not) recorded in the previous time step.
WDTYPE(DT)	SIN368	Flag (T,F) if this detector (is, is not) a surveillance detector.
WSMALF(DT)	SIN370	Flag (T,F) if this detector (is, is not) malfunctioning.
ZNDOCC(DT)	NDET02	Average occupancy for the last DPPINT number of seconds.
ZNDSPD(DT)	NDET01	Average speed for the last DPPINT number of seconds.
ZNDVOL(DT)	NDET03	Average volume for the last DPPINT number of seconds.

3.14.3 Actuated Controller Data

The actuated controller-specific arrays are one dimensional with the subscript IAC (actuated controller identification number). Only the arrays used to store data for subsequent time periods are exported.

Name	**Common**	**Description**
ACADDF(IAC)		Bit I (I = 1, 2, ..., 8) is on if phase I (I = 1, 2, ..., 8) has an added initial interval.
ACADDFS(IAC)		This array is the same as ACADDF but stores data for subsequent time periods and is used in the transition logic
ACCORT(IAC)		Coordination timer. It is set to cycle length and decremented each time step. Expiration of this timer indicates the time for execution of coordination logic.
ACCYCL(IAC)		Cycle length, in seconds.

ACCYCLS(IAC)		This array is the same as ACCYCL but stores data for subsequent time periods and is used in the transition logic
ACGPF(IAC)		Bit I is on if phase I will be extended for that vehicle passage. (I = 1, 2, ..., 8)
ACGPFS(IAC)		This array is the same as ACGPF but stores data for subsequent time periods and is used in the transition logic
ACMXRF(IAC)		Bit I is on if phase I is on maximum recall. (I = 1, 2, ..., 8)
ACMXRFS(IAC)		This array is the same as ACMXRF but stores data for subsequent time periods and is used in the transition logic
ACOFST(IAC)		Offset time, in seconds.
ACOFSTS(IAC)		This array is the same as ACOFST but stores data for subsequent time periods and is used in the transition logic
CONTME(IPHASE,IAC)		This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is the minimum amount of time required to provide the conditional service when a call for the phase (P) is issued. IP = (IAC – 1) * 8 + P
CONTMES(IPHASE,IAC)		This array is the same as CONTME but stores data for subsequent time periods and is used in the transition logic
INTRAN(IAC)		Transition Flag for reference controller (>0 If in transition)
WNEWPLN(IAC)		New plan Flag for reference controller. (Set to TRUE if controller inputs have been updated)
WFIRST(IAC)		Flag (.T., .F.) if this (is, is not) the first execution of Q5 actuated logic for the controller.
WTODOP(IAC)		The time-of-day flag of the referenced controller.
XPHFLG(IAC)	1-8	Phase specific bit flags. Each bit, I, contains a code (1,0) if phase, I (cannot, can) terminate before its defined force-off point is reached. If the controller is not coordinated, these bits are set to zero
	9-16	Each bit, I, contains a code (1,0) if phase I-8 (does, does not) require both phases in a barrier crossing situation to simultaneously gapout or have both maximum timers expire in order to end the phase
	17-24	Each bit, I, contains a code (1,0) if phase I-16 (may, may not) be serviced twice during a cycle. This bit must be zero for Lag phases on non-coordinated controllers and for all phases on coordinated controllers.
XPHFLGS(IAC)		This array is the same as XPHFLG but stores data for subsequent time periods and is used in the transition logic

XTEND(IAC)	1-4	Number of a leading side street left turn phase that can be extended beyond its normal force-off time when no pedestrian demand exists
	5-12	Number of seconds that force-off can be extended for phase in bits 1-4
	13-16	Number of another leading side street left turn phase that can be extended beyond its normal force-off time when no pedestrian demand exists.
	17-24	Number of seconds that force-off can be extended for phase in bits 13-16
XTENDS(IAC)		This array is the same as XTEND but stores data for subsequent time periods and is used in the transition logic

3.14.4 Pedestrian Phase Data

The pedestrian-specific variables are one-dimensional arrays with the subscript IPEDFZ, (pedestrian phase identification number). IPEDFZ = ACPDFZ((IAC - 1) * 8 + IPHASE) where IAC is the actuated node identification number and IPHASE is the signal phase. These arrays are exported.

Name	Common	Description
PDARVL(IPEDFZ)	SIN328	Pedestrian inter-arrival headway for deterministic arrival and pedestrian arrival rate for stochastic arrivals.
PDFCD(IPEDFZ)	SIN331	Index to data position of the first pedestrian continuous demand period.
PDNART(IPEDFZ)	SIN330	Clock time of next pedestrian arrivals, in seconds.
PDTYPE(IPEDFZ)	SIN329	Pedestrian arrival type code.

Code	Arrival Type
1	Stochastic
2	Deterministic
3	Continuous demand
4	Stochastic + Continuous demand
5	Deterministic + Continuous demand

The following arrays pertain to pedestrian phase data defining periods of continuous demand. Five periods of constant demand may be specified for each pedestrian phase. These arrays are one dimensional with the subscript, IPC (constant demand period identification number). The total number of defined constant demand periods for a network is NPEDCD. IPC is the identification number of the first constant demand period associated with pedestrian phase, IPEDFZ, where IPC = PDFCD(IPEDFZ).

Name	Common	Description
PDBGNT(IPC)	SIN333	Beginning time of the continuous demand period, in seconds.
PDENDT(IPC)	SIN334	Ending time of the continuous demand period.
PDNCD(IPC)	SIN332	Index, IPC to data position of the next pedestrian continuous demand period.

3.14.5 Interface Data

The following arrays and/or scalars provide the interface between the NETSIM and the Q5 actuated control logic. For each controller the required data is loaded into these arrays and transferred to the Q5 logic. Arrays declared in a named common are exported.

Name	Common	Description
ADDOPT	SIN362	Added initial option flag. Set to 1 if any phase has added initial interval.
BEGPM(3)	SIN360	Each element corresponds to one permissive period. The content of each element is the beginning time of the corresponding permissive period, in seconds.
DPPOUT(IPORT,IPHASE)		This array is a two dimension array where the first index is the port and the second index is the phase. The content is 1 if a detector actuation was recorded for the referenced phase and the referenced actuated controller.
DTEXST(IPORT,IPHASE)		This array is a two dimension array where the first index is the port and the second index is the phase. The content is 1 if there is at least one detector specified to serve the referenced node via the referenced input port.
ENDPM(3)	SIN361	Each element corresponds to one permissive period. The content of each element is the ending time of the corresponding permissive period, in seconds.
FORCEI(8)	SIN363	Each element corresponds to a phase. The content of each element is the force-off time for the corresponding phase.
GPTFLG	SIN356	Bit I (I = 1, 2, ..., 8) is on if phase I (I = 1, 2, ..., 8) is extended to guarantee the last vehicle passage.
IOVECT(8)	SIN346	Each element corresponds to a phase. The content of each element consist of a 6-bit representation corresponding to the 6 input ports of the 170 controller. Bit I (I = 1, 2, ..., 6) is on if an actuation is recorded for input port I (I = 1, 2, ..., 6) for the corresponding phase.
LOWRAM(I)	SIN355	This array is currently dimensioned at 751. It is an emulation of the MOTOROLA 6800 RAM memory. See Section 5 for a description of the array elements.
MXDEPZ(K)	SIN343	The subscript K of the array represents the phase number. Each element consists of the maximum actuation time (i.e., the time when last actuation was recorded) for the referenced phase.
OFFSET	SIN358	Offset time, in seconds.
PRMFZ(3)	SIN359	Permissive phase function flag. The first element is the phase function flag for permissive period one, the second element is the phase function flag for permissive period two, the third element is the phase function for third permissive period. Each flag consists of eight bits. Bit I (I = 1, 2, ..., 8) is on if phase I (I = 1, 2, ..., 8) is assigned to the corresponding period.
VXCALL	SIN357	Bit I (I = 1, 2, ..., 8) is on if phase I (I = 1, 2, ..., 8) is on maximum recall.

3.14.6 Phase Data

The actuated phase-specific arrays are two dimensional arrays. Only the arrays used to store data for subsequent time periods and the following arrays are exported: ACPDFZ and GRNMAX.

Name	Description
ACFORC(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is force-off time, in seconds, for the referenced phase of the referenced controller.
ACFORCS(IPHASE,IAC)	This array is the same as ACFORC but stores data for subsequent time periods and is used in the transition logic
ACFPRM(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is permissive period flags. Each element is packed z * 100 + y * 10 + x where x is code (0, 1) if sync phase (cannot, can) yield to the phase during first permissive period. Similarly, y and z are for 2nd and 3rd permissive periods, respectively.
ACFPRMS(IPHASE,IAC)	This array is the same as ACFPRM but stores data for subsequent time periods and is used in the transition logic
ACIGR(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is the gap reduction type code.

Code	Gap reduction code
0	Reduce by/Reduce every
1	Reduce by every second
2	Time to reduce to minimum gap

Name	Description
ACIGRS(IPHASE,IAC)	This array is the same as ACIGR but stores data for subsequent time periods and is used in the transition logic
ACIIC(IPHASE,IAC))	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is the initial interval type code for the referenced phase of the referenced controller.

Code	Initial interval type
0	Extensible
1	Added
2	Computed

Name	Description
ACIICS(IPHASE,IAC)	This array is the same as ACIIC but stores data for subsequent time periods and is used in the transition logic
ACMXIN(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is the number of actuations to reach maximum initial interval for the referenced phase of the referenced controller.

ACMXINS(IPHASE,IAC)	This array is the same as ACMXIN but stores data for subsequent time periods and is used in the transition logic
ACPDFZ(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is associated pedestrian phase number for the actuated node and the referenced phase number.
ACTRMC(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is the termination code for the controller.
ACTSEQ(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. This array stores the phase sequence for the controller.
ACTSEQS(IPHASE,IAC)	This array is the same as ACTSEQ but stores data for subsequent time periods and is used in the transition logic
ALPTRM(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is the number of the phase that is terminating.
ATRMIN(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is the time to reduce to minimum gap for the referenced phase of the referenced controller (seconds).
ATRMINS(IPHASE,IAC)	This array is the same as ATRMIN but stores data for subsequent time periods and is used in the transition logic
GRNMAX(IPHASE,IAC)	This a two dimensional array where the first index is the phase and the second index is the referenced actuated node. The stored value is the maximum green time for the referenced phase, in seconds.
ACAUSE(IRING,IAC)	This a two dimensional array where the first index is the ring and the second index is the referenced actuated node. The stored value is the cause for activation of the active phase for the ring.
APHASE(IRING,IAC)	This a two dimensional array where the first index is the ring and the second index is the referenced actuated node. The stored value is the current active phase for the ring.
ATIME(IRING,IAC)	This a two dimensional array where the first index is the ring and the second index is the referenced actuated node. The stored value is the time the current phase has been active for the ring.

3.14.7 Miscellaneous Arrays

Only the arrays used to store data for subsequent time periods and the following arrays are exported: ACNLNK, ACPADD, ACPSUB, ACTRNM, and DTFPRT.

Name	Description
ACBPRM(IPERM,IAC)	This array is a two dimensional array where the first index is permissive period and the second index is the actuated reference node. The content of each element is the beginning time of the referenced permissive period for the referenced controller.
ACBPRMS(IPERM,IAC)	This array is the same as ACBPRM but stores data for subsequent time periods and is used in the transition logic
ACEPRM(IPERM,IAC)	This array is a two dimensional array where the first index is permissive period and the second index is the actuated reference node. The content of each element is the ending time of the referenced permissive period for the referenced controller.
ACEPRMS(IPERM,IAC)	This array is the same as ACEPRM but stores data for subsequent time periods and is used in the transition logic
ACNLNK(I,IAC)	This array is a two dimensional array where the first index is the link sequence number and the second index is the actuated reference node. The stored value is the link identification number of the referenced approach link. Link sequence number has a value between 1 and IMACLK.
ACPADD(IAC)	The content of each element is the maximum amount of adjustment per cycle for the Dwell or Add methods, expressed as a percent of the cycle length. This limit is also used by the Short Way method when it is operating in the add mode.
ACPSUB(IAC)	The content of each element is the maximum amount of adjustment per cycle for the Subtract method, expressed as a percent of the cycle length. This limit is also used by the Short Way method when it is operating in the subtract mode.
ACTIMT(IAC)	The content of each element is the inhibit max termination flag for the referenced controller.
ACTIMTS(IAC)	This array is the same as ACTIMT but stores data for subsequent time periods and is used in the transition
ACTRAM(K,IAC)	The array is an emulation of the 751 element MOTOROLA 6800 RAM memory for the referenced actuated node. The array is a two dimensional array where the first index is the index into the Q5 memory and the second is the referenced actuated node.
ACTRAMS(K,IAC)	This array is the same as ACTRAM but stores data for subsequent time periods and is used in the transition logic

ACTRNM(IAC)	The content of each element is the transition method for the referenced controller.
AMOVSP(IAP,IPHASE,IAC))	This array is a three dimensional array where the first index is the approach number, the second index is the phase and the third index is the actuated node identification number. Each element is a packed word of movement - specific codes (0,1) if movement (is not, is) allowed on given approach during given phase. Each word is packed as follows: 1 Code for right-diagonal turn movement. 2 Code for left-diagonal turn movement. 3 Code for right turn movement. 4 Code for through movement. 5 Code for left turn movement.
AMOVSPS(IAP,IPHASE,IAC)	This array is the same as AMOVSP but stores data for subsequent time periods and is used in the transition logic
DTFPRT(IPORT,IPHASE,IAC)	This array is a three dimensional array where the first index is the port, the second index is the phase and the third index is the referenced actuated node. The content of each element is the detector identification number of the first detector serving the referenced phase number and the input number for the referenced actuated node number.
PHWGN(IPHASE)	This a single dimensional array indexed by phase. The stored value is the dwell green point for the phase.
PHTRMC(IRING)	This a single dimensional array indexed by ring. The stored value is the number of the last phase that is terminating.

3.15 NETSIM Interchange Data

These arrays are dynamically allocated and defined in the NETSIM_INTERCHANGES module. They are exported and can be externally referenced as NETSIM_INTERCHANGES _mp_XXX, where XXX represents the listed name of the array.

Name	Common	Description
ICHMAP(ICHG)		
NTCHG(ICHG)		Pointer to first element in XTRPTB pertaining to interchange, ICHG
NUM96C(ICHG)		Number of elements in XTRPTB array for interchange, ICHG
NUMLNK(ICHG)		Number of links in interchange, ICHG
XIPATH(IC, IP, IS)		Turn movement code +1 for IS turn movement through interchange IC on path IP
XNCGLK (ICHG, N)		Link number of a link within interchange, ICHG
XTRPTB_ORIG(K)		Link number of an origin link within interchange, ICHG

XTRPTB_DEST(K)	Link number of a destination link within interchange, ICHG
XTRPTB_TCODE(K)	Turn code on destination link
XTRPTB_PERCNT(K)	Percentage of traffic on origin link that travels to destination link and executes turn movement in bits 21-23

3.16 NETSIM Micro-Node Data

These arrays are exported.

Name	Common	Description
ARBAPR(IMR)	SIN860	Link number
ARBDIS(IMR)	SIN861	The smaller of the distance from the stop-line to point at upstream end of link where curb or median ends
ARBFED(IMR)	SIN862	Feeder link
ARBPAR(IMR)	SIN863	Parallel link number
		IMR is defined to be:
	1-5	index for up to 5 possible approaches to a micro-node
	6-10	index for up to 5 possible departing links from a micro-node
	11-55	index for up to 45 possible secondary links associated with a micro-node, i.e., for each departing link from a micro-node there are up to 4 other approaches and 5 departing links for the downstream node of that receiver (It is important to store information about these links when determining geometry of the intersection)
COLCNT(K)	SIN888	Count of collisions by movements
		K is defined to be (IMIC - 1) * 40 + (IAP - 1) * 8 + (IMV - 1) * 2 + IMV2, where IMIC is the micro-node index, IAP is the approach number in SIGI array, IMV is the primary movement (1=L, 2=T, 3=R, 4=D) and IMV2 is the conflicting movement (1=T/D, 2=L/R).
CUMICM(K)	SIN890	Cumulative vehicle trips through intersection by movement
		K is defined to be (IMIC - 1) * 20 + (IAP - 1) * 4 + IMV, where IMIC is the micro-node index, IAP is the approach number in SIGI array, and IMV is the primary movement (1=L, 2=T, 3=R, 4=D)
CUMMIC(IMIC)	SIN889	Number of vehicles discharging micro-intersection from beginning of simulation
ZRBANG (IMR)	SIN872	Angle link makes with x-axis
ZRBCNX (IMR)	SIN873	X coordinate of central point, 0 if straight link
ZRBCNY (IMR)	SIN874	Y coordinate of central point, 0 if straight link
ZRBNOD(IMR)	SIN875	Distance from stop-line of link to downstream node
		IMR = 1 to 5
ZRBRCC(IMR)	SIN876	Radius from center to curb on downstream end, 0 if straight link
ZRBRCN(IMR)	SIN877	Radius from center to upstream/downstream node, 0 if straight link

ZRBSLX(IMR)	SIN878	The x coordinate of the point where the stop-line intersects the curb, 0 if straight link
ZRBSLY(IMR)	SIN879	The y coordinate of the point where the stop-line intersects the curb, 0 if straight link
ZRBSTP(IMR)	SIN880	Distance from stop-line of link to the upstream node IMR is defined to be: 1-5 index for up to 5 possible approaches to a micro-node 6-10 index for up to 5 possible departing links from a micro-node 11-55 index for up to 45 possible secondary links associated with a micro-node, i.e., for each departing link from a micro-node there are up to 4 other approaches and 5 departing links for the downstream node of that receiver. (It is important to store information about these links when determining geometry of the intersection).
XMDIST(K)	SIN894	Array containing distance traveled (vehicle feet) through micro - intersection, stratified by vehicle type. This array is the vector (i.e., one-dimensional) representation of the equivalent array, XMDIST (IMIC, I) where, IMIC = Micro-node: 1, 2, ..., MAXMIC I = Vehicle type: 1, 2, 3 (auto, truck, bus) K is defined to be MAXMIC * (I - 1) + IMIC, where MAXMIC is the maximum number of micro-nodes.
XREBP(K)	SIN869	1-15 Pointer to word of XLKREB array for the reb associates with path index K. 16-19 Pointer to field of XLKREB array for the reb associates with path index K. 20 Code (0,1) if reb as associated with path index K is last reb in path. 21-29 Distance from upstream stop-line to tail of reb as associated with path index K. K is defined to be the index of sequential rebs on the paths through the micro-intersection.
YBLKNO(IMIC)	SIN895	Flag (T, F) if any vehicle within intersection (is, is not) blocked by an intersection blockage.
ZRBACM(IMR)	SIN870	The angle the link from the center of the basic arc to the (curbside, median side) upstream point, if (rdisc, rdism) is the minimum distance from the stop-line with respect to the center line, 0 if straight link.
ZRBACS(IMR)	SIN871	The angle the radial to curbside stop-line point makes with x-axis, 0 if straight line.
XLKREB(K+1)	SIN868	1-3 Status of reb 1. 4-6 Status of reb 2. 21-23 Status of reb 8.
XLKREB(K+2)		1-15 Vehicle ID influencing reb 1.

		16-31	Vehicle ID influencing reb 2.
XLKREB(K+3)		1-15	Vehicle ID influencing reb 3.
		16-31	Vehicle ID influencing reb 4.
XLKREB(K+4)		1-15	Vehicle ID influencing reb 5.
		16-31	Vehicle ID influencing reb 6.
XLKREB(K+5)		1-15	Vehicle ID influencing reb 7.
		16-31	Vehicle ID influencing reb 8.
			K is defined to be (IMIC - 1) * 175 + (IAP - 1) * 35 + (ILN - 1) * 5, where IMIC is the index to the micro-node, IAP is the approach number in SIGI array, and ILN is the lane number.
XMBUF(K)	SIN893		Array of data describing resources consumed through each micro - intersection, stratified by vehicle type. The array is the vector representation of the equivalent array, XMBUF (IMIC, J, I) where,
			IMIC = Micro-node: 1, 2, ..., MAXMIC
			J = Resource code: 1, 2, 3, 4 (Fuel, HC, CO, NOX)
			I = Vehicle type: 1, 2, 3 (auto, truck, bus)
			K is defined to be MAXMIC * 4 * (I - 1) + MAXMIC * (J - 1) + IMIC, where MAXMIC is the maximum number of micro-nodes.
PTHREB(K+1)	SIN866	1-15	Pointer to XREBP for first reb in path for given micro-node approach, lane with left movement.
		16-30	Pointer to XREBP for first reb in path for given micro-node, approach, lane with through movement.
PTHREB(K+2)		1-15	Pointer to XREBP for first reb in path for given micro-node, approach, lane with right movement.
		16-30	Pointer to XREBP for first reb in path for given micro-node, approach, lane with diagonal movement.
			K is defined to be (IMIC - 1) * 70 + (IAP - 1) * 14 + (ILN -1) * 2, where IMIC is the index to the micro-node, IAP is the approach number in SIGI array, and ILN is the lane number.
VMICRO(IVIN)	SIN886		Vehicle number
			IVIN is defined to be 1, 2, ... MAXVIN
VREBP(K)	SIN887	1-3	Status of reb
		4	Code indicating if reb is the last in path
		5-15	Vehicle ID of blocking vehicle.
			K is defined to be 1, 2, ... MAXTRJ
XCOREB(K+1)	SIN867	1-15	X coordinate of first corner of reb
		16-31	Y coordinate of first corner of reb

XCOREB(K+2)		1-15	X coordinate of second corner of reb
		16-31	Y coordinate of second corner of reb
XCOREB(K+3)		1-15	X coordinate of third corner of reb
		16-31	Y coordinate of third corner of reb
XCOREB(K+4)		1-15	X coordinate of fourth corner of reb
		16-31	Y coordinate of fourth corner of reb
			K is defined to be (IAP - 1) * 224 + (ILN - 1) * 32 + (IREB - 1) * 4, where IAP is the approach number in SIGI array, ILN is the lane number and IREB is the reb number.
DATMIC(ITRAJ)	SIN891	1-7	Vehicle speed, feet/sec. Value is less than or equal to 128
		8-10	Vehicle type code: **Code** **Vehicle Type** 0 Auto 1 Truck 2 Bus
		11-14	Vehicle acceleration/deceleration in fpss (Less than or equal to 12) 15 Set to (0, 1) if the contents of bits 11-14 indicate (acceleration, deceleration)
		15	Set to (0,1) if the contents of bits 11-14 indicate (acceleration, deceleration)
			DATMIC is a micro-node trajectory word
			ITRAJ is defined to be DATBUF(1) and is less than or equal to 250.
DEPLNK(K)	SIN884		Link Number
			K is defined to be (IMIC - 1) * 5 + I, where IMIC is the micro-node index and I is from 1 to 5 for the 5 possible links departing the micro-node
MCNOD(IMIC)	SIN883		Node number corresponding to micro-node, IMIC
PEDLY(K)	SIN885		Remaining pedestrian delay (10ths of secs) on last reb of this departing link
			K is defined to be (IMIC -1) * 5 + I, where IMIC is the micro-node index and I is from 1 to 5 for the 5 possible links departing the micro-node

4 Parameters

The following module is used by CORSIM to store parameters that are used to dimension arrays that are defined in other modules.

```
MODULE DATA_MOD
INTEGER,PARAMETER::MAXENTRYNODES = 2000
INTEGER,PARAMETER::MAXTIMEPOINTS = 304
INTEGER,PARAMETER::MAXNODESINPATH = 100
INTEGER,PARAMETER::MAXPATHS = 5000
INTEGER,PARAMETER::MAXPROBES = 200
END MODULE DATA_MOD
```

Name	Description
MAXLINKS	Number of links in the turn data.
MAXENTRYNODES	Number of nodes in the entry volume data.
MAXTIMEPOINTS	Number of data points in the time and entry volume data.
MAXNODESINPATH	Number of nodes in a path.
MAXPATHS	Number of paths that can be stored internally.
MAXPROBES	Number of vehicle probes allowed.

4.1 Time-Varying Data

The following module is used by CORSIM to store time-varying turn percentages for NETSIM links. Variables that are exported (DLLEXPORT attribute) can be referenced as NETSIM_TURNDATA_MOD_mp_XXX, where XXX represents the listed name of the variable.

```
MODULE NETSIM_TURNDATA_MOD
USE DATA_MOD
TYPE NTURNDATA
  SEQUENCE
  INTEGER::UP
  INTEGER::DOWN
```

```
  INTEGER,DIMENSION(MAXTIMEPOINTS)::TIME
  INTEGER,DIMENSION(4,MAXTIMEPOINTS)::IPCT
  INTEGER::INDEX
  INTEGER::CARDTYPE
  INTEGER::LINKID
  LOGICAL::TIME_VARYING
END TYPE
TYPE(NTURNDATA),ALLOCATABLE,DIMENSION(:)::NTURN
INTEGER::NTURNS = 0
!DEC$ATTRIBUTES DLLEXPORT::NTURNS
!DEC$ATTRIBUTES DLLEXPORT::NTURN
END MODULE NETSIM_TURNDATA_MOD
```

Name	Description	
NTURNDATA	User defined type storing NETSIM turn percentage data.	
	Each entry in NTURNDATA contains the following:	
	UP	User-specified upstream node of the link.
	DOWN	User-specified downstream node of the link.
	TIME	Array of time points entered on Record Type 23 or beginning and ending times for a time period when turn percentages are defined by Record Type 21.
	IPCT	Array of four turn percentages for each time point.
	INDEX	Number of time points entered for the link.
	CARDTYPE	Identifier of Record Type that was used to enter the data.
	LINKID	The internal link ID for the link.
	TIME_VARYING	Logical flag indicating time-varying data was entered for the link.
NTURN	Array of NTURNDATA records for the entire NETSIM network. There is one NTURNDATA record for each link in the network.	
NTURNS	Number of elements in the NTURN array.	

The following module is used by CORSIM to store time-varying turn percentages for FRESIM links. Variables that are exported (DLLEXPORT attribute) can be referenced as FRESIM_TURNDATA_MOD_mp_XXX, where XXX represents the listed name of the variable.

```
MODULE FRESIM_TURNDATA_MOD
USE DATA_MOD
TYPE FTURNDATA
  SEQUENCE
```

```
  INTEGER::UP
  INTEGER::DOWN
  INTEGER,DIMENSION(MAXTIMEPOINTS)::TIME
  REAL,DIMENSION(2,MAXTIMEPOINTS)::RPCT
  INTEGER::INDEX
  INTEGER::CARDTYPE
  INTEGER::LINKID
  LOGICAL::TIME_VARYING
END TYPE
TYPE(FTURNDATA),ALLOCATABLE,DIMENSION(:)::FTURN
INTEGER::FTURNS = 0
!DEC$ATTRIBUTES DLLEXPORT::FTURNS
!DEC$ATTRIBUTES DLLEXPORT::FTURN
END MODULE FRESIM_TURNDATA_MOD
```

Name	Description	
FTURNDATA	User defined type storing FRESIM turn percentage data.	
	Each entry in FTURNDATA contains the following:	
	UP	User-specified upstream node of the link.
	DOWN	User-specified downstream node of the link.
	TIME	Array of time points entered on Record Type 26 or beginning and ending times for a time period when turn percentages are defined by Record Type 25.
	RPCT	Array of two turn percentages for each time point.
	INDEX	Number of time points entered for the link.
	CARDTYPE	Identifier of Record Type that was used to enter the data.
	LINKID	The internal link ID for the link.
	TIME_VARYING	Logical flag indicating time-varying data was entered for the link.
FTURN	Array of FTURNDATA records for the entire FRESIM network. There is one FTURNDATA record for each link in the network.	
FTURNS	Number of elements in the FTURN array.	

The following module is used by CORSIM to store time-varying entry volumes. Variables that are exported (DLLEXPORT attribute) can be referenced as FLOWDATA_MOD_mp_XXX, where XXX represents the listed name of the variable.

```
MODULE FLOWDATA_MOD
USE DATA_MOD
```

```
TYPE FLOWDATA
  SEQUENCE
  INTEGER::UP
  INTEGER::DOWN
  INTEGER::SSNODE
  INTEGER,DIMENSION(MAXTIMEPOINTS)::TIME
  INTEGER,DIMENSION(MAXTIMEPOINTS)::FLOW
  INTEGER::INDEX
  INTEGER::CARDTYPE
  INTEGER::LINKID
  INTEGER::NETWORK
  INTEGER::LANES
  LOGICAL::TIME_VARYING
  LOGICAL:: INTERPOLATE
END TYPE
TYPE(FLOWDATA),DIMENSION(MAXENTRYNODES)::ENTRYLINK
REAL,DIMENSION(MAXENTRYNODES)::RSPEC
INTEGER,DIMENSION(MAXENTRYNODES)::NEMITD
INTEGER::ENTRIES = 0
!DEC$ATTRIBUTES DLLEXPORT::ENTRIES
!DEC$ATTRIBUTES DLLEXPORT::ENTRYLINK
END MODULE FLOWDATA_MOD
```

Name	**Description**	
FLOWDATA	User defined type storing CORSIM entry volume data.	
	Each entry in FLOWDATA contains the following:	
	UP	User-specified upstream node of the link.
	DOWN	User-specified downstream node of the link.
	SSNODE	User-specified ID of the source/sink node servicing the link.
	TIME	Array of time points entered on Record Type 53 or beginning and ending times for a time period when turn percentages are defined by Record Type 50.
	FLOW	Entry volume for each time point.
	INDEX	Number of time points entered for the link.
	CARDTYPE	Identifier of Record Type that was used to enter the data.
	LINKID	The internal link ID for the link.
	NETWORK	Network ID; 3 for NETSIM, 8 for FRESIM

	LANES	Number of lanes on the entry link.
	TIME_VARYING	Logical flag indicating time-varying data was entered for the link.
	INTERPOLATE	Logical flag indicating that entry volumes should be interpolated. Entry volumes specified as counts are not interpolated.
ENTRYLINK	Array of FLOWDATA records for the entire CORSIM network.	
ENTRIES	Number of elements in the ENTRYLINK array.	
NEMITD	Actual number of vehicles that have been emitted from the entry link, from the beginning of the simulation until the current time.	
RSPEC	Number of vehicles that should have been emitted from the entry link, from the beginning of the simulation until the current time.	

4.2 Vehicle Generation Data

The following module is used by CORSIM to store data representing vehicles that are being generated.

```
MODULE VEHICLE_MOD
USE DATA_MOD
TYPE VEHICLE
  SEQUENCE
  INTEGER::ENTRYLINKID
  REAL::DEPARTURE
  INTEGER::NETWORK
  INTEGER::PATHID
  INTEGER::DRIVER
  INTEGER::FLEET
  INTEGER::VTYPE
  INTEGER::GLOBALID
  INTEGER::TURNCODE
  INTEGER::VMSCOMPLIANCE
  LOGICAL::PROBEFLAG
  INTEGER::NETFLOW
END TYPE VEHICLE
INTEGER::NPROBES = 0
INTEGER::PROBE(MAXPROBES)
END MODULE VEHICLE_MOD
```

Name	Description
VEHICLE	User defined type storing CORSIM entry volume data.

	Each entry in VEHICLE contains the following:	
	ENTRYLINKID	The internal link ID for the link where the vehicle will enter the network.
	DEPARTURE	Time when the vehicle will enter the network.
	NETWORK	Network ID; 3 for NETSIM, 8 for FRESIM
	PATHID	ID of the path the vehicle will follow; 0 if the vehicle is not following a path
	DRIVER	Driver type
	FLEET	Auto, Truck, Bus or Carpool
	VTYPE	Vehicle type.
	GLOBALID	Vehicle's Global ID.
	TURNCODE	Turn code representing the movement the vehicle will perform at the end of the link if it is entering the network on a NETSIM link.
	VMSCOMPLIANCE	Compliance factor for VMS instructions.
	PROBEFLAG	Logical flag indicating vehicle is a probe vehicle.
	NETFLOW	Indicates that the traffic flow at a source/sink node is positive or negative.
NPROBES	Number of vehicles that are flagged as probe vehicles.	
PROBE	Array of global vehicle IDs for probe vehicles.	

The following module is used by CORSIM to construct a queue of vehicles that are waiting to be emitted into the network.

```
MODULE QUEUE_MOD
USE DATA_MOD
USE VEHICLE_MOD
TYPE QNODE
  SEQUENCE
  TYPE(VEHICLE)::VEHICLE
  TYPE(QNODE),POINTER::NEXT
END TYPE QNODE
TYPE QPOINTER
  SEQUENCE
  TYPE(QNODE),POINTER::PTR
END TYPE QPOINTER
TYPE(QPOINTER),DIMENSION(4,MAXENTRYNODES)::FRONT
TYPE(QPOINTER),DIMENSION(4,MAXENTRYNODES)::REAR
END MODULE QUEUE_MOD
```

Name	Description	
QNODE	User defined type storing an element in a linked list queue.	
	Each entry in QNODE contains the following:	
	VEHICLE	VEHICLE structure defining the vehicle to be emitted.
	NEXT	Pointer to the next QNODE.
QPOINTER	User defined type storing pointers to the elements in a linked list queue.	
	Each entry in QPOINTER contains the following:	
	PTR	Pointer to a QNODE structure.
FRONT	Array that stores the pointer to the front of each entry queue.	
REAR	Array that stores the pointer to the rear of each entry queue.	

4.3 DTA Modules

The following module is used by CORSIM to construct a queue of vehicles that are waiting to be emitted into the network by an external DTA system.

```
MODULE DTA_QUEUE_MOD
USE QUEUE_MOD
TYPE(QPOINTER)::DTAFRONT
TYPE(QPOINTER)::DTAREAR
END MODULE DTA_QUEUE_MOD
```

Name	Description
DTAFRONT	Pointer to the front of the DTA entry queue.
DTAREAR	Pointer to the rear of the DTA entry queue.

The following module is used by CORSIM to store data representing paths that DTA generated vehicles can follow. Variables that are exported (DLLEXPORT attribute) can be referenced as PATH_MOD_mp_XXX, where XXX represents the listed name of the variable.

```
MODULE PATH_MOD
USE DATA_MOD
INTEGER,ALLOCATABLE::NNODES(:)
INTEGER,ALLOCATABLE::DTA_PATH(:,:)
INTEGER::PATH(MAXNODESINPATH)
!DEC$ATTRIBUTES DLLEXPORT::PATH
INTEGER::MXPATHS = MAXPATHS
!DEC$ATTRIBUTES DLLEXPORT::MXPATHS
INTEGER::MXNODESINPATH = MAXNODESINPATH
!DEC$ATTRIBUTES DLLEXPORT::MXNODESINPATH
```

INTEGER::NPATHS = 0

!DEC$ATTRIBUTES DLLEXPORT::NPATHS

END MODULE PATH_MOD

4.4 Vehicle Data Module

The following module is used to get vehicle data from CORSIM. It is used by the exported function GetVehicleData. Variables that are exported (DLLEXPORT attribute) can be referenced as Vehicle_Data_Module_mp_XXX, where XXX represents the listed name of the variable. Additional information may be obtained by checking the NETSIM and FRESIM vehicle databases, previously described. The comments on the right in the module give the NETSIM and FRESIM variables that the data items were defined from.

```
Module Vehicle_Data_Module
    Real    Acceleration                          ! ACCEL (IMXVEH)// ZACCEL (IMXFVH)
!DEC$ATTRIBUTES DLLEXPORT::Acceleration
   Integer AccelerationCode                       ! Code (0,1)if accelerating/decelerating.
                                                  ! ACCODE(IMXVEH)/
   Integer QueueCode                              ! Code (0,1)if vehicle (is not,is)in queue.
                                                  ! BLQUE (IMXVEH)
   Integer FollowerID                             ! FOLOWR(IMXVEH)/FOLOW1 (IMXFVH)
   Integer TurnCode                               ! Current Turn movement code.TCODE
                                                  (IMXVEH)/
   Integer VehiclePathID
!DEC$ATTRIBUTES DLLEXPORT::VehiclePathID
   Integer DestinationForVeh                      !NodeID for destination;if -1,vehicle does not
                                                  have a network
                                                  !destination (because it is generated by
                                                  !CORSIM and does not have prespecified
                                                  destination)
!DEC$ATTRIBUTES DLLEXPORT::DestinationForVeh
   INTEGER OriginForVeh                           !NodeID for origin.
!DEC$ATTRIBUTES DLLEXPORT::OriginForVeh
   Real    DistanceToUpNode                       ! DISTUP(IMXVEH)
   Real    DistanceToDownNode                     ! LENGTH - DISTUP(IMXVEH)
                                                  ! current link.   ENTRTM(IMXVEH)/
   Real    DesiredFreeFlowSpeed                   ! Desired free flow speed for this vehicle on
                                                  ! this link. VFFSPD(IMXVEH)/ZFSPD
                                                  (IMXFVH)
   Real    LastLaneChangingTime                   ! LCHGTM(IMXVEH)

   Integer LeadVehForLaneChngToFrceGapOpen        !LCLEAD
   Integer LeaderOfSubjectVehicle                 ! LEADER(IMXVEH)/ VLEAD1 (IMXFVH)
   Integer LaneChangingCode                       !Code(0,1)if veh. (has not,has)changed lanes.
                                                  !LSWCH (IMXVEH)/
   Integer DriverType                             !NDRVRC(IMXVEH)/VDRVRC (IMXFVH)
!DEC$ATTRIBUTES DLLEXPORT::DriverType
   Integer VehicleFleet                           !Vehicle fleet
                                                  component.NFLEET(IMXVEH)/FLEET
                                                  (IMXFVH)
!DEC$ATTRIBUTES DLLEXPORT::VehicleFleet
   Integer VehicleProcessCode                     !NVHCDE(IMXVEH)/VEHCDE (IMXFVH)
   Integer UpStreamNodeofCurrentLink
```

```
!DEC$ATTRIBUTES DLLEXPORT::UpStreamNodeofCurrentLink
     Integer DownStreamNodeofCurrentLink
!DEC$ATTRIBUTES DLLEXPORT::DownStreamNodeofCurrentLink

     Integer VehicleType                          !NVHTYP(IMXVEH)/ VEHTYP (IMXFVH)--
                                                  0,1,..6
!DEC$ATTRIBUTES DLLEXPORT::VehicleType
     Integer LaneID                               !NLANE (IMXVEH)/
!DEC$ATTRIBUTES DLLEXPORT::LaneID
     Real    RemainingPedestrianDelay             !PEDDLY(IMXVEH)/
     Integer LaneOnPreviousLink                   !Lane occupied on previous link.
                                                  !PREVLN(IMXVEH)/
     Integer TurnMovementOnPreviousLink           !PREVTC(IMXVEH)/
     Integer PreferredLaneID                      !PRFDLN(IMXVEH)/
     Integer PreviousGoalLane                     !PRVGLN(1-7)
     Integer PreviousTurnCode                     !PRVGLN(8-10)
     Real    Speed                                !SPDLN / ZSPEED (IMXFVH)
!DEC$ATTRIBUTES DLLEXPORT::Speed
     Integer MovementStatus                       !LKSTOP
     Integer DriverTypeFactorForLaneChange        !AGGFCT/
                              !Used to compute driver aggressiveness.
     Integer LaneChangeAggressivenessFactor
     Real    TimeRemainingToDischargeFromQue      !TIMDIS(IMXVEH)/ZVDSCH (IMXFVH)
     Integer LaneChangeStatus                     !Flag (1,0)if (is,is not)changing lanes.
                                                  !VCHNG(IMXVEH)/
     Integer TargetLane                           !VHLANE(1-3)
     Integer Goalane                              !VHLANE(4-10)
     Integer DriverCooperativeForLaneChange       !VHLANE(11)
                                                  !which seeks a lane change because
                                                  ! another bus is in dwell downstream
                                                  !and it does not service that station
     Real    RemainingTimeOfLaneChange            !VHLANE(13-15)/ZCHTIM (IMXFVH)
     Integer StatusCode                           !VSTATE  / VFSTAT (IMXFVH)??
     Real EnterTimeOnCurrentLink

                                                  !XGOALN(IMXVEH * 3):should not be used.
     Real    EnterNetworkTime                     !ENTIME(IMXVEH)/ZENTIM (IMXFVH)
!DEC$ATTRIBUTES DLLEXPORT::EnterNetworkTime

c    The followings are specific to FRESIM:
     Integer OnRampID                             !Lane No. at which desired free-flow speed is
                                                  reset.
                                                  !/ONRAMP (IMXFVH)
     Integer CandidateLaneCode                    !(0,1,2,3)the lane {is,is not,exists,does't exist}
                                                  !candidate lane. /VCAND  (IMXFVH * 11)
     Integer DetectorIDLastCrossedByVeh           !/VDETCH (IMXFVH)
     Integer UpNodeofLinkIDVehExitsMainLane       !Upnode of Link at which veh exits the main
                                                  lane in segment.
                              !/VEHDST (IMXFVH)
     Integer DwnNodeofLinkIDVehExitsMainLane      !Downnode of Link at which veh exits the main
                                                  lane in segment.
                                                  !/VEHDST (IMXFVH)
     Integer AuxillaryLeader                      !VehicleID of the auxillary leader./ VLEAD2
                                                  (IMXFVH)
     Real    PreviousAcceleration                 !/ ZACCP  (IMXFVH)
     Real    LaneChangeImpatienceFactor           !/ZIMPT  (IMXFVH)
```

```
Integer CodeVehFormerInFrontLaneChange      !CODE (0, > 0) IF VEHICLE FORMERLY IN
                                            FRONT OF THIS
                                             !VEHICLE (IS NOT, IS) STILL IN PROCESS
                                            OF A LANE CHANGE.
Real    TolerableSpeed                      !/ ZTLRSP (IMXFVH)
Real    RubberneckingFactor                 !/   ZVRUB  (IMXFVH)
Integer ComplianceFactor                    !
                                            NCOMPLIANCEFACTOR(IMXVEH)/FCOM
                                            PLIANCEFACTOR(IMXVEH)
Integer ATISAccessability                   !NATISACCESS(IMXVEH)/
                                            FATISACCESS(IMXFVH)

End  Module Vehicle_Data_Module
```

Name	**Description**
Acceleration	Vehicle acceleration
AccelerationCode	Code to identify whether vehicle is accelerating (0) or decelerating (1).
QueueCode	Set to (0,1) if vehicle (is not, is) in queue formed by blocker or parker. NETSIM only
FollowerIDTurnCode	Number of vehicle following this vehicle, IV, in the same lane and link
VehiclePathID	Path ID for a path following vehicle
DestinationForVeh	Node id for destination for a path following vehicle
OriginForVeh	Node id for origin for vehicle
DistanceToUpNode	Distance of vehicle, IV, from upstream node
DistanceToDownNode	Distance of vehicle, IV, to downstream node
DesiredFreeFlowSpeed	Desired free-flow speed
LastLaneChangingTime	Last time vehicle changed lanes
LeadVehForLaneChngToFrceGapOpen	Lead vehicle for lane change to force gap (NETSIM only)
LeaderOfSubjectVehicle	Vehicle leading this vehicle in current lane and link
LaneChangingCode	Set to (0,1) if vehicle (has not, has) switched lanes to bypass blocker (NETSIM only)
DriverType	Driver code of vehicle. Values range from 0 (timid) to 10 (aggressive)
VehicleFleet	Code identifying vehicle fleet component: 0 Auto, 1 Truck, 2 Car pool , 3 Bus
VehicleProcessCode	Vehicle process code (0,1) if vehicle, (has not, has) been processed in the current time step.
UpStreamNodeofCurrentLink	User-defined upstream node number for the current link
DownStreamNodeofCurrentLink	User-defined downstream node number for the current link
VehicleType	Vehicle type
LaneID	Lane currently occupied

Name	Description
RemainingPedestrianDelay	Time remaining for pedestrian delay (NETSIM only)
LaneOnPreviousLink	Lane number occupied by the vehicle on previous link (NETSIM only)
TurnMovementOnPreviousLink	Turn code on previous link (NETSIM only)
PreferredLaneID	Preferred lane (NETSIM only)
PreviousGoalLane	Vehicle's previous goal lane before changed due to an intersection blockage (NETSIM only)
PreviousTurnCode	Vehicle's previous turn code before changed due to an intersection blockage (NETSIM only)
Speed	Vehicle speed,
MovementStatus	Code set to 1 if vehicle stops on link currently occupied (NETSIM only)
DriverTypeFactorForLaneChange	Driver type factor used to compute driver aggressiveness (NETSIM only)
LaneChangeAggressivenessFactor	1.0 + (DriverType - 5.5) / (DriverTypeFactorForLaneChange) (NETSIM only)
TimeRemainingToDischargeFromQue	Time to discharge from queue (NETSIM sec*10, FRESIM sec)
LaneChangeStatus	1 if vehicle is changing lanes; 0 if vehicle is not
TargetLane	Target lane (NETSIM only)
Goalane	Goal lanes (NETSIM only)
DriverCooperativeForLaneChange	1 if driver is cooperative to a lane changer (NETSIM only)
RemainingTimeOfLaneChange	Time vehicle has left to complete the lane change
StatusCode	Vehicle status code
EnterTimeOnCurrentLink	Entry time to the current link
EnterNetworkTime	Time in seconds that vehicle, IV, entered network (NETSIM only)
OnRampID	When vehicle is merging (as in an onramp merge) the vehicle will set its speed to the speed of the adjacent lane. This array holds the lane ID that when reached, the vehicle will reset its desired speed (FRESIM only)
CandidateLaneCode	Candidate lane code (FRESIM only)
DetectorIDLastCrossedByVeh	ID of detector vehicle is currently activating (FRESIM only)
UpNodeofLinkIDVehExitsMainLane	Upnode of the exit link of the mainline for which the vehicle is destined. (FRESIM only)
DwnNodeofLinkIDVehExitsMainLane	Downnode of the exit link of the mainline for which the vehicle is destined. (FRESIM only)
AuxillaryLeader	Secondary leader of vehicle (FRESIM only)
PreviousAcceleration	Acceleration of vehicle IV in the previous time step (FRESIM only)

Name	Description
LaneChangeImpatienceFactor	Lane changing impatience factor (FRESIM only)
CodeVehFormerInFrontLaneChange	If greater than zero, time until its previous leader completes its lane change. Otherwise it is set to 0 (leader not changing lanes) (FRESIM only)
TolerableSpeed	Vehicle tolerable speed (FRESIM only)
RubberneckingFactor	Rubbernecking factor (FRESIM only)
ComplianceFactor	Degree of compliance with VMS instructions
ATISAccessability	Integer representing ATIS accessibility

5 LOWRAM

The LOWRAM array is an emulation of the MOTOROLA 6800 RAM memory used in the modified Q5 actuated control logic. The following table describes the contents of the array. If the description for an array index is blank, the memory location is not used in CORSIM's implementation of the Q5 logic. In the code, some of the array elements have equivalent variable names set via the FORTRAN EQUIVALENCE statement. These variable names are indicated in the "Equivalence" column.

Array Index	Input	Description	Equivalence
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19		Coordination plan: 1 = coordinated, 14 = free	CURCP
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			

Array Index	Input	Description	Equivalence
33			
34		Extended leading side-street left-turn phase force-off adjustment flags	FO37ADJ
35		Hold phases	HOLD
36		Upcoming phases	FAZNX
37		Local cycle timer	LCYTMR
38		Master cycle timer	MCYTMR
39		Coordinated phases	COORD
40		Force off flags	FORCE
41		Motorola accumulator A	ACCA
42		Motorola accumulator B	ACCB
43		Motorola index register X	IXREG
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			

Array Index	Input	Description	Equivalence
87			
88			
89			
90		Sync phases that have yielded this cycle	YIELD
91			
92			
93			
94			
95			
96			
97			
98			
99			
100		Call status	CLSTAT
101			
102		Detector phase assignment	FAZBIT
103		New calls	NUCALL
104			
105			
106			
107		Temporary storage	T3P
108		Previous green phases (0.1 sec ago)	OLGFZ
109			
110			
111			
112			
113			
114		Master sync point (coordination) flag	CORDA
115			
116			
117			
118			
119			
120		Currently allowed phases	ALLOW
121			
122			
123			
124		Type III detector flags	T3ACT
125			
126			
127		Temporary storage	TEMP1
128		Phase 1 detector actuation flag (by port) current	
129		Phase 2 detector actuation flag (by port) current	
130		Phase 3 detector actuation flag (by port) current	
131		Phase 4 detector actuation flag (by port) current	
132		Phase 5 detector actuation flag (by port) current	
133		Phase 6 detector actuation flag (by port) current	
134		Phase 7 detector actuation flag (by port) current	
135		Phase 8 detector actuation flag (by port) current	
136		Phase 1 detector actuation flag (by port) previous second	
137		Phase 2 detector actuation flag (by port) previous second	
138		Phase 3 detector actuation flag (by port) previous second	
139		Phase 4 detector actuation flag (by port) previous second	
140		Phase 5 detector actuation flag (by port) previous second	

Array Index	Input	Description	Equivalence
141		Phase 6 detector actuation flag (by port) previous second	
142		Phase 7 detector actuation flag (by port) previous second	
143		Phase 8 detector actuation flag (by port) previous second	
144		Phases 3 & 4 signal output	OUT1
145		Phases 1 & 2 signal output – not used	
146		Phases 7 & 8 signal output	OUT3
147		Phases 5 & 6 signal output – not used	
148		Pedestrian yellows, auxiliary, watchdog	OUT5
149			
150			
151			
152			
153			
154		Temporary index save	XTEMP
155		Temporary save and Phase number x 16, used for indexing	
156		Ring index save	RNGIDX
157			
158		Temporary save	TEMP
159		Ring service flags	RGSERV
160		Green phases	GFAZE
161		Walk phases	WFAZE
162		Don't walk phases	DWFAZE
163		Yellow phases	YFAZE
164		Red phases	RFAZE
165		Current phase flags	FAZIN
166		zeroed but not used	
167		zeroed but not used	
168		zeroed but not used	
169		zeroed but not used	
170		Phases to be advanced	ADVAN
171		Internal recall	IRCALL
172		Max termination calls	MXCALL
173		Vehicle calls	VCALL
174		Pedestrian calls	PCALL
175		Composite calls	XCALL
176			
177		Saved pedestrian calls	SVPED
178		Saved vehicle calls	SVVEH
179			
180			
181			
182			
183			
184			
185			
186			
187			
188			
189			
190			
191			
192		Ring A Current Phase – RAIDX – Ring A call mask = 192	
193		Ring A phase attribute 2 for current phase	
194		Ring A phase attribute 3 for current phase: paired phase	

Array Index	Input	Description	Equivalence
195		Ring A phase attribute 4 for current phase: cross-street phases	
196		Ring A phase attribute 5 for current phase: don't walk output signal code	
197		Ring A phase attribute 6 for current phase	
198		Ring A phase attribute 7 for current phase: signal output address	
199		Ring A phase attribute 8 for current phase – loaded but not used	
200		Ring A state/termination flags (see index 216 for description)	
201		unknown	
202			
203		unknown	
204		Ring A color pointer	RACOLR
205		unknown – loaded but not used	
206		xxx	
207		Ring A phase number x 16, used for indexing	
208		Ring B Current Phase – RBIDX – Ring B call mask = 208	
209		Ring B phase attribute 2 for current phase	
210		Ring B phase attribute 3 for current phase: paired phase	
211		Ring B phase attribute 4 for current phase: cross-street phases	
212		Ring B phase attribute 5 for current phase: don't walk signal output code	
213		Ring B phase attribute 6 for current phase	
214		Ring B phase attribute 7 for current phase: signal output address	
215		Ring B phase attribute 8 for current phase – loaded but not used	
216		Ring B state/termination flags bit 0 = peds ? bit 1 = peds ? bit 2 = in min green timing bit 3 = in variable initial timing bit 4 = ? bit 5 = ? bit 6 = gap termination bit 7 = max termination	
217		unknown	
218			
219		unknown	
220		Ring B color pointer	RBCOLR
221		unknown – loaded but not used	
222		unknown	
223		Ring B phase number x 16, used for indexing	
224		Ring A guaranteed passage timer	
225		Ring A extended leading side-street left-turn phase	RAXFAZ
226		Ring A extended leading side-street left-turn phase force-off extension time	RAXTIM
227			
228			
229			
230			
231			
232			
233			
234			
235			
236			
237			
238			
239			

Array Index	Input	Description	Equivalence
240		Ring B guaranteed passage timer	
241		Ring B extended leading side-street left-turn phase	RBXFAZ
242		Ring B extended leading side-street left-turn phase force-off extension time	RBXTIM
243			
244			
245			
246			
247			
248			
249			
250			
251			
252			
253			
254			
255			
256			
257		Expired red clearance timer flag set by subroutine YELRT	
258			
259			
260			
261			
262			
263			
264			
265			
266		Ring A Type III detector limit timer	RA3LM
267		Ring B Type III detector limit timer	RB3LM
268			
269			
270			
271			
272	●	Phase 1 Walk time	
273	●	Phase 1 Don't walk time	
274	●	Phase 1 Minimum green time	
275	●	Phase 1 Type III detector limit time setting	
276	●	Phase 1 Time added to the initial interval for each vehicle actuation	
277	●	Phase 1 Unit extension time (passage time)	
278	●	Phase 1 Maximum gap	
279	●	Phase 1 Minimum gap	
280	●	Phase 1 Maximum extension	
281	●	Phase 1 Number of actuations serviced during minimum initial	
282	●	Phase 1 Maximum initial time	
283	●	Phase 1 Red revert time	
284	●	Phase 1 Reduce by time	
285	●	Phase 1 Reduce every time	
286	●	Phase 1 Duration of yellow change interval	
287	●	Phase 1 Duration of red clearance interval	
288	●	Phase 2 Walk time	
289	●	Phase 2 Don't walk time	
290	●	Phase 2 Minimum green time	
291	●	Phase 2 Type III detector limit time setting	
292	●	Phase 2 Time added to the initial interval for each vehicle actuation	

Array Index	Input	Description	Equivalence
293	●	Phase 2 Unit extension time (passage time)	
294	●	Phase 2 Maximum gap	
295	●	Phase 2 Minimum gap	
296	●	Phase 2 Maximum extension	
297	●	Phase 2 Number of actuations serviced during minimum initial	
298	●	Phase 2 Maximum initial time	
299	●	Phase 2 Red revert time	
300	●	Phase 2 Reduce by time	
301	●	Phase 2 Reduce every time	
302	●	Phase 2 Duration of yellow change interval	
303	●	Phase 2 Duration of red clearance interval	
304	●	Phase 3 Walk time	
305	●	Phase 3 Don't walk time	
306	●	Phase 3 Minimum green time	
307	●	Phase 3 Type III detector limit time setting	
308	●	Phase 3 Time added to the initial interval for each vehicle actuation	
309	●	Phase 3 Unit extension time (passage time)	
310	●	Phase 3 Maximum gap	
311	●	Phase 3 Minimum gap	
312	●	Phase 3 Maximum extension	
313	●	Phase 3 Number of actuations serviced during minimum initial	
314	●	Phase 3 Maximum initial time	
315	●	Phase 3 Red revert time	
316	●	Phase 3 Reduce by time	
317	●	Phase 3 Reduce every time	
318	●	Phase 3 Duration of yellow change interval	
319	●	Phase 3 Duration of red clearance interval	
320	●	Phase 4 Walk time	
321	●	Phase 4 Don't walk time	
322	●	Phase 4 Minimum green time	
323	●	Phase 4 Type III detector limit time setting	
324	●	Phase 4 Time added to the initial interval for each vehicle actuation	
325	●	Phase 4 Unit extension time (passage time)	
326	●	Phase 4 Maximum gap	
327	●	Phase 4 Minimum gap	
328	●	Phase 4 Maximum extension	
329	●	Phase 4 Number of actuations serviced during minimum initial	
330	●	Phase 4 Maximum initial time	
331	●	Phase 4 Red revert time	
332	●	Phase 4 Reduce by time	
333	●	Phase 4 Reduce every time	
334	●	Phase 4 Duration of yellow change interval	
335	●	Phase 4 Duration of red clearance interval	
336	●	Phase 5 Walk time	
337	●	Phase 5 Don't walk time	
338	●	Phase 5 Minimum green time	
339	●	Phase 5 Type III detector limit time setting	
340	●	Phase 5 Time added to the initial interval for each vehicle actuation	
341	●	Phase 5 Unit extension time (passage time)	
342	●	Phase 5 Maximum gap	
343	●	Phase 5 Minimum gap	
344	●	Phase 5 Maximum extension	
345	●	Phase 5 Number of actuations serviced during minimum initial	
346	●	Phase 5 Maximum initial time	

Array Index	Input	Description	Equivalence
347	●	Phase 5 Red revert time	
348	●	Phase 5 Reduce by time	
349	●	Phase 5 Reduce every time	
350	●	Phase 5 Duration of yellow change interval	
351	●	Phase 5 Duration of red clearance interval	
352	●	Phase 6 Walk time	
353	●	Phase 6 Don't walk time	
354	●	Phase 6 Minimum green time	
355	●	Phase 6 Type III detector limit time setting	
356	●	Phase 6 Time added to the initial interval for each vehicle actuation	
357	●	Phase 6 Unit extension time (passage time)	
358	●	Phase 6 Maximum gap	
359	●	Phase 6 Minimum gap	
360	●	Phase 6 Maximum extension	
361	●	Phase 6 Number of actuations serviced during minimum initial	
362	●	Phase 6 Maximum initial time	
363	●	Phase 6 Red revert time	
364	●	Phase 6 Reduce by time	
365	●	Phase 6 Reduce every time	
366	●	Phase 6 Duration of yellow change interval	
367	●	Phase 6 Duration of red clearance interval	
368	●	Phase 7 Walk time	
369	●	Phase 7 Don't walk time	
370	●	Phase 7 Minimum green time	
371	●	Phase 7 Type III detector limit time setting	
372	●	Phase 7 Time added to the initial interval for each vehicle actuation	
373	●	Phase 7 Unit extension time (passage time)	
374	●	Phase 7 Maximum gap	
375	●	Phase 7 Minimum gap	
376	●	Phase 7 Maximum extension	
377	●	Phase 7 Number of actuations serviced during minimum initial	
378	●	Phase 7 Maximum initial time	
379	●	Phase 7 Red revert time	
380	●	Phase 7 Reduce by time	
381	●	Phase 7 Reduce every time	
382	●	Phase 7 Duration of yellow change interval	
383	●	Phase 7 Duration of red clearance interval	
384	●	Phase 8 Walk time	
385	●	Phase 8 Don't walk time	
386	●	Phase 8 Minimum green time	
387	●	Phase 8 Type III detector limit time setting	
388	●	Phase 8 Time added to the initial interval for each vehicle actuation	
389	●	Phase 8 Unit extension time (passage time)	
390	●	Phase 8 Maximum gap	
391	●	Phase 8 Minimum gap	
392	●	Phase 8 Maximum extension	
393	●	Phase 8 Number of actuations serviced during minimum initial	
394	●	Phase 8 Maximum initial time	
395	●	Phase 8 Red revert time	
396	●	Phase 8 Reduce by time	
397	●	Phase 8 Reduce every time	
398	●	Phase 8 Duration of yellow change interval	
399	●	Phase 8 Duration of red clearance interval	
400	●	Phase 1 Max green time	

Array Index	Input	Description	Equivalence
401	●	Phase 2 Max green time	
402	●	Phase 3 Max green time	
403	●	Phase 4 Max green time	
404	●	Phase 5 Max green time	
405	●	Phase 6 Max green time	
406	●	Phase 7 Max green time	
407	●	Phase 8 Max green time	
408			
409			
410			
411			
412			
413			
414			
415			
416		Ring A Walk timer	
417		Ring A Pedestrian clearance timer	
418		Ring A Min green timer	
419		Ring A Variable initial timer	
420		Ring A Gap timer (current remaining gap time)	
421		Ring A Vehicle extension time (passage time)	
422		Ring A Current gap value	
423		Ring A Minimum gap	
424		Ring A Max extension timer	
425		Ring A Vehicles served during min initial	
426		Ring A Max initial	
427		Ring A Red revert timer	
428		Ring A Gap reduction factor (Reduce By)	
429		Ring A Step timer	
430		Ring A Yellow timer (current remaining yellow time)	
431		Ring A Red clearance timer	
432		Ring B Walk timer	
433		Ring B Pedestrian clearance timer	
434		Ring B Min green timer	
435		Ring B Variable initial timer	
436		Ring B Gap timer (current remaining gap time)	
437		Ring B Vehicle extension time (passage time)	
438		Ring B Current gap value	
439		Ring B Minimum gap	
440		Ring B Max extension timer	
441		Ring B Vehicles served during min initial	
442		Ring B Max initial	
443		Ring B Red revert timer	
444		Ring B Gap reduction factor (Reduce By)	
445		Ring B Step timer	
446		Ring B Yellow timer (current remaining yellow time)	
447		Ring B Red clearance timer	
448			
449			
450			
451			
452			
453			
454			

Array Index	Input	Description	Equivalence
455			
456			
457			
458			
459			
460			
461			
462			
463			
464			
465			
466			
467			
468			
469			
470			
471			
472			
473			
474			
475			
476			
477			
478			
479			
480			
481			
482			
483			
484			
485			
486	●	Bit oriented word indicating Overlap C phases	
487	●	Bit oriented word indicating Overlap D phases	
488			
489			
490			
491			
492			
493			
494			
495			
496	●	Bit oriented word indicating Permitted phases	PERMIT
497	●	Bit oriented word indicating Detector lock on red phases	RLOCK
498	●	Bit oriented word indicating Detector lock on yellow phases	YLOCK
499	●	Bit oriented word indicating Vehicle minimum recall phases	VRCAL
500	●	Bit oriented word indicating Pedestrian recall phases	PRCAL
501	●	Bit oriented word indicating Pedestrian permitted phases	PED
502	●	Bit oriented word indicating Overlap A phases	OLAPA
503	●	Bit oriented word indicating Overlap B phases	OLAPB
504	●	Bit oriented word indicating Double entry phases	DBLENT
505			
506	●	Bit oriented word indicating lag phases on both rings	LAG
507	●	Bit oriented word indicating red rest phases	RREST
508	●	Bit oriented word indicating pedestrian rest codes	PEDRST

Array Index	Input	Description	Equivalence
509		Simultaneous gap out phases	SIMGAP
510		Lag hold phases	LGHOLD
511		Conditional service phases	CONSRV
512	●	Phase 1 Detector 1 delay time setting	
513	●	Phase 1 Detector 2 delay time setting	
514	●	Phase 1 Detector 3 delay time setting	
515	●	Phase 1 Detector 4 delay time setting	
516	●	Phase 1 Detector 5 delay time setting	
517	●	Phase 1 Detector 6 delay time setting	
518	●	Phase 2 Detector 1 delay time setting	
519	●	Phase 2 Detector 2 delay time setting	
520	●	Phase 2 Detector 3 delay time setting	
521	●	Phase 2 Detector 4 delay time setting	
522	●	Phase 2 Detector 5 delay time setting	
523	●	Phase 2 Detector 6 delay time setting	
524	●	Phase 3 Detector 1 delay time setting	
525	●	Phase 3 Detector 2 delay time setting	
526	●	Phase 3 Detector 3 delay time setting	
527	●	Phase 3 Detector 4 delay time setting	
528	●	Phase 3 Detector 5 delay time setting	
529	●	Phase 3 Detector 6 delay time setting	
530	●	Phase 4 Detector 1 delay time setting	
531	●	Phase 4 Detector 2 delay time setting	
532	●	Phase 4 Detector 3 delay time setting	
533	●	Phase 4 Detector 4 delay time setting	
534	●	Phase 4 Detector 5 delay time setting	
535	●	Phase 4 Detector 6 delay time setting	
536	●	Phase 5 Detector 1 delay time setting	
537	●	Phase 5 Detector 2 delay time setting	
538	●	Phase 5 Detector 3 delay time setting	
539	●	Phase 5 Detector 4 delay time setting	
540	●	Phase 5 Detector 5 delay time setting	
541	●	Phase 5 Detector 6 delay time setting	
542	●	Phase 6 Detector 1 delay time setting	
543	●	Phase 6 Detector 2 delay time setting	
544	●	Phase 6 Detector 3 delay time setting	
545	●	Phase 6 Detector 4 delay time setting	
546	●	Phase 6 Detector 5 delay time setting	
547	●	Phase 6 Detector 6 delay time setting	
548	●	Phase 7 Detector 1 delay time setting	
549	●	Phase 7 Detector 2 delay time setting	
550	●	Phase 7 Detector 3 delay time setting	
551	●	Phase 7 Detector 4 delay time setting	
552	●	Phase 7 Detector 5 delay time setting	
553	●	Phase 7 Detector 6 delay time setting	
554	●	Phase 8 Detector 1 delay time setting	
555	●	Phase 8 Detector 2 delay time setting	
556	●	Phase 8 Detector 3 delay time setting	
557	●	Phase 8 Detector 4 delay time setting	
558	●	Phase 8 Detector 5 delay time setting	
559	●	Phase 8 Detector 6 delay time setting	
560		Phase 1 Detector 1 delay timer	
561		Phase 1 Detector 2 delay timer	
562		Phase 1 Detector 3 delay timer	

Array Index	Input	Description	Equivalence
563		Phase 1 Detector 4 delay timer	
564		Phase 1 Detector 5 delay timer	
565		Phase 1 Detector 6 delay timer	
566		Phase 2 Detector 1 delay timer	
567		Phase 2 Detector 2 delay timer	
568		Phase 2 Detector 3 delay timer	
569		Phase 2 Detector 4 delay timer	
570		Phase 2 Detector 5 delay timer	
571		Phase 2 Detector 6 delay timer	
572		Phase 3 Detector 1 delay timer	
573		Phase 3 Detector 2 delay timer	
574		Phase 3 Detector 3 delay timer	
575		Phase 3 Detector 4 delay timer	
576		Phase 3 Detector 5 delay timer	
577		Phase 3 Detector 6 delay timer	
578		Phase 4 Detector 1 delay timer	
579		Phase 4 Detector 2 delay timer	
580		Phase 4 Detector 3 delay timer	
581		Phase 4 Detector 4 delay timer	
582		Phase 4 Detector 5 delay timer	
583		Phase 4 Detector 6 delay timer	
584		Phase 5 Detector 1 delay timer	
585		Phase 5 Detector 2 delay timer	
586		Phase 5 Detector 3 delay timer	
587		Phase 5 Detector 4 delay timer	
588		Phase 5 Detector 5 delay timer	
589		Phase 5 Detector 6 delay timer	
590		Phase 6 Detector 1 delay timer	
591		Phase 6 Detector 2 delay timer	
592		Phase 6 Detector 3 delay timer	
593		Phase 6 Detector 4 delay timer	
594		Phase 6 Detector 5 delay timer	
595		Phase 6 Detector 6 delay timer	
596		Phase 7 Detector 1 delay timer	
597		Phase 7 Detector 2 delay timer	
598		Phase 7 Detector 3 delay timer	
599		Phase 7 Detector 4 delay timer	
600		Phase 7 Detector 5 delay timer	
601		Phase 7 Detector 6 delay timer	
602		Phase 8 Detector 1 delay timer	
603		Phase 8 Detector 2 delay timer	
604		Phase 8 Detector 3 delay timer	
605		Phase 8 Detector 4 delay timer	
606		Phase 8 Detector 5 delay timer	
607		Phase 8 Detector 6 delay timer	
608	●	Phase 1 Detector 1 carry-over time setting	
609	●	Phase 1 Detector 2 carry-over time setting	
610	●	Phase 1 Detector 3 carry-over time setting	
611	●	Phase 1 Detector 4 carry-over time setting	
612	●	Phase 1 Detector 5 carry-over time setting	
613	●	Phase 1 Detector 6 carry-over time setting	
614	●	Phase 2 Detector 1 carry-over time setting	
615	●	Phase 2 Detector 2 carry-over time setting	
616	●	Phase 2 Detector 3 carry-over time setting	

Array Index	Input	Description	Equivalence
617	●	Phase 2 Detector 4 carry-over time setting	
618	●	Phase 2 Detector 5 carry-over time setting	
619	●	Phase 2 Detector 6 carry-over time setting	
620	●	Phase 3 Detector 1 carry-over time setting	
621	●	Phase 3 Detector 2 carry-over time setting	
622	●	Phase 3 Detector 3 carry-over time setting	
623	●	Phase 3 Detector 4 carry-over time setting	
624	●	Phase 3 Detector 5 carry-over time setting	
625	●	Phase 3 Detector 6 carry-over time setting	
626	●	Phase 4 Detector 1 carry-over time setting	
627	●	Phase 4 Detector 2 carry-over time setting	
628	●	Phase 4 Detector 3 carry-over time setting	
629	●	Phase 4 Detector 4 carry-over time setting	
630	●	Phase 4 Detector 5 carry-over time setting	
631	●	Phase 4 Detector 6 carry-over time setting	
632	●	Phase 5 Detector 1 carry-over time setting	
633	●	Phase 5 Detector 2 carry-over time setting	
634	●	Phase 5 Detector 3 carry-over time setting	
635	●	Phase 5 Detector 4 carry-over time setting	
636	●	Phase 5 Detector 5 carry-over time setting	
637	●	Phase 5 Detector 6 carry-over time setting	
638	●	Phase 6 Detector 1 carry-over time setting	
639	●	Phase 6 Detector 2 carry-over time setting	
640	●	Phase 6 Detector 3 carry-over time setting	
641	●	Phase 6 Detector 4 carry-over time setting	
642	●	Phase 6 Detector 5 carry-over time setting	
643	●	Phase 6 Detector 6 carry-over time setting	
644	●	Phase 7 Detector 1 carry-over time setting	
645	●	Phase 7 Detector 2 carry-over time setting	
646	●	Phase 7 Detector 3 carry-over time setting	
647	●	Phase 7 Detector 4 carry-over time setting	
648	●	Phase 7 Detector 5 carry-over time setting	
649	●	Phase 7 Detector 6 carry-over time setting	
650	●	Phase 8 Detector 1 carry-over time setting	
651	●	Phase 8 Detector 2 carry-over time setting	
652	●	Phase 8 Detector 3 carry-over time setting	
653	●	Phase 8 Detector 4 carry-over time setting	
654	●	Phase 8 Detector 5 carry-over time setting	
655	●	Phase 8 Detector 6 carry-over time setting	
656		Phase 1 Detector 1 carry-over timer	
657		Phase 1 Detector 2 carry-over timer	
658		Phase 1 Detector 3 carry-over timer	
659		Phase 1 Detector 4 carry-over timer	
660		Phase 1 Detector 5 carry-over timer	
661		Phase 1 Detector 6 carry-over timer	
662		Phase 2 Detector 1 carry-over timer	
663		Phase 2 Detector 2 carry-over timer	
664		Phase 2 Detector 3 carry-over timer	
665		Phase 2 Detector 4 carry-over timer	
666		Phase 2 Detector 5 carry-over timer	
667		Phase 2 Detector 6 carry-over timer	
668		Phase 3 Detector 1 carry-over timer	
669		Phase 3 Detector 2 carry-over timer	
670		Phase 3 Detector 3 carry-over timer	

Array Index	Input	Description	Equivalence
671		Phase 3 Detector 4 carry-over timer	
672		Phase 3 Detector 5 carry-over timer	
673		Phase 3 Detector 6 carry-over timer	
674		Phase 4 Detector 1 carry-over timer	
675		Phase 4 Detector 2 carry-over timer	
676		Phase 4 Detector 3 carry-over timer	
677		Phase 4 Detector 4 carry-over timer	
678		Phase 4 Detector 5 carry-over timer	
679		Phase 4 Detector 6 carry-over timer	
680		Phase 5 Detector 1 carry-over timer	
681		Phase 5 Detector 2 carry-over timer	
682		Phase 5 Detector 3 carry-over timer	
683		Phase 5 Detector 4 carry-over timer	
684		Phase 5 Detector 5 carry-over timer	
685		Phase 5 Detector 6 carry-over timer	
686		Phase 6 Detector 1 carry-over timer	
687		Phase 6 Detector 2 carry-over timer	
688		Phase 6 Detector 3 carry-over timer	
689		Phase 6 Detector 4 carry-over timer	
690		Phase 6 Detector 5 carry-over timer	
691		Phase 6 Detector 6 carry-over timer	
692		Phase 7 Detector 1 carry-over timer	
693		Phase 7 Detector 2 carry-over timer	
694		Phase 7 Detector 3 carry-over timer	
695		Phase 7 Detector 4 carry-over timer	
696		Phase 7 Detector 5 carry-over timer	
697		Phase 7 Detector 6 carry-over timer	
698		Phase 8 Detector 1 carry-over timer	
699		Phase 8 Detector 2 carry-over timer	
700		Phase 8 Detector 3 carry-over timer	
701		Phase 8 Detector 4 carry-over timer	
702		Phase 8 Detector 5 carry-over timer	
703		Phase 8 Detector 6 carry-over timer	
704		Phase 1 Detector 1 counts	
705		Phase 1 Detector 2 counts	
706		Phase 1 Detector 3 counts	
707		Phase 1 Detector 4 counts	
708		Phase 1 Detector 5 counts	
709		Phase 1 Detector 6 counts	
710		Phase 2 Detector 1 counts	
711		Phase 2 Detector 2 counts	
712		Phase 2 Detector 3 counts	
713		Phase 2 Detector 4 counts	
714		Phase 2 Detector 5 counts	
715		Phase 2 Detector 6 counts	
716		Phase 3 Detector 1 counts	
717		Phase 3 Detector 2 counts	
718		Phase 3 Detector 3 counts	
719		Phase 3 Detector 4 counts	
720		Phase 3 Detector 5 counts	
721		Phase 3 Detector 6 counts	
722		Phase 4 Detector 1 counts	
723		Phase 4 Detector 2 counts	
724		Phase 4 Detector 3 counts	

Array Index	Input	Description	Equivalence
725		Phase 4 Detector 4 counts	
726		Phase 4 Detector 5 counts	
727		Phase 4 Detector 6 counts	
728		Phase 5 Detector 1 counts	
729		Phase 5 Detector 2 counts	
730		Phase 5 Detector 3 counts	
731		Phase 5 Detector 4 counts	
732		Phase 5 Detector 5 counts	
733		Phase 5 Detector 6 counts	
734		Phase 6 Detector 1 counts	
735		Phase 6 Detector 2 counts	
736		Phase 6 Detector 3 counts	
737		Phase 6 Detector 4 counts	
738		Phase 6 Detector 5 counts	
739		Phase 6 Detector 6 counts	
740		Phase 7 Detector 1 counts	
741		Phase 7 Detector 2 counts	
742		Phase 7 Detector 3 counts	
743		Phase 7 Detector 4 counts	
744		Phase 7 Detector 5 counts	
745		Phase 7 Detector 6 counts	
746		Phase 8 Detector 1 counts	
747		Phase 8 Detector 2 counts	
748		Phase 8 Detector 3 counts	
749		Phase 8 Detector 4 counts	
750		Phase 8 Detector 5 counts	
751		Phase 8 Detector 6 counts	